Représentation dynamique dans le cortex préfrontal : comparaison entre reservoir computing et neurophysiologie du primate

Pierre Enel

▶ **To cite this version:**

Pierre Enel. Représentation dynamique dans le cortex préfrontal : comparaison entre reservoir computing et neurophysiologie du primate. Psychologie et comportements. Université Claude Bernard - Lyon I, 2014. Français. NNT : 2014LYO10091 . tel-01056696

HAL Id: tel-01056696

https://theses.hal.science/tel-01056696

Submitted on 20 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N°d'ordre 91 - 2014 Année 2014

THESE DE L'UNIVERSITE DE LYON
présentée devant
L'UNIVERSITE CLAUDE BERNARD LYON 1
Ecole Doctorale Neurosciences et Cognition
pour l'obtention du
DIPLÔME DE THÈSE
(arrêté du 7 août 2006)
Discipline : Neurosciences
Option : INFORMATIQUE
présentée et soutenue publiquement le 2 Juin 2014
par

Pierre ENEL

Représentation Dynamique dans le Cortex Préfrontal: Comparaison entre Reservoir Computing et Neurophysiologie du Primate

Dynamic Representation in the Prefrontal Cortex: Insights from Comparing Reservoir Computing and Primate Neurophysiology

dirigée par Peter F. DOMINEY

Pr. Rémi GERVAIS	Président du jury
Dr. Mark STOKES	Rapporteur
Dr. Angelo ARLEO	Rapporteur
Pr. Xiao Jing WANG	Examinateur
Dr. Emmanuel PROCYK	Examinateur
Dr. Peter Ford DOMINEY	Directeur de thèse

Stem-Cell and Brain Research Institute, INSERM U846
18, avenue Doyen Lepine
6975 Bron Cedex

École doctorale Neurosciences et Cognition (ED 476 - NSCo)
UCBL - Lyon 1 - Campus de Gerland
50, avenue Tony Garnier 69366 Lyon Cedex 07

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université	**M. François-Noël GILLY**
Vice-président du Conseil d'Administration	M. le Professeur Hamda BEN HADID
Vice-président du Conseil des Etudes et de la Vie Universitaire	M. le Professeur Philippe LALLE
Vice-président du Conseil Scientifique	M. le Professeur Germain GILLET
Directeur Général des Services	M. Alain HELLEU

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard	Directeur : M. le Professeur J. ETIENNE
Faculté de Médecine et de Maïeutique Lyon Sud – Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté d'Odontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. le Professeur Y. MATILLON
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme. la Professeure A-M. SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. FLEURY
Département Chimie Biochimie	Directeur : Mme Caroline FELIX
Département GEP	Directeur : M. Hassan HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. Georges TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. Jean-Claude PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de l'Univers de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. P. FOURNIER
Ecole Supérieure de Chimie Physique Electronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. C. VITON
Ecole Supérieure du Professorat et de l'Education	Directeur : M. A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

Résumé

Les primates doivent pouvoir reconnaître de nouvelles situations pour pouvoir s'y adapter. La représentation de ces situations dans l'activité du cortex est le sujet de cette thèse. Les situations complexes s'expliquent souvent par l'interaction entre des informations sensorielles, internes et motrices. Des activités unitaires dénommées sélectivité mixte, qui sont très présentes dans le cortex préfrontal (CPF), sont un mécanisme possible pour représenter n'importe quelle interaction entre des informations. En parallèle, le Reservoir Computing a démontré que des réseaux récurrents ont la propriété de recombiner des entrées actuelles et passées dans un espace de plus haute dimension, fournissant ainsi un pré-codage potentiellement universel de combinaisons pouvant être ensuite sélectionnées et utilisées en fonction de leur pertinence pour la tâche courante. En combinant ces deux approches, nous soutenons que la nature fortement récurrente de la connectivité locale du CPF est à l'origine d'une forme dynamique de sélectivité mixte. De plus, nous tentons de démontrer qu'une simple régression linéaire, implémentable par un neurone seul, peut extraire n'importe qu'elle information/contingence encodée dans ces combinaisons complexes et dynamiques. Finalement, les entrées précédentes, qu'elles soient sensorielles ou motrices, à ces réseaux du CPF doivent être maintenues pour pouvoir influencer les traitements courants. Nous soutenons que les représentations de ces contextes définis par ces entrées précédentes doivent être exprimées explicitement et retournées aux réseaux locaux du CPF pour influencer les combinaisons courantes à l'origine de la représentation des contingences.

Mots-clefs: cortex préfrontal, réseaux récurrents, sélectivité mixte, dynamique d'attracteur, simulation/modélisation, adaptation du comportement

Abstract

In order to adapt to new situations, primates must be able to recognize these situations. How the cortex represents contingencies in its activity is the main subject of this thesis. First, complex new situations are often explained by the interaction between sensory, internal and motor information. Recent studies have shown that single-neuron actvities referred to as mixed selectivity which are ubiquitous in the prefrontal cortex (PFC) are a possible mechanism to represent arbitrary interaction between information defining a contingency. In parallel, a recent area of reasearch referred to as Reservoir Computing has demonstrated that recurrent neural networks have the property of recombining present and past inputs into a higher dimensional space thereby providing a pre-coding of an essentially universal set of combinations which can then be selected and used arbitrarily for their relevance to the task at hand. Combining these two approaches we argue that the highly recurrent nature of local prefrontal connectivity is at the origin of dynamic form of mixed selectivity. Also, we attempt to demonstrate that a simple linear regression, implementable by a single neuron, can extract any information/contingency encoded in these highly complex and dynamic combinations. In addition, previous inputs, whether sensory or motor, to these PFC networks must be maintained in order to influence current processing and behavioral demand. We argue that representations of contexts defined by these past inputs must be expressed explicitely and fed back to the local PFC networks in order to influence the current combinations at the origin of contingencies representation.

Acknowledgments

First, I would like to thank Peter Dominey my thesis advisor for his constant support and his valuable and unconditional positive attitude. I owe him the discovery of the world of scientific research that I set my self to further explore. The disparate research topics approached by his team has led to rich discussion and allowed to gain perspective on my own field.

I also want thank Emmanuel Procyk that recruited me for my master internship which eventually led to the present thesis. His insights have had a strong influence on the work this thesis.

I would like to thank the Mark Stokes, Angelo Arleo who have had the patience to read this thesis, and accepted to be part of my thesis jury along with Rémi Gervais, Simon Thorpe and Xiao-Jing Wang.

I wamrly thank the members of my team for being such great guys during all these years I have spent in the lab ;)

Xavier deserves a special thank for lively, though very rich discussions we have had through our parallel voyage to becoming doctors. He has opened me to machine learning after an initial refractory period from my part, but eventually led me to develop a strong interest in it.

I am thankful to the electrophy guys, namely, Fred, Mamilys and Chacha who, on top of being great colleagues, have been great friends outside of the lab. I want particularly to thank Charlie who took the time to help me on this present thesis and showed me how I could do better.

I want to thank my parents for their constant care, from bringing me into this world to the unconditional support that eventually led them to host me for more than two weeks in order for me to write this thesis in a small cabin at the back of their garden where these very words are being written.

Contents

Introduction

Une des capacités les plus fondamentales pour s'adapter à l'environnement est la capacité à se représenter les situations auxquelles nous faisons face dans cet environnement. En effet, comment pourrions-nous nous adapter à différentes contingences si nous ne pouvons les reconnaître? La recherche en neuroscience a démontré que l'activité de différentes aires corticales est sélective pour des éléments de l'environnement qui sont particulièrement pertinents dans une situation donnée. La région du cortex qui montre les représentations les plus élaborées et robustes et la partie la plus antérieure du cortex, dénommée le cortex préfrontal (CPF). Cette région affiche des activités sélectives pour des éléments seuls de l'environnement (e.g. l'identité d'un stimulus visuel), mais aussi pour des informations précédentes (e.g. est-ce que la récompense a été mise dans le trou à gauche ou à droite), et, encore plus compliqué, pour une combinaison d'informations sensorielles, internes et motrices. Certaines de ces activités complexes semblent facilement interprétables comme représentant l'association entre différents éléments, e.g. l'identité d'un stimulus et son comportement associé. Cependant, un grand nombre d'activité de neurones n'ont pas de corrélation évident avec un stimulus pertinent, un comportement observable ou une association des deux. Historiquement, l'électrophysiologie s'est concentrée sur l'interprétation d'enregistrements unitaires (l'activité de neurones seuls), essayant de corréler l'activité de ces neurones avec un comportement appris en laboratoire. Donc, les activités complexes qui n'ont pas de corrélation directe avec un comportement observable ont longtemps été ignorées. Récemment seulement elles ont été proposées comme contribuant à la représentation de l'information au niveau de la population, et leur importance pour l'adaptation du comportement a été démontrée ces dernières années par Fusi et collègues.

Cette équipe a proposé de répondre à trois questions:

- est-ce que ces activités complexes sont importantes, bien qu'il n'y ait pas de preuve de leur intérêt pour la tâche effectuée?
- quelle est l'origine de ces activités complexes?
- comment sont représentées les contingences complexes?

Dans un premier article (Rigotti et al., 2010), ils proposent une réponse aux deux dernières questions: les connections aléatoires recombinent les informations déjà représentées, produisant des activités complexes qui sont expliquées par l'influence de plusieurs variables d'une seule tâche, et ces activités peuvent représenter n'importe qu'elle contingence qui dépend de ces variables. Dans un article suivant, (Rigotti et al., 2013), ils démontrent que quand des singes font des erreurs dans une tâche cognitive, les combinaisons complexes (non-linéaires) sont absentes de l'activité neuronale.

Dans cette thèse, nous nous proposons de développer cet argument. Les connexions récurrentes locales du CPF pourrait être à l'origine des combinaisons complexes observées dans l'activité unitaire et pourrait sous-tendre la représentation de n'importe qu'elle contingence qui peut être expliquée par la combinaison d'information sensorielles et motrices récentes et actuelles entrant dans le CPF. Parce que les réseaux récurrents ont une mémoire limitée dans le temps des entrées, les informations précédentes influençant

le comportement présent, i.e. l'information contextuelle, doivent être maintenues. Nous proposons que la représentation explicite d'une information contextuelle (i.e. une activité spécifiquement sélective pour différents contextes) générée par le réseau et retournée au réseau récurrent permet à des entrées non-récentes de se recombiner avec les entrées présentes et de créer des représentations de contingences qui dépendent d'informations non-récentes.

Le premier chapitre de ce manuscrit aborde l'importance de la représentation dans l'activité du CPF pour l'adaptation du comportement. Après une bref revue de l'adaptation du comportement et de la représentation d'information dans le CPF, nous nous proposons de transposer la théorie du sélectionisme en psychologie à la génération de représentation complexes dans l'activité du CPF. Le second chapitre introduit les réseaux de neurones comme outil de modélisation pour comprendre le traitement de l'information et la représentation au niveau corticale. Une courte histoire du connectionisme, le paradigme englobant les réseaux de neurones, nous permet de présenter les différences fondamentales entre les approches précédentes qui tentent d'expliquer la cognition. Après une brève description des architectures classiques dans les réseaux de neurones, nous présentons la théorie des connexions récurrentes pour la représentation de n'importe qu'elle contingence. Le troisième chapitre introduit les réseaux récurrents pour le traitement temporel. Nous présentons d'abord les bases du traitement temporel avec des réseaux récurrent et l'inhérente complexité rencontrée lors de l'apprentissage de tels réseaux, puis nous introduisons les réseaux temporels canoniques et finalement le paradigme du reservoir computing. Le chapitre quatre introduit les représentations explicites de contextes dans différentes architectures de réseaux de neurones, et, particulièrement pertinent pour cette thèse, les représentations de contexte dans des attracteurs. Le dernier chapitre de la section état de l'art présente l'hypothèse et les objectifs. La seconde partie du manuscrit contient les deux articles auxquels l'auteur de cette thèse a contribué. Le premier, publié dans PlosONE, et un article méthodologique qui compare plusieurs décodeurs d'activité neuronale, tandis que le second, à publier, constitue le coeur expérimental de cette thèse. Dans la troisième et dernière partie, nous discutons les résultats de ces deux papiers et les confrontons avec notre hypothèse et la littérature.

Introduction

One of the most fundamental capacity necessary to adapt to the environment is the capacity to represent the situations that we encounter in this environment. Indeed, how would we adapt to different contingencies if we cannot recognize them? Research in neuroscience has shown that the activity of different cortical regions is selective for elements of the environment that are especially pertinent for given situations. The region of the cortex that has shown the most elaborated and robust representations is the anterior most part of the cortex, namely the prefrontal cortex (PFC). This region diplays activities selective for single elements of the environment (e.g. the identity of a visual stimulus), but also for past information (e.g. was the reward put in the left or right pellet), and, even more complicated, for combination of sensory, internal and motor informations. Some of these complex activities seem easily interpreted as representing the association between different elements, e.g. the stimuli identity and the currently associated behavior. However, numerous single-neuron activities have no evident correlation to any pertinent stimulus, observable behavior or their association. Historically, electrophysiology concentrated on the interpretation of single-neuron activities, trying to relate the activity of these neurons with a trained behavior. So complex activities with no direct correlation to observable behaviors have long been ignored. Only recently they have been proposed to contribute to the representation of information at the population level, and their importance for adaptive behavior has been demonstrated these last years by Fusi and colleagues.

This team proposed to answer three questions:

- are these complex activities important, even though there is no clear evidence of their relevance for the task at hand?
- what is the origin of these complex activities?
- how are represented complex contingencies?

In a first article (Rigotti et al., 2010), they propose an answer the two last questions: random connections combine information already represented, eliciting complex activities that are explained by the influence of several variables of a single task, and these activities can represent any contingency that depend on these variables. In a subsequent article (Rigotti et al., 2013), they demonstrate that when monkeys make errors in a cognitive task, complex (non-linear) combinations are absent of neural activities.

In the present thesis, we propose to develop this argument. Local recurrent connections in the PFC may be at the origin of complex combinations observed in the activity of single neurons and would underlie the representation of any contingency that can be explained by the combination of recent and present sensory and motor inputs to the PFC. Because recurrent networks have a time-limited memory of previous inputs, past information that influence current behavior, i.e. contextual information, must be maintained. We propose that an explicit representation of context information (i.e. activity specifically selective for the different contexts) generated by the network and fed back to the recurrent network allow for non-recent inputs to combine with present inputs and participate in creating representations of contingencies that depend on non-rencent information.

The first chapter of this manuscript stress the importance of representation in the activity of the prefrontal cortex for behavioral adaptation. After briefly reviewing behavioral adaptation and representation of information in the PFC, we propose to transpose the theory of selectionism in psychology to the generation of complex representation in PFC activity. The second chapter introduces neural networks as a modeling tool to understand information processing and representation in the cortex. A short history of connectionism, the paradigm encompassing neural networks, allows us to present the fundamental differences between previous approaches attempting to explain cognition. After a brief description of classical architectures in neural networks, we present the theory of random connections for arbitrary contingency representation. The third chapter introduces recurrent networks for temporal processing. We will first present the basics of temporal processing with recurrent networks and the inherent complexity of training such networks, and then introduce the canonical temporal networks and eventually dwell on the presentation of reservoir computing. The fourth chapter introduces explicit representation of context in different neural network architectures, and, particularly relevant to this thesis, the representation of context in attractors. The last chapter of the state of the art section presents the hypothesis and objectives. The second part of the present manuscript includes the two articles in which the author contributed. The first one, published in PlosONE, is a methodological paper comparing several decoders of neural activity, while the second, to be published, constitutes the core experimental work of this thesis. In a third part, we discuss the results of these two papers and confront them with our hypothesis and the litterature.

Part I

State of the Art

Chapter 1

Adaptive Behavior and its Neural Correlates

1.1 Facing a Changing Environment

1.1.1 Short Introduction to Adaptive Behavior

Our environment is constantly changing, at multiple levels. A given action may not always have the same outcomes, and currently rewarded actions may not always remain rewarded over time. In either case adequate behavior must be generated to alleviate the organism's needs. Changes in action-outcome relationships in the environment may occur because resources have simply been depleted. In this case the organism has to forage to find new resources. Alternatively, the way in which resources are available, or the way in which the organism must act to access them, may depend on conditions independent of the organism. In this case complex abilities are required, like recognizing the changing conditions and acting adequately to access resources in new ways. These examples, focused on the primordial aspect of access to resources, do not take into account the myriad of adaptive behaviors that increase the chances of survival of a given organism.

These behaviors are part of the adaptation needed to face the unpredictability of the environment. A broad range of adaptive behavior exists within biological life, from gene regulatory networks to the highly complex cognitive functions found in primates. The advantage of animals over unicellular and vegetable life is their nervous system, which allows them to react to external stimuli at a comparatively short time scale. Nervous systems are therefore well suited to rapidly adapt behavior, but they come with a cost in energy that must be compensated by a higher access to resources. We won't review the diversity of nervous systems and their associated adaptation abilities,

but it is nonetheless worth mentioning that animals with simple sensorimotor loops within their nervous system already display rapid adaptive behaviors. As an example, the renowned study of Carew et al. (1981) demonstrated that an animal that has a relatively simple nervous system – *Aplysia Californica* – displays adaptive behavior on a very short time scale in the form of *habituation* and *sensitization* on top of the animal's naturally occurring gill- and siphon-withdrawal reflex. The richness and diversity of adaptive behavior seem to increase with nervous system complexity. From animals having complex brains to mammals endowed with a cortex, adaptive capacity appears to culminate in primates, which include humans.

1.1.2 Learning and Executive Functions

Any type of learning subserves adaptation. Simple predictions of the state of the environment and the way in which it is changing subserve adaptation. After habituation and sensitization, the most simple form of adaptive behavior is associative learning which includes *classical* and *operant conditioning*. The former is the association of a neutral (conditioned) stimulus and a (unconditioned) stimulus that elicits a response in an animal, through repeated exposure of both stimuli. The latter involves the association between a behavior and its outcome. A mathematical model, known as the *Rescorla–Wagner rule*, has been developed to model classical conditioning (Rescorla and Wagner, 1972), while *reinforcement learning* is the most popular model for operant conditioning (Barto, 1998). The former has inspired the first iterative learning methods for neural networks, while the latter is widely used in neuroscience and psychology to model decision making, and in artificial intelligence (including neural networks) and robotics as a learning method.

However, primates have the ability to learn especially complex relationships between stimuli, context and the outcome of their actions. Their cognitive processes are themselves remarkably flexible. The capacity to manage efficiently the different cognitive processes is usually referred to under the umbrella terms of *executive functions* or *executive control*. A precise definition of executive functions is still lacking, rather, the terms refer to a set of individual cognitive functions that are impaired with PFC lesions. Funahashi (2001), in an attempt to define executive functions, refers to them as "a product of the coordinated operation of various processes to accomplish a particular goal in a flexible manner".

Amongst all the functions falling into this set, we must mention the following ones for their interest to the present thesis: *working memory* (Miller et al., 1968; Baddeley, 2003), the ability to manipulate and actively maintain information for a limited amount of time, *decision making* (Kable and Glimcher, 2009), the process by which an agent

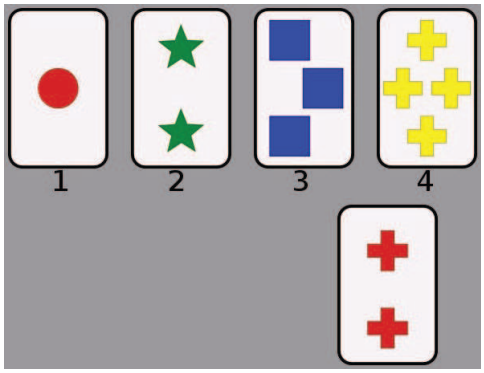

Figure 1.1: Wisconsin Card Sorting Test, a psychological test to assess task switching, an instance of adaptive behavior (screenshot of PEBL software). *Upper row show stimuli cards that are presented to subjects. Each card display visual patterns with three features: their number, shape and color. A test card is presented to the subject who is asked to match it with one of the stimuli cards. In the present example, card 1 is a match by color rule, card 2 by number rule and card 4 by shape rule of the objects. With successive test cards and the feedback of the experimenter, the subject has to find to correct rule. Without notification, the experimenter changes the feedback contingency to assess the capacity of the subject to adapt its response.*

evaluate possible actions and select one among them, and *task switching* (Monsell, 2003), the capacity to switch between different goals, rules or strategies.

We are interested in the ability of primates to associate the same stimuli with different actions depending on context. This involves the ability to switch rapidly between contexts without relearning the behavior associated with a context each time the previous one no longer leads to the desired outcome. Moreover, context must be maintained actively in memory in order to consistently behave adequately.

This cognitive ability has been explored experimentally in psychology and neuropsychology. In these domains, context can refer to a rule that defines the association between stimuli, actions and outcomes. In human psychology, several tests have been developed to assess the ability of subjects to switch between rules, amongst which the *Wisconsin Card Sorting Test* (WCST) is renowned. In this test developed by Grant and Berg (1948), stimuli cards with visual elements are presented to human subjects. Subjects are asked to match new cards with one of the original stimuli cards without being explicitly told how to match them (Fig. 1.1). Subjects learn to match cards by trial and error using the experimenter's feedback, and learn a behavioral rule in doing so (e.g. same color). Without notifying the subject the experimenter changes the rule. As a consequence, the subject has to adapt his response and find the new rule (e.g. same pattern). In this test, subjects demonstrate their ability to switch effectively between rules, i.e. inhibit a learned rule and actively search for a new one.

Another psychological test named the *Stroop test* (Stroop, 1935) illustrates the need for the behavioral control through inhibition. Subjects are asked to name the color of words which themselves refer to colors (e.g. **green**). It takes more time to subjects to name the color of the word than reading it, because there is a conflict between two processes. The automatic process of reading the word interferes with the task of transforming the perception of the color into a word. In this case, adaptation is needed through the executive function referred to as *inhibitory control* which prevents automatic processes to allow relevant behavior to be expressed.

1.1.3 Neuropsychology of PFC

The mammalian neural system is classified in two parts, the central and peripheral nervous system. The peripheral system has its own set of reflex mechanisms contributing to the repertoire of adaptive mechanisms of the animal, however the central nervous system displays very complex flexible behaviors that are responsible for the successful completion of the above mentioned tasks in humans. While the basal ganglia seem to be critical for reinforcement learning (Doya, 1999), the involvement of the cortex in flexible behavior has been progressively revealed by tragic but very informative historical events.

Perhaps the most famous case in neuropsychology is Phineas Gage, a railroad construction foreman who survived the passage of an iron rod through his head, permanently damaging his brain. After his rapid recovery, his wife, workers and employers soon discovered that Gage's personality had change, he became impulsive and could not follow social rules anymore. In addition, his ability to lead and work efficiently were profoundly impaired, as he was easily distracted and was unable to focus on one task, in other words he could not inhibit automatic behavior and switch effectively between tasks (Harlow, 1999). The part of the brain injured was the *prefrontal cortex* (PFC).

Indeed, the rich adaptive behaviors encompassed by the cognitive abilities of primates are attributed to their greatly expanded PFC compared to other mammals (Barbas, 1999). This brain region is one of the cortical structures that underwent the latest development in evolution. The precise delimitation of the PFC is still debated, but a simplistic definition would describe it as the part of the frontal cortex anterior to the motor and premotor areas.

The report of Gage's symptoms was the first well documented case of PFC injury. Subsequent cases along with animal lesion experiments have informed the extent of impairments following PFC lesions. Symptoms are neither related to sensory nor motor behavior but to the mapping between them, especially if a delay is involved. Although very diverse, most typical symptoms include deficits in working memory and discrimination reversal tasks. For example, it has been shown that patients with PFC lesions are

impaired in task switching as revealed with WCST tests (Milner, 1963). Also of interest for this thesis, lesions in the lateral regions of PFC, have shown to impair sequential behaviors (Luria, 1966).

1.1.4 Theory of PFC Function

Globally, neuropsychology has demonstrated the key role of PFC in executive functions. The PFC has a crucial role in the flexible mapping of sensory information with actions directed to an internal goal. To explain this ability, one of the most popular theories of PFC function is the *cognitive control* theory developed by Miller and Cohen (2001). They hypothesize that the main role of the PFC is to actively maintain representations of goals and the means to achieve them. To do so, PFC bias processing in the sensory and motor regions in order to "guide the flow of neural activity along pathways that establish the proper mappings between inputs, internal states, and outputs needed to perform a given task". Increasing the gain of sensory and motor neurons would exert a bias to preferentially process sensory features and motor actions that subserve the current goal.

They list the minimum requirements for this theory to hold:

- PFC must provide a source of activity that can exert the required pattern of biasing signals to other structures
- PFC must maintain its activity robustly against distractions until a goal is achieved, yet also be flexible enough to update its representations when needed
- PFC must house the appropriate representations, those that can select the neural pathways needed for the task
- PFC representations must have a high capacity for multimodality and integration.
- PFC must exhibit a high degree of plasticity

In the third and fourth points, they lay down the requirement of representations that subserve the top-down control of the PFC. Here, *representation* refers to the physically observable activity of neurons that can be correlated by one means or another with an information, whether it is sensory, motor, endogenous or a combination of them. Representations must be highly multidimensional in order to integrate all the information that may be relevant to achieve a goal. In addition, temporal organization of these information may be relevant to goal-directed behavior, therefore it must also be represented in the PFC. Miller and Cohen review few electrophysiological findings that support the existence of diverse representations of rules or mappings between sensory and motor

information in the PFC. However, the mechanisms underlying the genesis and diversity of these representations have only recently been a focus of attention. This question will be central to the present thesis.

1.2 Neuronal Activity Correlates

In this section we will briefly review the single unit and neural population activity correlates of representations related to cognitive control in the PFC. In general, we will use the terms *task variable* to refer to any feature of an experimental task that varies, whether it is a sensory, internal or motor feature.

1.2.1 Basic Single Unit Correlates

Electrophysiological recordings in single neurons of the PFC historically starts with persistent activity during delayed response tasks (Fuster and Alexander, 1971; Fuster, 1973). In the first of these experiments, monkeys are presented with a piece of apple within one of two wells placed in front of them. Wells are covered by objects and a blind is lowered to prevent the monkeys to fix an object. After a delay, the blind is removed and monkeys can reach one of the two objects. Fuster and colleagues observed that neurons in the PFC and thalamus persistently fired during this delay, i.e. tonic activity started with the cued position of the reward and ended as soon as the blind was lifted. They associated this persistent activity with the maintenance in memory of the food location.

Funahashi (1989) replicated these findings with a spatial task that required eye saccades to the spatial location of a cue presented before a delay. Persistent activity has been associated with working memory (Goldman-Rakic, 1995), and has been correlated with the maintenance of different types of information (e.g. visual (Miller et al., 1996), tactile (Romo et al., 1999)). Furthermore, other studies have shown representations of stimuli features and their associated response in the activity of PFC neurons (Watanabe, 1986; Sakagami and Niki, 1994; Hasegawa et al., 1998).

Later, researchers have started to focus on the representation of rules or strategy in the activity of PFC neurons. For example, White and Wise (1999) recorded single neurons in three distinct prefrontal regions while monkeys performed a demanding cognitive task. The animals were required to respond differently when presented with the exact same stimuli, i.e. they had to learn the rules that prescribed the right mapping between stimuli and response (an equivalent of the WCST). The authors reported that between one-third and one-half of the neurons in each region were modulated by the rule. Similar task switching experiments have shown rule-related activity modulation in

PFC (Wallis et al., 2001; Mansouri et al., 2006), and recently Genovesio et al. (2005) reported modulation with strategy.

Globally, single PFC neurons display activities representing a wide variety of information, ranging from sensory features to context and rules. This diversity might be explained by the rich connectivity of the PFC with sensory and motor cortical areas, which may be a critical feature of the PFC to achieve its executive control function (Miller and Cohen, 2001).

1.2.2 Distributed and Rich Representations

While the activities presented in the previous section reflect the encoding of single task variables, some of these experiments also revealed more complex activities explained by the combination of diverse information. Indeed, single prefrontal neurons are rarely selective for a single task variable, but usually respond to a complex combination of several of them. Because of this complexity, the neuronal responses of numerous recorded neurons in the PFC have often been hard to correlate with any aspect of the task at hand, and may have been discarded as irrelevant.

However, recent studies have proposed that these complex activities might play a role in mapping stimuli with responses. In a reversal learning task[1] involving two cues and two motor responses, Asaad et al. (1998) found PFC neurons whose activity differentiate the two cues and both responses (fig. 1.2c). Other neurons responded to the presentation of one of the two cues only when it was associated with a particular motor response (fig. 1.2d). In other words, the activity of these neurons is explained by the combination of two task variables, the cue identity and the motor response. The first case is a linear combination of task variables, which means that the activity of the neuron can be explained by the additive effect of each variable. The second case displays interaction between the task variables, and is therefore a non-linear combination of the task variables.

Previous studies have similarly shown the interactions between several task variables (Watanabe, 1986, 1992; Sakagami and Niki, 1994). These complex activities have been the focus of recent studies that referred to them as (non-linear) *mixed selectivity* (Rigotti et al., 2013). In other words, activities of neurons that could not merely be explained by the additive influence of the task variables, as mentioned above. In this key paper, mixed selectivity refers to non-linear and also to linear combination of task variables. Often overlooked because of the difficulty to interpret them, Rigotti and colleagues showed that the dimensionality of the activity in the neural population recorded was lower

[1] Typically, a task in which the associations between stimuli and rewarded responses are reversed.

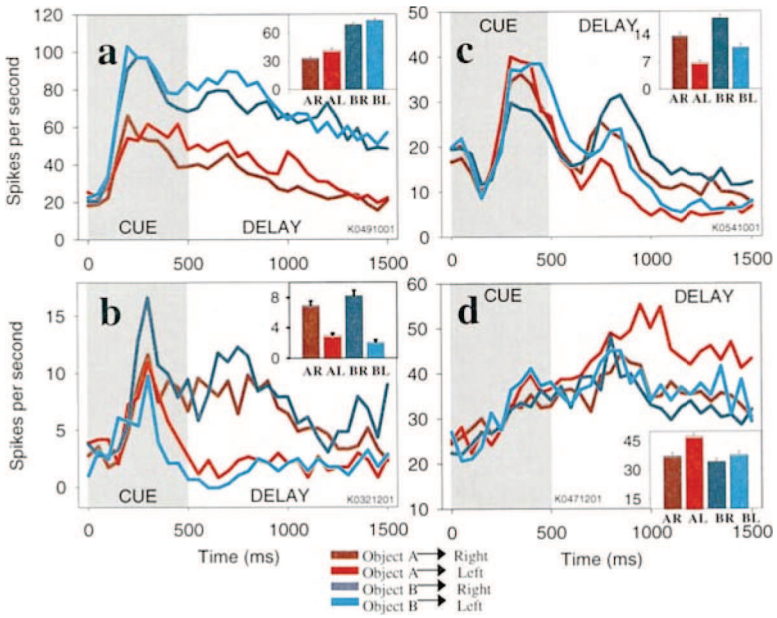

Figure 1.2: Non-linear combination of task variables in PFC (Asaad et al., 1998). *Monkeys in front of a screen have to learn by trial and error to match a visual cue (Object A or B) to a saccade (left or right). Neurons were recorded in the lateral PFC while monkeys performed this task with alternation of the reward contingency. The figure shows the response of 4 neurons for all possible combinations of visual cue and saccade direction. Neuron* ***a*** *and* ***b*** *have easily interpretable responses: they preferentially respond for object B and right saccade, respectively. Activities from neuron* ***c*** *show a more complex nonetheless linear combination of task-variable influences. Neuron* ***d*** *has a non-linear combination of cue and motor response, and can be interpreted as a specific mapping between them.*

when monkeys made errors. Moreover, the difference in dimensionality was attributed to less non-linear activities during these errors trials (Fig. 1.3).

These results demonstrate the necessity for non-linear combination of task variables, and imply that the dimensionality of the neuronal population activity is higher when monkeys perform correctly. The authors showed also that, to some extent, task variables can even be decoded only from the non-linear influence of task variables in population activity. Similarly, it has been shown that neurons whose activity has a low statistical correlation with a particular task variable actually help a decoder to better retrieve the state of that variable (Meyers et al., 2012). Activities that seem unrelated to the task could expand the dimensionality of the neuronal activity. As we will see in the next chapters, high dimensional activity might increase the representational power of the cortex.

In this thesis we will use the term *mixed selectivity* to refer to its non-linear component only. It is defined as the non-linear combination of task variable influences on

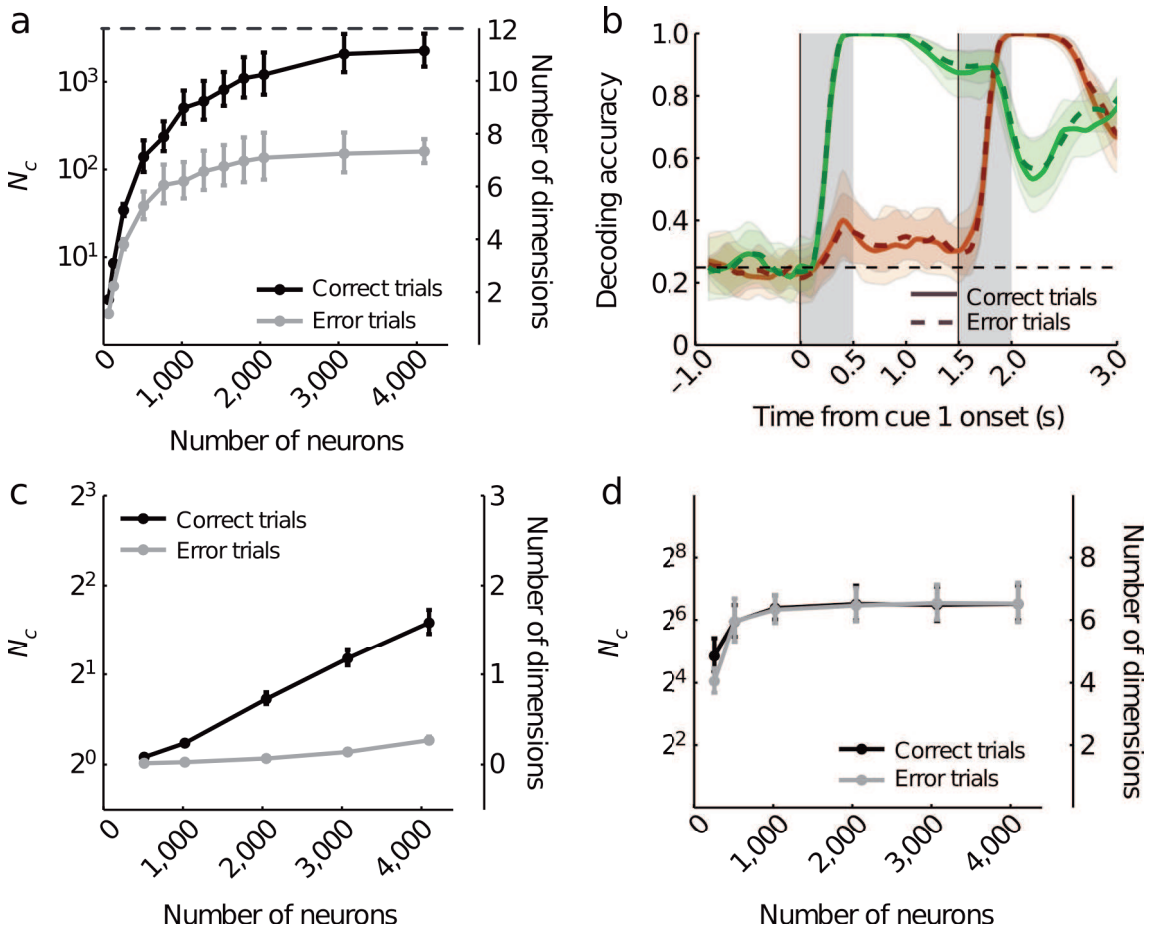

Figure 1.3: Monkey performance drop in a recall task correlate with non-linear mixed selectivity decrease (Figure from Rigotti et al. (2013)). *Activity of PFC neurons was recorded while monkeys performed a recognition or recall task. The animals had to remember the identity and order of two objects presented on a screen. In graph* ***a****,* ***c*** *and* ***d****, left y axis is the number of implementable classifier that reflect the dimensionality of the data shown in right y axis.* ***a****. The dimensionality of population activity is lower when monkeys make errors.* ***b****. However, cue identity can be decoded as accurately when monkeys make errors (dashed lines) as when they perform correctly (continuous lines). Each grey area corresponds to the presentation of one cue. Therefore, errors are not related to degraded representation of the cues. Green and orange lines are the decoding accuracy for the first and second cue, respectively.* ***c****. The non-linear component of mixed displays a drop in dimensionality when monkeys make errors.* ***d****. However, the linear component shows a similar number of dimensions with and without errors, implying that the presence of the non-linear component is correlated with correct behavior.*

single-neuron activity. These non-linear combinations can be quite complex and more difficult to interpret than, for example, the cue-response interaction shown in fig. 1.2d.

1.2.3 Sequence Processing and Single Unit Correlates

So far, we focused on the static mapping between stimuli and responses, however any sequence processing and production requires temporal information to be taken into account. Indeed, most complex behaviors require temporal organization, from mere preparation of a coffee to complex language comprehension and production. This section will briefly introduce neural activity correlates of temporal and sequential representations.

Pioneering work from Barone and Joseph (1989) revealed how stimulus-related activity of PFC neurons was influenced by previous stimuli when presented in a sequence. They recorded single neurons from the dorsolateral PFC of monkeys that observed sequences of targets on a screen. Animals had to reproduce these sequences by touching the targets in the same order. They found neurons responsive for the rank of a particular target during the presentation of the sequence. Also, they described neurons responding for a fixated target only when it followed or preceded a specific target. They named these neurons *context cells*, as they responded for a target only in a specific context. These activities are reminiscent of the interaction between task variables. But in this case, the interaction involves the temporal information related to the order of presentation of the targets. Similarly, in a delayed sequential reaching task, Funahashi et al. (1997) found that 32 of the 72 neurons that responded during a delay were selective for a target only when presented in a specific order of the sequence. This type of contextual selectivity has also been found in the supplementary and presupplementary motor areas as well (Clower and Alexander, 1998). In an experiment where monkeys had to perform cued motor sequences, Shima et al. (2007) report neurons whose activity corresponded to the category of the sequence to be produced. A set of three possible movements (A, B, C) composed 4-element sequences. For example, category neurons were responsive to movement sequences A-A-B-B and C-C-A-A.

We see with these results that the representation of temporal information is embedded in the representation of stimuli. This presuppose that the temporal dimension related to sequential behavior may not be processed in parallel to non-temporal information. Rather, it appears to be distributed and embedded within the non-temporal representations. This point is developed in chapter III.

1.2.4 Neural Activity Dynamics

The neural activity dynamics underlying context representation is currently actively explored. These studies find insights in dynamical system theory to explain the population activity dynamics of cortical neuron networks. Along this approach, cognitive functions may be carried out by specific dynamics. For example, a popular view is that

persistent activity in working memory task is maintained thanks to attractors of the neural dynamics. In the case of neural activity, an *attractor* is a state of the network that is relatively stable because it attracts all neighboring states to itself. The state of a neural network is defined as the set of activity of each neuron in the network, and the ensemble formed by all possible combinations of these states is called the *state space*.

Recent studies suggest that rules and context are maintained thanks to attractors. Durstewitz et al. (2010) show an abrupt transition in the neural population activity between two rules that occupy distinct region of the state space. They found that population activity was rather stable after the rules were learned, suggesting that both rules were maintained in attractors. In-depth analysis from (Balaguer-Ballester et al., 2011) showed that the successive states of a delayed win-shift task[2] are represented in successive semi-stable and converging activities of rats medial PFC neurons. They observed "cognitive-epoch-specific neural ensemble states" which suggest that the neural dynamics jumps between successive attractors. Each stable state may represent the current cognitive state related to an epoch of the task.

The opposite dynamical regime to attractors is transient dynamics, in which the activity of neural ensembles continuously change. Studies of odor representation in locust antennal lobe (Mazor and Laurent, 2005) and mammalian olfactory bulb (Bathellier et al., 2008) revealed that different odors can be discriminated in the transient dynamics that follow odor presentation. In an attempt to verify predictions of recurrent dynamics (that we will address in the third chapter), Nikolić et al. (2009) showed that primary visual cortex neurons showed a contextual encoding of elements presented in a sequence. These results are reminiscent of the findings of Barone and Joseph (1989). Nikolic and colleagues argued that sequences of visual elements are encoded in spatio-temporal patterns of activity, in other words, in transient activities.

While attracting dynamics have received a lot of attention, these last results have led to the proposal of "transient dynamics for neural processing" (Rabinovich et al., 2008; Durstewitz and Deco, 2008) or spatio-temporal processing (Buonomano and Maass, 2009). Both groups of researchers support transient dynamics as behaviorally and computationally relevant in many situations. Rabinovich et al. (2008) suggest that neural systems may best be described by transient dynamics that link successive attractor states.

[2] In a win-shift task, a different choice must be made at each trial because the same choice is not rewarded in two consecutive trials.

1.3 The Need for Arbitrary Contingency Representations

A wealth of research findings has uncovered the fundamental mechanisms behind learning in primates. Most of learning models rely on the prior representation of information relevant to adaptation before any link between a contingency and a behavior is created. In addition, in this chapter we have seen that activities in the PFC correlate with the necessary representations of contingencies. However, the origin of these representations is poorly understood. In this section, we will present the bases that lead to our hypothesis on the process that generate such representations in the activity of PFC neurons. But first, we will present the behavioral theory of selectionism that develop ideas similar to our hypothesis at the level of behavior.

1.3.1 Complex Behavior Through Selectionism

In order to explain the diversity and complexity of behaviors, Thorndike put forward ideas of selectionism(Donahoe, 1999), a theory which can be described as a three-step process: variation, selection and retention (Dennett, 1995) (fig. 1.4). He argued that the complexity of behavioral repertoires can be explained by this process.

Variation is undirected and provides the raw material upon which selection operates, and is the source of novelty in the selectionist process. Selection is the mechanism by which a variation is favored by the environment. Finally, retention refers to the maintenance of the selected variations. Maintained variations subsequently contribute to variation through an accumulation process. Repetition of these processes eventually provide complex behaviors. This description is very general. In the context of psychology it must be understood in terms of behavior genesis. In other words, variation refers to the undirected generation of behaviors, and selection and retention to the strengthening and consolidation of a relevant behavior. Much later, Skinner, a strong proponent of behaviorism, developed similar ideas which focused on the selection aspects. He based his theory of *selection by consequences* on his work on operant conditioning Skinner (1981).

1.3.2 Selection of Pertinent Representations Among a Preexisting Set as a Mechanism to Adapt

Although the selectionist theory may be viewed as limited in its explanation of behavior, it echoes the hypothesis developed in this thesis. We are particularly interested

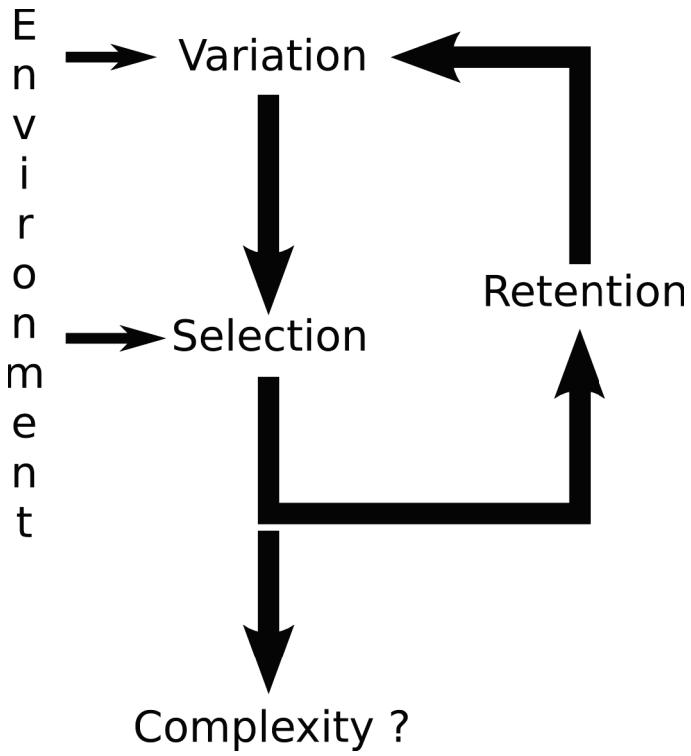

Figure 1.4: Selectionism (figure reproduced from Donahoe (1999)). *According to selectionism, complexity arise from simple processes (variation, selection and retention) that engage in a cyclic global process. It can be interpreted as an explanation of the emergence of complex behavior through the cumulative effect of reinforcement.*

in the variation process, whereby undirected "raw material" is generated. This raw material is what selection acts upon. We want to transpose this concept to the generation of representations in the prefrontal cortex (PFC). We present evidence and argue that PFC represents all relevant contingencies of a given task in its distributed activity. We propose that this property is inherent to the structure of the cortex. Finally, we argue that these representations are present before any learning takes place in PFC, and that they are the necessary precursors of prefrontal learning.

Such representations are crucial for learning, because any behavior that is not innate must be learned. Associative and reinforcement learning involve the creation of links between two or more representations. These representations can refer to inner states or pertain to the sensory or motor domain. Yet, in order to be linked, any two representations must exist in a context amenable to learning. For example, to associate any stimulus with a behavior, the representation of the stimulus must exist before the association can be made. Of course, representations may be altered with learning, for example strengthening them when they lead to relevant behaviors. So specific activity appearing with learning would be the reflection of the already existing representations

that have been shaped by learning.

So a key question is how the pattern of activity that underlies these representations can arise. This may be explained by the same iterative process proposed in selectionism. At any given start point, for example before an adaptation in behavior, or in a laboratory setting before the start of a task, complex combinations of sensory, motor and already learned internal representations preexists as complex activity patterns of prefrontal neurons. Specific activity patterns, or combinations of patterns, are selected and learned because they represent contingencies that lead to relevant behavior. Learning strengthens these patterns, which can in turn participate in a new set of more complex combinations to be subsequently selected and strengthened, and hence the iterative process continues. We must stress that the theory developed here differs from neural Darwinism (Edelman, 1987), which involves the duplication of neural groups that have been selected.

A striking example illustrating this theory in the auditory system lies within the development of language. Kuhl et al. (1992) demonstrated that prior to a certain age, babies can discriminate vowels within the full vowel space of human speech. Through exposure to a particular language, they gradually specialize to recognize the vowels of this language. This implies that, before learning their native language, babies potentially possess universal representation of the vowels produced by humans, yet this variation process undergoes selection in order to refine the vowel space given initial use.

The variation process that generates any arbitrary combination of already existing representations is the topic of this thesis. In the remaining chapters of the state of the art section, we will refine the hypothesis that the prefrontal cortex represents all contingencies in its distributed activity before any learning takes place, and that this property is inherent to the structure of the cortex. Learning would be the mechanism by which representations relevant to behavior are strengthened and subsequently used in combination with other representations to eventually produce complex adaptive behavior.

Chapter 2

Representations in Neural Networks

2.1 Neural Networks: Substrate to cognition

The origins of connectionism

In the previous chapter, we developed the argument that adaptation is based on selection and learning of pre-existent representations. In this chapter we will first introduce neural networks as a model to understand how the brain underlie these representations. Then, we will discuss how arbitrary combinations of sensory, motor inputs and internal representations can arise from inherent properties of neural networks.

Throughout the state of the art section of this thesis, we will use the term *neural network* to refer to artificial versions of biological neural networks for modelling purposes. The larger philosophical paradigm behind neural networks is refereed to as connectionism. The first part of this chapter will be dedicated to the history of the emergence of connectionism, because of its relevance to the understanding of information representation in neural systems. It is especially interesting in the context of the universal representation problem. As we will see, lack of representational power in early neural networks hindered research in the field for more than a decade, shortly after their introduction.

After defining and setting the philosophical and psychological origins of the connectionist paradigm, we will look into the first neural-network models and explain the reason for their temporary demise. We will then introduce neural networks that can learn to perform non-linear problems followed by attracting dynamics in simple recurrent neural networks. Finally, we will show the benefits of randomly generated connections for universal representations.

2.1.1 Definition

Connectionism, like cognitivism, is a theory of information processing. However, as we will see, connectionism is based on sub-symbolic representations and processing, in opposition to traditional paradigms like *symbolic logic*. This latter approach is derived from mathematical reasoning and involves manipulating specific and discrete symbols with explicit and logical rules organized in hierarchy (Medler, 1998). In contrast, connectionism is based on sub-symbolic, distributed and statistical representations and processing, where each processing unit does not necessarily has a meaning. Furthermore, the former approach operates on symbols sequentially, whereas connectionism introduces parallel processing of information.

In order to avoid any confusion, here we are interested in connectionism related to *cognitive science*, whose main objective is to understand the human mind. Neural networks can be approached from an engineering perspective, where the goal is to develop systems that are efficient but are not necessarily inspired by the brain. Nowadays, connectionism, among which neural network is the main implementation, is well integrated in the meta field of cognitive science. This multidisciplinary approach has had a strong influence on the understanding of human mind because of the convergence of its encompassed fields towards a more unified theory of human mind. However, as cognitive science emerged in the 1950's, the most advanced research on artificial intelligence was dominated by symbolic logic, while psychology saw the advent of *cognitivism* that proposes similar symbolic processing through successive mental states. Proponents of both these approaches seemed to disapprove the emergent connectionism which developed radically different theories. Indeed, connectionism explained information processes at a lower level than their respective fields, which seemed to contradict their own theories.

In 1986, in their book entitled *Parallel Distributed Processing*, Rumelhart, McClelland and the PDP research group proposed a definition of the connectionist approach, and more specifically, of neural networks. They list eight properties as essential to the paradigm (Rumelhart et al., 1986a):

- A *set of processing units*
- A *state of activation*
- An *output function* for each unit
- A *pattern of connectivity* among units
- A *propagation rule* for propagating patterns of activities through the network of connectivities

- An *activation rule* for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit
- A *learning rule* whereby patterns of connectivity are modified by experience
- An *environment* within which the system must operate

In other words, a neural network is composed of a structured set of units endowed with processing properties that statistically learn to compute inputs. This processing paradigm is denoted as distributed and parallel because its computational power is derived from the joint operation of each individual unit. Connectionists base their models upon the building block of the brain, the neuron, and the functional properties of its neurophysiology which they believe are critical to understand cognition (Medler, 1998). This new modelling approach accounts for properties of human cognition poorly handled by classical paradigms, like *graceful degradation*, *content-addressable memory* and *supervised learning*. We will define these properties in their respective context in the following sections.

2.1.2 Philosophical, Psychological and Neurophysiological Roots of Connectionism

Connectionism takes its roots in *associationism*, that has philosophical origins that can be traced back to Aristotle in ancient Greece. He described memory as composed of simple elements linked together by temporal succession, object similarity, and spatial proximity. Indeed, this proposition matches very well with the description of psychological processes developed by proponents of materialism and empiricism (including Locke and Hume) under the paradigm of associationism (Medler, 1998). The list of concepts defined by Bechtel and Abrahamsen (1991) will help us to define the associationist paradigm:

- mental elements or ideas become associated with one another through experience,
- experience consists of such things as spatial contiguity, temporal contiguity, similarity, and dissimilarity of ideas,
- complex ideas can be reduced to a set of simple ideas,
- simple ideas are sensations, and
- simple additive rules are sufficient to predict complex ideas composed from simple ideas

Before behaviorism became the leading paradigm in the field and occluded any theory on the mechanisms at work in the brain, early psychology theories included connectionist ideas that arose with the combination of associationism and neurology. Indeed, Spencer (1855) and James (1890) (father of the associative memory theory) developed very similar theories stating that links between neural processes representing events in the world are strengthened in the brain to reflect the co-occurrence of these events in the world.

Thorndike, a student of James, can be considered as one of the founding behaviorists through experiments like the puzzle box, and is certainly one of the first true proponent of connectionism, developing the concept of sub-symbolic neural association. He insisted on the demarcation of connectionism from associationism (Thorndike, 1932) through the development of what would later be known as distributed processing, hidden units and supervised learning. Indeed, he developed the theory of *Law of Effect*, which is a precursor of the reinforcement learning paradigm (Thorndike, 1898).

Finally on our brief psychological account of the origins of connectionism, Hull (1943) is of particular interest because of his insightful conjecture on neural activity. Specifically, he proposed the idea of "interactions between two or more afferent neural impulses which implies that behavior to the same stimulus is not constant" (Medler, 1998). This is strongly reminiscent of the interaction between task variables likely to produce mixed selectivity in neural activity. Furthermore, and critical for connectionism, he developed a theory of learning that is echoed in the *Rescorla-Wagner rule*, and which has been demonstrated to be identical to the *Widrow-Hoff rule* for training early neural networks.

One must briefly mention the life-long work of Lashley (1950) on lesion experiments he summarized with the *Mass Action principle*: "the reduction in learning is proportional to the amount of tissue destroyed, and the more complex the learning task, the more disruptive lesions are." This observation is related to the concept of *graceful degradation* or *fault tolerance*, which is the progressive functional impairment of a system in the event that one of its component is failing. Neural networks show similar progressive performance degradation as connections within the model are severed.

Early accounts of connectionism in neurophysiology include the cornerstone work of Hebb (1949) on *synaptic efficiency*. The principle states that if a first neuron repeatedly contribute to the firing of a second neuron, the synapses that link them become more efficient. This rule has been derived mathematically and is used widely in realistic neural network models to modify weights between connections. This simple learning method, referred to as Hebbian learning, has had a strong impact on the explanation of associative processes in the brain.

2.1.3 Early neural network models and their limitations

The mathematical foundations of neural modeling can be attributed to McCulloch and Pitts (1943). In their seminal paper *A logical calculus of the ideas immanent in nervous activity*, they simplify the activity of biological neurons into five functional states:

1. The activity of the neuron is an "all-or-none" process.
2. A certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron.
3. The only significant delay within the nervous system is synaptic delay.
4. The activity of an inhibitory synapse absolutely prevents excitation of the neuron at that time.
5. The structure of the net does not change with time.

Networks are composed with basic elements referred to as neurons or units. A connection between two units is directed and weighted, meaning that the activation of the first unit will excite the second unit proportionally to the value of the connection weight.

This effort at reducing the property of neural activity set the basis for neural modeling. Less than a decade later, Hodgkin and Huxley (1952) built a neuron model that incorporated much more detailed electrophysiology. Both versions simulate the spiking behavior of neurons. Less detailed models simulate the firing rate of the neurons. In this case, a neuron is represented by two variables, its *activation state* x and its *output* y. The activation state of neuron j at time t is obtained through the weighted contribution (*weights* denoted w) of afferent neurons (inputs u):

$$x_j(t) = \sum_{i=1}^{m} (w_{ij} u_i(t))$$

The output is derived from the application of the *transfer function* [1] $\varphi()$ on the activation state:

$$y_j(t) = \varphi(x_j(t))$$

The models developed by the author use firing rate neurons (also denoted analog neurons), as a consequence, we will not get into the details of the wide literature on its spiking counterparts.

[1] Also denoted activation function.

Importantly, McCulloch and Pitts showed that networks of such simple processing units can represent any propositional logic statement, and that these networks potentially have the computational power of a *Universal Turing Machine*. A *Turing Machine* is an hypothetical system that can manipulate symbols in sequence with logical rules. It can be seen as a machine that sequentially process information according to symbolic logic. Conversely, the Universal Turing Machine is the system that can simulate any Turing Machine. In other words, a machine that can potentially implement all possible logical operations. In principle (if the connections of the network are well set), a neural network can reproduce any logic implemented in a Turing Machine, and therefore, compute the full extent of operations developed by symbolic logic.

Based on these principles, Rosenblatt (1958) developed the Perceptron, which laid the bases of the canonical feedforward network architecture (fig. 2.1).

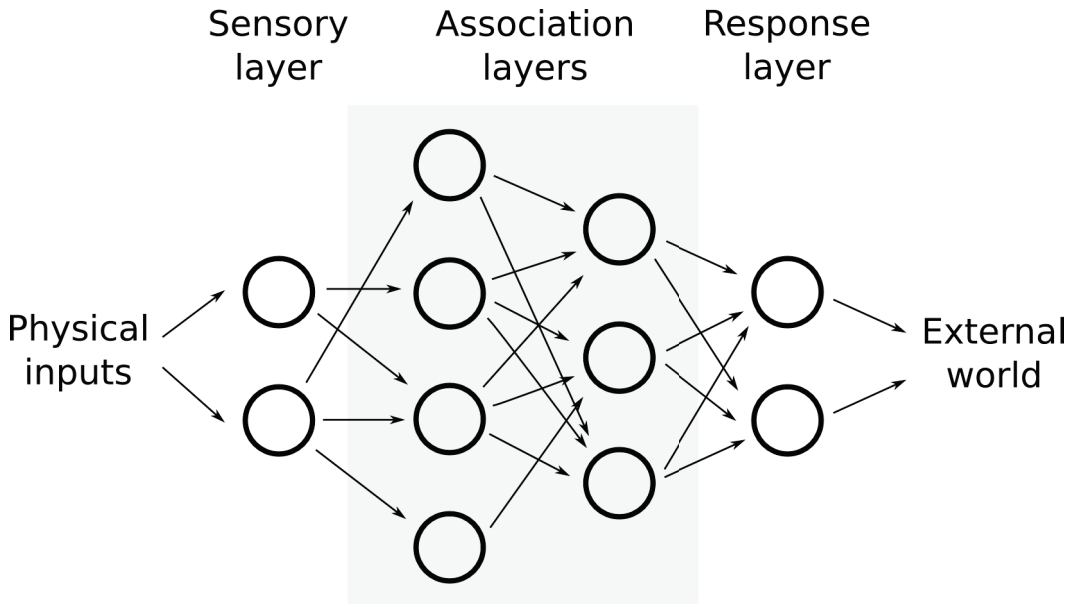

Figure 2.1: Perceptron architecture. *Physical properties of the external world are transduced by the sensory units which pass on their activation to association units. Through successive association layers, activation finally reaches the response layer that connects back to the external world. Activation flows in one way only, from sensory to response units, and equally important, without connection between units within a layer. Note that the conventional terminology refers to these units as input, hidden and output units.*

Neural network models define their architecture with layers, which are pool of neurons. In the Perceptron architecture or feedforward network, layers are connected serially. Units from one layer receive activation through directed connections from the previous layer, and send connections to the next layer. Also, a defining property of this architecture is that there are no connections between neurons within a layer. The first

layer of the network is called the input layer (bottom layer) and receives the information to be processed. The last layer is the output layer (top layer), which provides the answer, response or output of the network depending on the terminology. In between, hidden layers process the inputs received the bottom layer through successive activation of the layers until reaching the top layer. Depending on the type of neurons implemented each layer produce a non-linear transformation. However, if one considers the case of a linear activation function, connections between two layers perform a linear regression. And in this case, it is the succession of linear transformations that elicit non-linear processing of inputs. Rosenblatt demonstrated that this architecture is able to solve the exclusive OR problem (XOR, or exclusive disjunction) through non-linear processing of the inputs. The XOR is a logical operation that returns true when only one of two inputs is true.

This architecture was developed for pattern recognition purposes, yet "the Perceptron, more precisely, the theory of statistical separability, seems to come closer to meeting the requirements of a functional explanation of the nervous system than any system previously proposed" (Rosenblatt, 1958). Connections between the input and hidden layers, and between hidden layers are fixed. Learning takes places as a modification of the weights of the connections between the last hidden layer and the output layer. The architecture, the type of neuron model and the learning method of a network constitute the essential components of its description. However, one must keep in mind that they have to be considered independently. For example, different learning methods can sometimes operate on the same architecture.

Indeed, it is the work of Widrow and Hoff (1960) on the Adaline that seeded the established learning rule for training feedforward networks, known as the *Delta Rule*. Its denomination comes from the modification of connection weights that follows successive small quantitative steps denoted "delta". This method pertain to the *supervised learning* class because the network is trained with a specific *desired output* for each input. The gap between the actual output of the network and its desried output is referred to as the error and is used to modify the weights. This first version was limited to training the connections between the last hidden layer and the output layer, like the rule developed by Rosenblatt for the Perceptron.

However, Minsky and Papert (1969) who were also interested in visual pattern recognition, analyzed carefully the capabilities of Perceptrons. Because these networks can only learn the connections between the hidden layer and the output layer, they showed in their work entitled *Perceptrons* that these networks could not technically "learn" to perform a XOR classification (Fig. 2.2). In the case of the original Perceptron, the weights between the input layer and the hidden layer must be hand tuned and cannot be learned. Yet, the XOR classification is the most simple linearly inseparable problem.

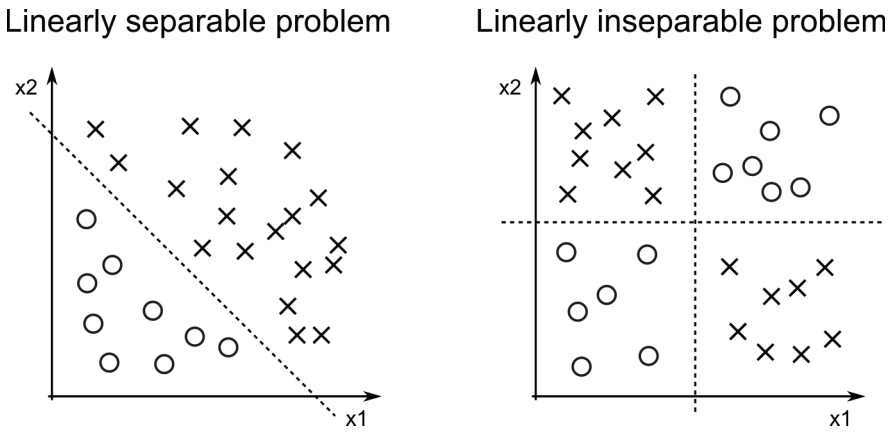

Figure 2.2: Linear separability of a problem. *This figure presents two example "problems" where elements defined by two variables (that correspond to input neurons in the case of the perceptron) must be linearly separated according to their class (x or o). In the left hand graph one can see that the x and o elements can be separated by a single line (dashed line). However, the problem presented on the right hand graph cannot be separated with a single line, and is therefore non-linear. This particular problem is reminiscent of the exclusive OR (XOR) operation, a basic logical operation. A Perceptron can learn to separate the first problem but not the second.*

If the perceptron cannot learn to solve it, what can be expected for more complex problems, Minksy and Papert asked. Their work was directed towards an assessment of the capacity of the Perceptron, yet, their results were understood as a serious limitation of neural networks by the scientific community, and set the demise of neural networks.

Following the downfall of behaviorism popularity and the simultaneous rise of cognitivism and symbolic processing, connectionist research disappeared from psychological literature until the mid 1980's (Medler, 1998).

2.2 Resurgence of Connectionism:

Learning Non-Linear Representations

Connectionism has had difficulties imposing itself as an interesting paradigm to explain cognitive processes, as it developed in apparent opposition to the symbolic logic (Newell and Simon, 1976). It was the dominant approach in the modeling of human intelligence at that time. Furthermore, cognitivist approach itself was based on the manipulation of symbols or mental states.

Early neural networks showed that non-linear problems could be solved by feed-forward networks with a hidden layer. However, connection between input and hidden layers had to be hand tuned. Although neural network research almost disappeared after

the demonstration of the limitations of Perceptrons, some research actively contributed to the resurgence of the field through the development of learning methods able to train the connections at every level of a feedforward network.

2.2.1 Parallel Distributed Processing and Error Back-Propagation

Independently discovered by several researchers (Werbos, 1974; Parker, 1982; Le Cun, 1986), the *Generalized Delta Rule* (GDR) was popularized by researchers who referred to themselves as the Parallel Distributed Processing (PDP) group. In a book rightfully named *Parallel Distributed Processing* they generalized the GDR, introducing the *error back-propagation* algorithm (or simply *backpropagation*) (Rumelhart et al., 1986c).

The name stems from the process by which error is propagated from the output layer towards the input layer, in the direction opposite to the activation of the network. The error is defined as the discrepancy between the desired output and the actual output of the network. To first obtain the actual output, the network is activated through the classic propagation of activation in the network (bottom up direction). Inputs first activate the input layer that propagates its activity to the successive layers, to finally reach the output layer whose activation defines the output of the network. The gradient of the error is calculated by comparing the actual output to its desired counterpart. It is then propagated from the output layer connections towards the connections of each successive hidden layer, until reaching the input layer (top down direction). Repeating this method with all the available pairs of input and desired output, for several iterations, allows the weights to be modified in the direction that decreases the error. Eventually, the weights should converge towards a minimum of error, a state where the actual outputs should match the desired output. This method allows the network to successively extract the pertinent features of the inputs for the problem at hand. Each layer adds a degree of precision to the extracted features.

Now able to learn the connections between all layers, the backpropagation algorithm has been shown to learn to solve the XOR problem. Later, Hornik et al. (1989) have analytically demonstrated that a feedforward network with only one hidden layer using backpropagation algorithm is a universal approximator. This implies that it can approximate any continuous mathematical function. The degree of approximation is determined by the number of hidden units. Discontinuous functions are approached with a second hidden layer, but any additional hidden layer should not, in principle, add precision in terms of function approximation. Feedforward networks with the backpropagation rule have taken the name of multilayer perceptron for their ability to learn more than one

layer of connections.

With appropriate neural properties (non-linear activation functions in single neurons), backpropagation converges on weights that elicit non-linear combinations of inputs in the successive layers. Importantly, these activities are sub-symbolic. This means that the activity of each neuron do not necessarily represent symbols as in the symbolic logic paradigm. The connectionist approach proposes mechanisms for observations poorly explained by classical symbolic and cognitive approached like *generalization*, the ability to predict data outside of the original training set, previously mentioned *fault tolerance* and *content-addressable memory* that will we present in the next section. These observations reinforced the idea that the brain might use very different computational mechanisms to represent and process information than those from computers, originally laid down by John von Neuman. However, the same researcher later proposed a redundancy principle (Von Neumann, 1956), reminding of the distributed approach later developed within the symbolic framework by Winograd and Cowan (1963).

From a cognitive modelling approach, the backpropagation method itself does not seem quite biologically plausible. However, this work demonstrated the capacity of neural networks to solve quite complex non-linear problems. Here, we are interested in the demonstration that arbitrary static pattern classification and function approximation can be solved by non-linear networks. Indeed, the key feature of the second generation of neural networks was their non-linear processing capabilities. Even if the learning method itself is not plausible, properties of the network after learning can be of interest to study non-linear distributed representations. For example, Zipser and Andersen (1988) used backpropagation to simulate the activity of posterior parietal cortex neurons of monkeys performing a task with saccadic eye movements towards visual cues. They reproduced in hidden units the interaction between the eye position and saccade response observed in the activity of parietal neurons. Interaction implies that the activity of these neurons is explained by a non-linear combination of the variables, a property referred to as mixed selectivity as was mentioned in the first chapter.

2.2.2 Early Recurrent Networks with Attracting Dynamics

Recurrent neural networks (RNN) are defined by a connectivity pattern that allows connections between neurons within a layer, and between neurons from a higher layer to a lower layer. Since the connections are recurrent, activity will not only flow from the input layer to the output layer, but also propagate in loop and interact with current inputs, thereby adding dynamic properties to the network. A wide variety of architectures have been developed which have added substantial processing abilities to the neural network paradigm.

Recurrent neural networks can be divided in two families with different purposes (Lukoševičius and Jaeger, 2009). Both families are of interest to the present thesis. The temporal processing family will be the focus of the next chapter. The attractor family is briefly introduced in this section while specific models will be covered in the fourth chapter. Some instances of RNN have properties pertaining to both families.

The first instance of attracting RNN was developed by Hopfield (1982). Referred to as *Hopfield net*, this class of networks implement *content-addressable memory*. It is a property of memory function stating that an encoded memory can be retrieved by activating a subset of the components of the memory. The brain would in turn activate the remaining components which would form the whole memory together with the already active components. It can also be considered as a mechanism to generalize, based on the similarity of the inputs to the learned pattern.

Each attractor of the network is trained and represents a pattern or class of inputs. Any input constantly activating the network will make its dynamic converge towards one of the trained attractors. By definition, inputs of one class should be close to the prototypical pattern that was used to train the network. Therefore, the activity produced within the network by one input should converge towards the attractor built from its prototype.

To elicit such dynamics, this type of network assumes full, symmetric connectivity between the units. In other words, each unit is connected with every other units with the same weight in each direction of the connection. This network is the first of a long list of networks that have since been derived from this initial instance. We must mention several extensions that brought this model closer to biology by separating excitatory and inhibitory neurons, which were simulated with plausible spike rate (Durstewitz et al., 2000). Stochastic instances of this type of RNN have been developed by Ackley et al. (1985) under the name *Restricted Boltzmann Machine*, now used in successive layers in *Deep Neural Networks* which constitute the state of the art in machine learning pattern recognition.

Back to recurrent and attracting dynamics, we must mention the *winner-take-all* (WTA) mechanism, often used as a sub-component of a larger architecture. In the prototypical instance, neurons in one layer compete with each other to be active in the detriment of other neurons. This elicits a dynamic that converges toward the activation of only one neuron. A balance between excitation and inhibition must be found to elicit the proper dynamics. In the simplest instance, inhibitory connections between all neurons of the layer is assumed. In more complex versions, competition can be implemented between pools of neurons. Self excitation of neurons, or within the pool of neurons in the related case, can be used to reinforce and sustain the competitive

dynamics. Softer competition can converge towards the co-activation of several neurons.

The richness of WTA dynamics has led to various models of decision making, either implemented as full model dedicated to decision making, like the sophisticated biological model of Wang (2002), or simply to provide a decision mechanism in a larger network (Dominey et al., 1995). The *Kohonen map* (or self organized map, SOM) uses the WTA mechanism along with an *unsupervised learning* rule to implement converging maps reducing the dimensionality of inputs (Kohonen, 1982). All the learning methods described so far are said to be supervised, because the objective of the training is to modify the weights to produce an output that converges towards a defined target output. Thus learning is carried out with pairs of input and desired output. In contrast, methods that rely on algorithms that learn the statistical relationships between inputs are of the unsupervised class, like the Kohonen map.

2.3 Distributed Non-Linear Representation and The Cortex

2.3.1 The Necessity for Complex Activity in PFC

The history of neural networks has shown how representational power of networks is a key issue. Networks must be able to represent complex non-linear combinations of the inputs in order to solve arbitrary problems, conferring them the much pursued and allegorical Universal Turing Machine capacity. Indeed, how can a network learn complex outputs if it is incapable of representing complex contingencies? Hence, the representational power of a neural network is one of its fundamental properties. On a similar tone, Minsky and Papert in their prologue to the 1988 edition of Perceptrons stated that "no machine can learn to recognize X unless it possess, at least potentially some scheme for representing X".

In addition, connectionism is born through the gradual implementation of sub-symbolic information representation. We want to insist on that aspect of neural networks as it is the key point in understanding the complex activity observed in models, and, by means of projection, in the recordings of cortical neurons. Indeed, universal representational power is conditional to the non-linear and sub-symbolic property of single-unit activity, as suggested by the work of Rigotti et al. (2013) mentioned in the first chapter. Therefore, we argue that the non-linear activities observed in the activity of prefrontal cortex (PFC) neurons is an inherent part of their representational power, and must be considered as essential for any adaptive behavior to occur. We argue that the explicit representation of complex contingencies in the activity of PFC neurons is the result of a

selection through learning of distributed, sub-symbolic and non-linear combinations of sensory, motor and internal variables.

Now that we have posited the fundamental properties of the representational power of the cortex, the logical question that follows is how does the cortex to provide such representations.

2.3.2 Random Connectivity as a Simple Mechanism for Universal Spatial Representations

As explained above, the second generation of feedforward networks – the multilayer perceptron (MLP) – can learn to produce these non-linear combinations. However, the plausibility of backpropagation being implemented in biological networks is rather low. Consequently, other solutions must be envisioned in order to understand how the cortex might generate such activities. In this respect, a mechanism as simple as the generation of random connections is used in several networks. Indeed, random connections inherently elicit a wide variety of activities. Before presenting networks with fixed random connectivity, we should mention that prior training several types of neural network require random generation of connections.

Indeed, it is necessary for MLPs to generate randomly the weights between layers before training in order to break the detrimental effect of weight symmetry on gradient descent. Also, one can view the learning process as a refinement of the non-linear activities produced by random generation of the weights. Interestingly, it turns out that the initial set of weights can have a strong influence on the performance of a MPL, as some instances converge towards local minimum of errors[2] which can be overcome with more sophisticated backpropagation algorithm. Similarly, Kohonen maps and other networks require random generation of the connections between the input layer and map to produce a variety of responses, learning consist in tuning the already generated weights. In the following paragraphs, we will address a quite different approach, one that uses random generation of weights which are not trained. The following study illustrates the power of this method.

In a task reminding of the WCST, Rigotti et al. (2010b) demonstrated that non-linear combination of task rule and feedback were required in order to perform the task correctly. This work is reminiscent of the problem of implementing a XOR operation in simple networks, and can be considered as its cognitive counterpart. The implemented task involved two distinct rules related to the features of visual stimuli. In short, visual stimuli are composed of two features comprising a shape and a color. If the color rule

[2] The network is stuck in a local minimum of error when the training procedure converge to set of weights that does not elicit the minimum possible error.

is in effect, the color of a cue appearing on a screen defines the correct color for this trial. Amongst two stimuli presented after the cue, the network performing the task must choose the one that matches its color. Similarly, when shape rule is active, the network must choose the stimulus that matches the shape of the cue. Feedback is provided to the network through an error signal that is sent to the network when the last choice is incorrect. Figure 2.3 demonstrates that non-linear combination (termed mixed selectivity here) of rule and feedback is necessary to perform this task.

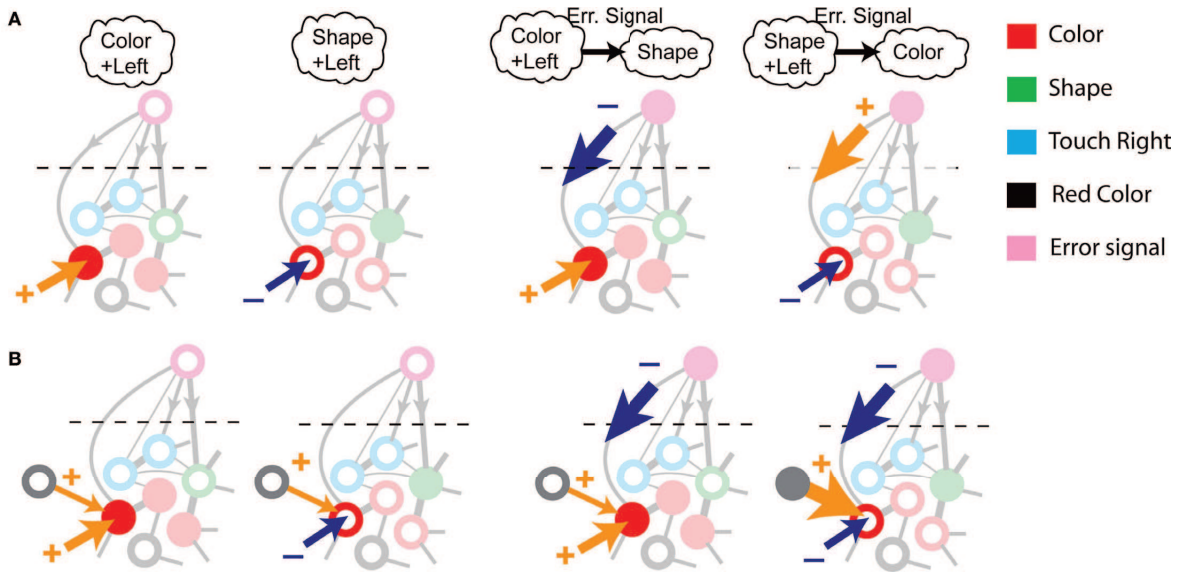

Figure 2.3: The need for mixed selectivity in a WCST model (Figure and caption from Rigotti et al. (2010b)). *Circles represent neurons, and colors denote their response preferences (e.g., red units respond when Color Rule is in effect). Filled circles are active neurons and black lines are synaptic connections.* ***A****. Impossibility of implementing a context-dependent task in the absence of mixed selectivity neurons. We focus on one neuron encoding Color Rule (red). In the attractors (two panels on the left), the total recurrent synaptic current (arrow) should be excitatory when the Color Rule neuron is active, inhibitory otherwise. In case of rule switching (two panels on the right), generated by the Error Signal neuron (pink), there is a problem as the same external input should be inhibitory (dark blue) when starting from Color Rule and excitatory (orange) otherwise.* ***B****. The effect of an additional neuron with mixed selectivity that responds to the Error Signal only when starting from Shape Rule. Its activity does not affect the attractors (two panels on the left), but it excites Color Rule neurons when switching from Shape Rule upon an Error Signal. In the presence of the mixed selectivity neurons, the current generated by the Error Signal can be chosen to be consistently inhibitory.*

In order to solve this problem, they simply use a layer of randomly connected neurons (RCN) that acts like the hidden layer of multilayer feedforward networks (Fig. 2.4). This RCN layer provides the mixed selectivity required to represent any combination between inputs and internal states. The mere random-fixed connections with input and internal

units is able to potentially produce any non-linear combination, in other words, any contingency of the task.

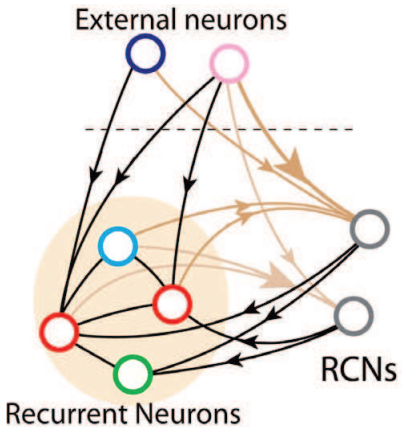

Figure 2.4: Random connections for arbitrary contingency representation (Figure from Rigotti et al. (2010b)). *The blue and pink neurons activate to represent states of the external word, while recurrent neurons (red, green, light blue) persistent activity represent internal states. Randomly connected neurons (RCN) recombine these external and internal states to produce arbitrary combinations that represent different contingencies.*

This solution fulfils the desired function of representing arbitrary combinations. In addition, the authors state that the number of units in the RCN layer, that is required to produce the representations essential to the task, is not much higher than the minimum number of neurons needed if the neural circuit was specifically designed to create these representations. The substantial advantages of an RCN layer are that it does not need to be trained, and that non-linear combinations produced by the RCN layer should be very rich. The RCN activity potentially include every possible combination of inputs and internal states, therefore arguing in favor of a simple mechanism for universal representation. This property entails the possibility to use the same RCN layer to perform different tasks, in agreement with the principle of parsimony.

In addition, the rationale behind the RCN approach includes the biological plausibility of such a network. In the Rigotti et al. model, learning takes places in the connections directed to the recurrent neurons, representing the internal states and outputs of the network. Internal representations are learned with a method related to hebbian learning which has been confirmed by electrophysiological experimentation. Furthermore, because connections to the RCN are generated prior learning, the richness of the representations is present before any learning takes place. If this is indeed the computational principle implemented in the cortex, Rigotti et al. state that one should observe mixed selectivity pattern in the activity of cortical neurons prior any learning.

In parallel to the selectionist view described in the first chapter, a possible mechanism

to obtain representations of increasing complexity would be through the learning of combinations present in the RCN units. Relevant combinations would be strengthened through learning and then fed back to the RCN layer in order to contribute in new combinations. In turn, these new combinations could be selected, learned and fed back, etc.

Chapter 3

Adding Temporal Information to Neural Networks Representations

Previous sections on neural network models focused on the processing of static inputs. If all the information necessary to perform the desired computation is present in the current inputs or stable internal states (as in the case of Rigotti et al. (2010b)) the processing is said to be purely spatial. However, if the order or timing of the inputs to the network is to be considered, these networks will be inefficient as they cannot process the temporal dimension. Therefore, these networks come up short at modeling any cognitive function that involves time. The temporal order of stimuli may be important in many situations, virtually in any situation that requires processing or producing sequences (Elman, 1990).

This chapter first introduces processing of temporal information in recurrent neural networks (RNN) thanks to reverberating activity, and exposes the inherent issue of training such networks. We will briefly present historical methods that have been developed to circumvent this problem and we will finally dwell on the reservoir computing paradigm, the method employed in this thesis.

3.1 Introduction to Temporal Recurrent Networks

3.1.1 Adding Temporal Information Through Recurrence

As a reminder, RNN are defined as such based on an architectural feature: connections between neurons within a layer and/or connections between a higher layer with a lower layer in a classical feedforward bottom-up architecture. Recurrent connections allow for activity of inputs fed in the network to reverberate, thanks to the existence of loops in the structure of the network. Because of this capacity, this type of network has

dynamical properties which can be approached by dynamical system theory.

As highlighted in the previous chapter, RNNs can be roughly divided in two categories based on their dynamics. Networks that display converging dynamics, representing information in attractors of the neural dynamics, and networks that display transient dynamics. The latter networks represent information in trajectories in the neural activity state space[1]. Because of their purpose oriented towards temporal information processing, we will refer to them as *temporal recurrent network*[2] (TRN). The attractor category is best known with the aforementioned Hopfield network (Hopfield, 1982) and restricted Boltzmann machine (Hinton and Sejnowski, 1986). The second type of network will be the subject of this chapter.

TRNs rely on the interaction of current inputs with the reverberatory activity elicited by previous inputs. This means that the representation of an input fed to a TRN will depend on the inputs previously fed to the network, each input driving the activity of the network in a different direction. This is why we often refer to the representation resulting from a sequence of inputs as a trajectory in the state space. Thus, in TRN, it is not the final state of the network activity that is of interest, but the transient activity elicited by the sequence of inputs. These networks process spatio-temporal information, where temporal information is embedded in the spatial representations within the activity of the network. This mechanism provides an implicit representation of time. TRN have been used to process sequential data among which language is well represented (Elman, 1991).

It should be noted that a wide diversity of architecture, sometimes with exotic neural dynamics, have been developed to process temporal information, some of them in feedforward networks, that we will not address here (Haykin, 1999). Contrary to the TRN principle, some of these approaches represent temporal information explicitly as additional spatial dimensions, for example, by adding input units that correspond to previous inputs, thereby "parallelizing time" (Elman, 1990). In contrast, TRN represent time implicitly, in the interaction of inputs separated in time.

3.1.2 The Recurrent-Network Training Problem

All the canonical networks described in the previous chapter come with learning rules that converge towards a supposedly optimal set of weights that minimize the error output of the network. This property of the training methods is quite convenient as

[1] The multi-dimensional space composed by the activity of all network neurons.

[2] Note that *temporal recurrent network* is also the name given to a model by Dominey and Ramus (2000). We will use this term in a broader sens to encompass all recurrent networks that process temporal information.

one can let these networks learn a task, knowing that the training will eventually find a set of weights[3]. Now, the difficulty to train recurrent network is that there is no obvious way or method that systematically converge towards a set of optimal weights. To understand this issue, one must take into consideration that the recurrence of the network poses a problem of recurrence in the propagation of errors. As an example let's consider neurons A and B as reciprocally connected neurons, B being an output neuron. With backpropagation, the error computed between the actual and desired output is propagated from B to A, but if one follows the connections that contribute to the activation of A, the error must be propagated back to B, then again to A, and so on endlessly. This toy example includes only two neurons, but the problem remains the same with a loop constituted by an arbitrary number of neurons. Thus, this chapter describes the methods that have been developed to overcome this issue inherent to TRNs.

3.2 Early Recurrent Networks for Temporal Processing

In this section we will review a few architectures and methods to train TRN that are of interest in introducing the Reservoir Computing paradigm, therefore this account of TRN is clearly not exhaustive.

3.2.1 Simple Recurrent Network

Based on the classical feedforward architecture, the first model of the class we are interested in has been developed by Jordan (1986). To integrate previous outputs with current inputs, he connected the output layer with fixed topographic[4] connections to a state layer that fed back its activity to the hidden layer, thereby creating a closed loop including the output, state and hidden layers. Since the connections to the state layer all have a weight value of 1, and are topographical, the network copies the output into the state layer. Thus, at the next time step, the activity of the output will influence the processing of inputs in the hidden layer. However, connections between the state layer and the hidden layer are fully distributed, like a regular connectivity between input and hidden layers. (Elman, 1990) based his own TRN model named *Simple Recurrent Network* (SRN) on this architecture except that instead of copying the output, it is the

[3] ... if one puts aside the suboptimal solutions and the obvious overfitting issue.

[4] In artificial neural networks, topographic connections usually imply that the sending layer and receiving layer have the same number of units and that each unit of the sending layer connects to its corresponding unit in the receiving layer (i.e., the weight matrix is diagonal).

hidden layer that is copied into what he calls a context layer (Figure 3.1. In fine, the context (or state) layer provides the network with memory.

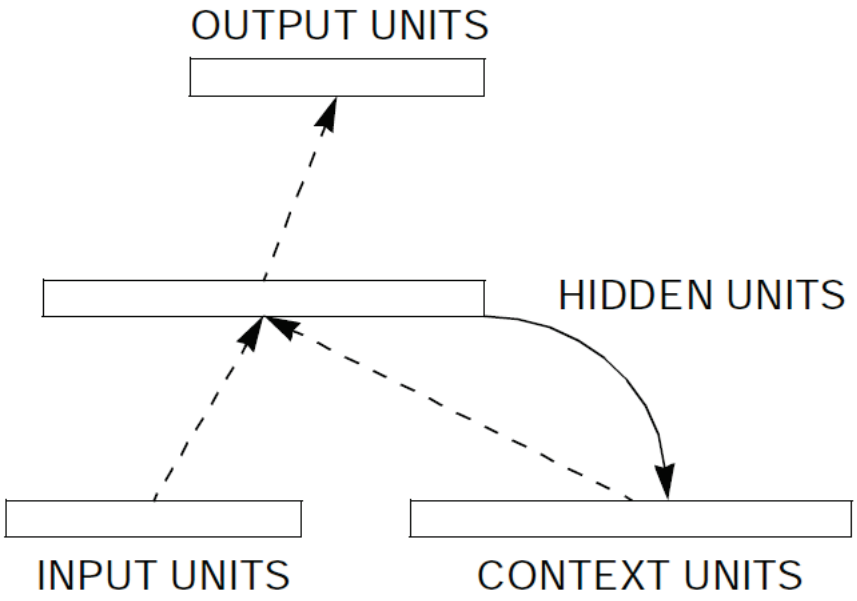

Figure 3.1: Simple Recurrent Network (Figure from Elman (1990)). *Dashed arrows represent trainable connections, the plain arrow represents fixed connections. Context layer receives copies of the hidden layer which recombines with future inputs in the hidden layer.*

Elman and Jordan trained their network as regular feedforward networks. For example, in Elman networks, this implies that the error is propagated from the hidden-to-output connections to the input-to-hidden and context-to-hidden connections. Of course, because of the complex interactions between hidden and context layers and inputs, the convergence of the weights is problematic. The next method is an attempt to extend backpropagation to recurrent networks.

3.2.2 Backpropagation Through Time

Because basic backpropagation was not suited to recurrent networks, many researchers developed independently a method referred to as *backpropagation through time* (BPTT) (Rumelhart et al., 1986b; Robinson and Fallside, 1987; Mozer, 1989; Werbos, 1988). The basic principle of this approach is to copy lagged version of the network for a given number of time step, and operate the classic backpropagation algorithm on it. It can be understood as unfolding time into a spatial representation with which the backpropagation can be applied. BPTT is still one of the most popular method to train recurrent connections (Jaeger, 2005), although, other methods also employing

gradient descent have been proposed like *Real-Time Recurrent Learning* (Williams and Zipser, 1989), *Atiya–Parlos Recurrent Learning* (Atiya and Parlos, 2000) or using different approaches as the *Extended Kalman Filters* (Puskorius and Feldkamp, 1994) and the *Expectation Maximization algorithm* (Ma and Ji, 1998).

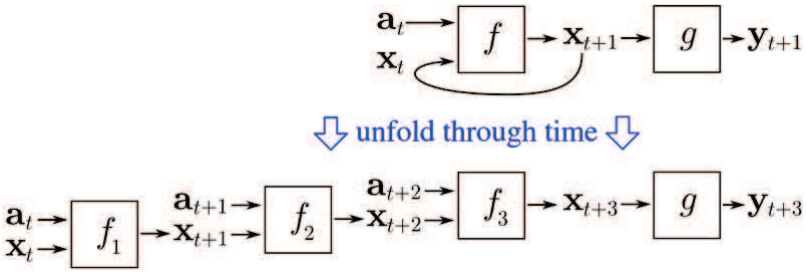

Figure 3.2: Schematic of Back-Propagation Through Time. *Schematic representation of the unfolding process used by BPTT. a_t denotes the inputs at time t, x_t the internal states at time t, y_t the output of the network at time t. f and g are the transformation operated by the input-to-hidden weights and hidden-to-output weights, respectively. The lower panel illustrates the unfolding of three time steps in classic feedforward network.*

Although this algorithm has been shown to be efficient in certain situations, the computational complexity[5] of teaching recurrent connections with such algorithm must be taken into account. Indeed, it is proportional to the squared number of neurons N in the recurrent layer: $O(T*N^2)$ where T is the number of time step back in time that are learned (Jaeger, 2005). Furthermore, classical methods are not guaranteed to converge, as bifurcations of the neural dynamics appear when the weights are modified gradually (Doya, 1992). Other more subtle issues in training RNN with these methods are exposed in Lukoševičius and Jaeger (2009).

3.2.3 Temporal Recurrent Network with Untrained Recurrent Connections

Dominey et al. (1995) developed a model of the cortico-striatal system for the sequential processing of visual stimuli. The network model incorporates a sub-network reminiscent of the SRN architecture. Indeed, the PFC is modeled with two layers recurrently connected. The main PFC layer receives visual inputs and is topographically connected to the second layer denoted as damped PFC layer. This latter layer is endowed with a slower dynamics thanks to longer time constant in their leaky integration, and is connected back with fully distributed random weights (see figure 3.3 for the rest of the architecture). The originality of this model is that the recurrent connections

[5] Computational complexity measures the efficiency of an algorithm in terms of resources.

are not modifiable, learning takes place between the main PFC layer and the output (the striatum in this model) through modification of the weights according to an early form of reinforcement learning. The rationale behind this method is that the dynamics elicited by the recurrent layers is rich enough even without learning and can therefore provide arbitrary representations of the temporal dependency between inputs of a sequence. Next section on Reservoir Computing will give a more detailed description of the benefits of this approach. Note that this model has been demonstrated to process

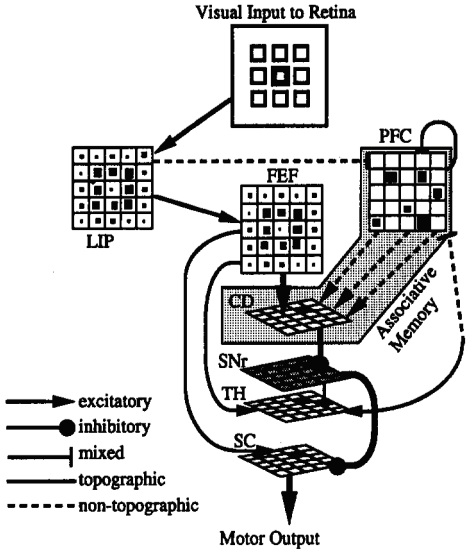

Figure 3.3: Cortico-striatal model for sequence processing (Figure from Dominey et al. (1995)). *This model learns to recognize and reproduce spatial sequences. Spatial information is represented in the lateral intra-parietal (LIP) layer which is connected to a cortico-striato-thalamic loop, and to a prefrontal cortex (PFC) recurrent network. This latter layer is composed of two recurrently connected layers. Connections between the two layers are not modifiable, they intrinsically allow for rich spatio-temporal representations. A readout layer representing the caudate nucleus (CD) receives connections from the PFC module that are modified with reinforcement learning. The caudate nucleaus is part of the cortico-striato-thalamic loop which includes the frontal eye field (FEF), the substantia nigra pars reticulata (SNr) and the thalamus (TH). The superior colliculus layer implement a winner-take-all decision mechanism which leads to a saccade.*

embedded sequences (Dominey, 1995) and was later used for the temporal processing of language (Dominey and Ramus, 2000; Blanc and Dominey, 2003).

Also using an untrained network with randomly generated connections, Buonomano and Merzenich (1995) demonstrated the temporal processing power of a network of spiking neurons endowed with paired-pulse facilitation and inhibitory postsynaptic potentials, two properties observed in cortical neurons.

3.3 Reservoir Computing

3.3.1 History: Convergence of Signal Processing and Neuroscience Modeling

Because training recurrent neural networks is a task that is computationally costly and difficult, a researcher working on neural networks for machine-learning found a very convenient method to teach the recurrent connections (Jaeger, 2001): not training them! Instead, the connections are randomly generated following a set of rules that ensure suitable dynamics in the activity of the recurrent nodes, much like the TRN models developed by Dominey (1995) which can be considered the first prototypes of this new paradigm. A readout layer performing a simple linear regression from the recurrent nodes provides the output of the network (Fig. 3.4). This new approach benefits from the inherent temporal processing abilities of recurrent networks, while using simple training methods of linear networks. Such networks were baptized *Echo State Networks* (ESN) for their reverberating dynamics (echo states) in the recurrent layer. ESNs have demonstrated impressive performances in processing temporal signals (Jaeger and Haas, 2004; Verstraeten et al., 2006; Jaeger et al., 2007a).

Independently from this research, a team, whose main focus is understanding the computational principles of the cortex, developed a surprisingly similar neural network model called *Liquid State Machine* (LSM) for the inherent dynamical states of the main layer, which display ripples of activity elicited by inputs (Maass et al., 2002). While units in Jaeger's work implement analog/continuous output values, the neural network models developed by Maass use more biologically plausible spiking neurons. It is only later that both teams realized they developed very similar networks, which consequently sparked the FP7 European research project Organic (comprising the author's team), whose main focus was the recently baptized paradigm of *Reservoir Computing* (Verstraeten et al., 2007).

3.3.2 Brief Overview of The Reservoir Computing Principles

The description of the reservoir paradigm in the present thesis is largely biased towards the Echo State Network (ESN) developed by Jaeger (2001), as this is the type of network used in the work exposed here. The typical reservoir computing network is comprised of an input layer, an internal/hidden recurrent layer called the reservoir, and a readout layer fully connected to the reservoir units. By metonymy, the full architecture (and the paradigm itself) is often referred to as a reservoir. If one omits the recurrent nature of the hidden layer, the architecture is very similar to a feedforward network

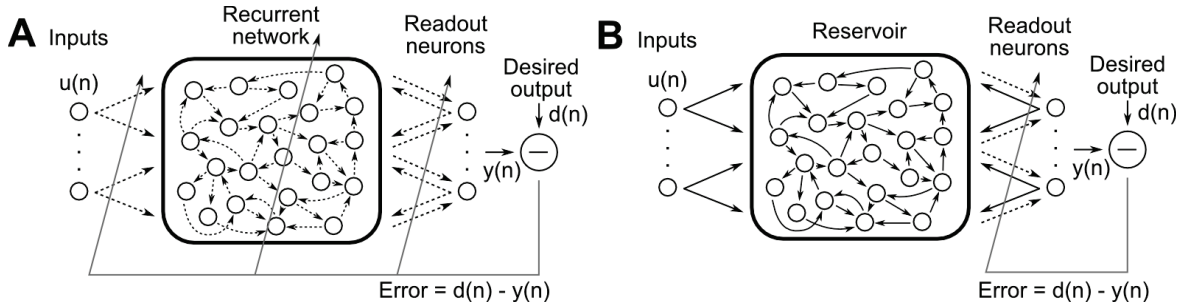

Figure 3.4: Comparison between the architecture of a reservoir network and previous recurrent networks. *The basic architecture of recurrent networks comprises an input layer, a recurrent network and an output layer (also denoted as readout). The error, defined as the difference between the desired output and the actual output, is used to modify the weights of the network (grey arrows). Dashed arrows are modifiable connections, plain arrows are fixed connections.* ***A.*** *Previous approaches involved training the recurrent network connections and the input connections.* ***B.*** *Reservoir computing approach trains only the readout connections (between the recurrent layer and the readout) thereby avoiding the complexity of training recurrent connections. Instead, these connections are randomly generated with parameters eliciting a suitable dynamics for the task at hand.*

with one hidden layer. In both types of models, the activity of the hidden layer is read out by an output layer, and successive transformations of the inputs result in non-linear processing. However, the analogy stops here, as the number of neurons in the reservoir is usually much higher than the number of input neurons, whereas the ratio between input and hidden neurons is generally inverse in canonical MLP. In addition to the recurrent connections within the reservoir layer, reservoir networks can include feedback connections from the readout/output layer to the hidden layer.

As in MLPs, the expansion of inputs into a higher dimensional space facilitates regression/approximation based on these inputs, and/or separation (e.g. for classification) of these inputs. Because of this property and the fact that learning takes place only between this expanded space and the output, reservoir are sometimes compared to temporal versions of *Support Vector Machines*[6] (SVM), which are the state of the art for a wide range of spatial problems in machine learning (Hermans and Schrauwen, 2012; Schrauwen et al., 2007; Jaeger et al., 2007b). Even though the representations of input sequences within the recurrent network are highly dynamic, a simple linear readout with fixed weights is able to continuously separate the trajectories of each sequence (Buonomano and Maass, 2009). This property is very important for the biological plausibility of such networks, because if such representational power exists in cortical networks, a single cortical neuron corresponding to the readout should be able to extract spatio-temporal

[6] Except that the expansion is computed explicitly in reservoirs, not in SVM.

representations.

The global architecture described so far is common with some RNN that are trained using classical gradient descent methods. As explained above, the main progress resides in the generation of recurrent connections that are fixed, not subjected to training. Because the reservoir method avoids the laborious task of training the recurrent connections it lowers the complexity to be proportional to the number of neurons in the reservoir *O(N)*.

Instead, the weights of the recurrent layer are generated prior learning with a stochastic process ensuring a favorable dynamical regime for the task at hand in the reservoir layer. Lower complexity also means that, computational resources being equal, a much higher number of neurons in the recurrent network can be used to perform the same task, so a much richer expansion of the inputs into the recurrent layer is available. The key to understanding the choice of fixed recurrent connections lies in the already very rich dynamics that can be obtained from untrained recurrent networks. The readout layer simply needs to combine linearly the already rich representations of the hidden layer to produce the target output (Fig. 3.5).

Echo State Networks are mostly used in the machine learning domain for signal processing. Specific implementations include speech recognition, robot motor control, financial forecasting and medical applications like seizure detection in epileptic patients (Lukoševičius et al., 2012). However, it has also been used in computational modelling to model fading memory of reward (Bernacchia et al., 2011) and in our team to model language processing (Hinaut and Dominey, 2013). Liquid State Machine has been mainly used in computational neuroscience to explore the computational property of generic microcircuits (Maass, 2011; Nikolić et al., 2009; Maass et al., 2002).

3.3.3 Reservoir Dynamics

The key to reservoir networks is the appropriate generation of the recurrent connections ensuring a rich dynamical representations in the recurrent layer. Indeed, one of the main issue in training a reservoir lies in the right set of parameters used for the generation of internal connections, which should underlie dynamics suited to the task at hand (Jaeger, 2001).

The seeding papers in ESN and LSM proposed similar capabilities for reservoir. As stated in Maass et al. (2002), if a reservoir possesses two properties, the *point-wise separation property* and *universal approximation property*, it "has universal power for computations with fading memory on functions of time", i.e. it is a Universal Turing Machine that include temporal processing of recent inputs. *Point-wise separation property* is related to the separability in the reservoir representations of two non-identical

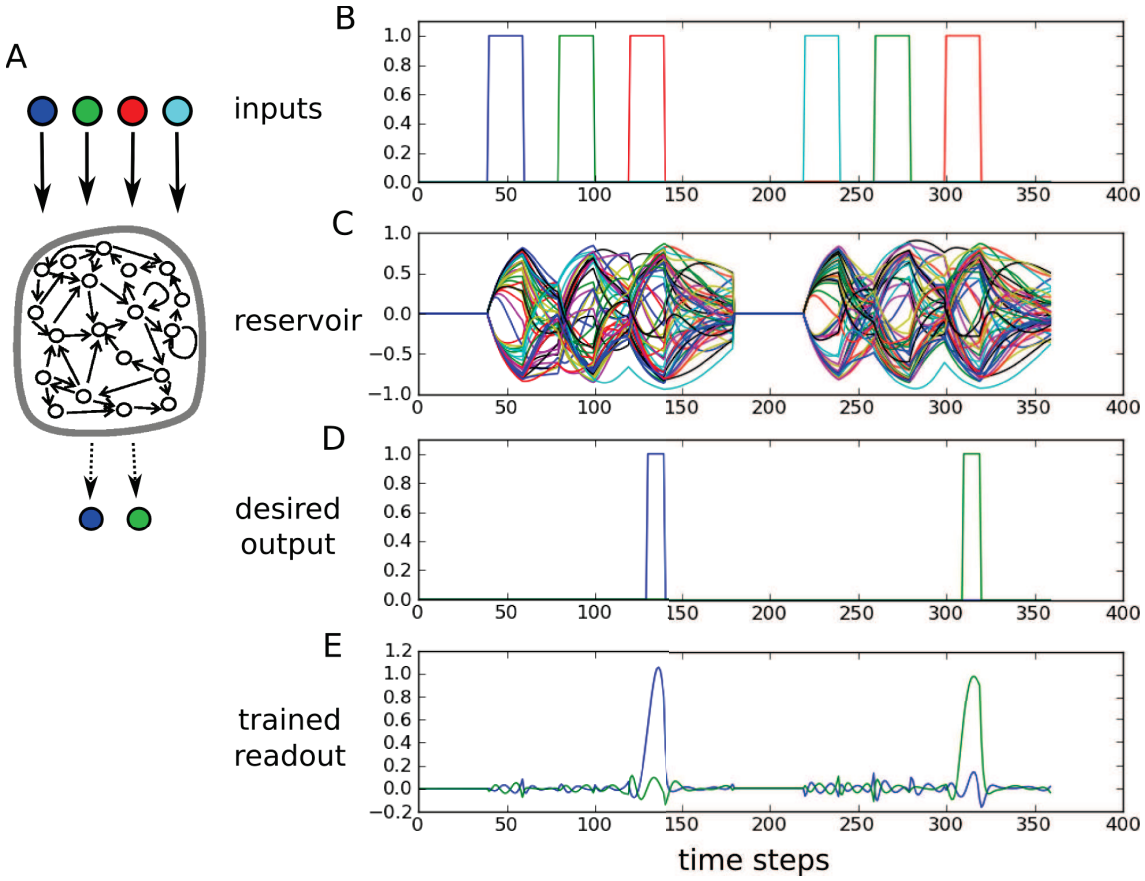

Figure 3.5: Training a reservoir with linear regression. *This figure presents an example of Echo State Network with leaky integrator neurons (**A**) that learns to discriminate two sequences of three consecutive inputs (**B**) with a linear regression between the activity of the reservoir (**C**) and the desired output of the network (**D**). Only the first element of each sequence is different and the network must learn to segregate the two sequences when the last input is fed to the network. Note that the activity within the reservoir is reset at the end of the first sequence. The reservoir layer displays rich dynamics that encode the successive inputs and their order. After the readout weights are trained with linear regression, the network discriminates the two sequences (**E**).*

inputs. The *universal approximation property* echoes what has been said on MLP but in the spatio-temporal domain, i.e. the capacity of the reservoir to approximate any finite-time function. The latter property has been demonstrated in RNN (Funahashi and Nakamura, 1993). In other words, a reservoir endowed with both these properties could theoretically process any spatio-temporal input if the temporal aspect is limited in time.

Indeed, the memory of previous inputs vanishes with time, which means that the influence of a specific segment of an input stream on subsequent input processing becomes negligible (Maass et al., 2004, 2002). This property is necessary for the reservoir to gen-

eralize over similar inputs and is part of the denominated *echo state property* (Jaeger, 2001). It is a formal definition of the desired dynamics within an ESN which echoes the fading memory principle from LSM literature. It states that the state of the reservoir is the result of the recent inputs fed to it and not by its own initial state, so in other words, it gradually forgets its own states.

To obtain this desired dynamics in the reservoir, some parameters have to be carefully chosen. Importantly, the recurrent weights must be scaled. For ESNs, a popular method consists in scaling the weight matrix with a parameter denoted as the *spectral radius*, which is the largest eigen-value of the weight matrix. As most parameters defining the connectivity of the network, its optimal value depends mostly on the inputs fed to the network. The distribution and sparseness of the recurrent layer is typically chosen to be low (~0.1), however it has a limited influence on the dynamics (Schrauwen et al., 2007).

Spectral radius has an influence on the dynamical excitability of the reservoir layer. It is at the heart of a trade off between fading-memory span and generalization. A low spectral radius elicits short fading memory of the inputs, accompanied with good generalization abilities because similar inputs elicit similar reservoir activities. Conversely, a higher spectral radius yields longer fading memory with good separation abilities, small differences in inputs are amplified because the dynamics becomes more chaotic. Indeed, these properties are linked to the degree of chaoticity of the reservoir dynamics. Key studies have shown the computational power of dynamical regimes close to chaos in rather simple dynamical systems (Langton, 1990), subsequently referring to the exploitation of this regime as *computation at the edge of chaos*. Interestingly, for best performances the spectral radius is often set to a value where the reservoir regime is almost chaotic. Likewise, in the LSM literature it has been shown that reservoirs perform optimally in this same dynamical regime (Legenstein and Maass, 2007).

A further fine-tuning of the recurrent connections can be achieved through *intrinsic plasticity*. It is a biologically inspired (homeostatic plasticity), local[7], information maximization[8], unsupervised tuning of each neuron activation level that has demonstrated its capacity to steer the dynamics of the entire network towards a computationally desirable regime (Schrauwen et al., 2008; Steil, 2007).

3.3.4 Learning Procedure

Training a reservoir consists in modifying the readout connections to produce the desired output. The easiest method is a linear regression in batch mode (single step learning, see fig. 3.5 for an example). While the sequence of inputs is fed to the network, the

[7] Modifications taking place at the level of the neuron, no global training signal is used.
[8] Along the formal information theory framework.

activity of reservoir neurons is stored. A least mean square regression is performed between the activity of the reservoir neurons and the desired output, producing the matrix of readout weights. This simple linear regression can be performed with regularization to prevent overfitting. In this case ridge regression[9] is the most commonly used method. Online/iterative learning can also be implemented with recursive least squares which is the iterative counterpart of least mean square (Jaeger, 2002; Jaeger and Haas, 2004). However this method is more complex and requires more computational resources.

A common issue arises when the readout is connected back to the reservoir, because the trained weights are part of the recurrence in the network. To train the network in batch mode, the desired output is clamped to the reservoir as an input during the training. However, the actual output after training is rarely identical to the desired output, so after training the reservoir layer will be fed back an output that differs from the one it receives during learning. In this case, the readout may diverge even more from the desired out, successively amplifying errors in the network. Several basic solutions exist to overcome this circular dependency, however, there is no guarantee of convergence. Ridge regression is the easiest method. Adding noise to the reservoir units is another way to sample slight deviation from the expected activity, thereby making the network more robust to output errors. The training succeeds if the set of weights creates an attracting dynamics around the desired output, reducing (instead of amplifying) the errors and keeping them within an acceptable distance from the target output.

FORCE learning, a recent method for training RNN, is a clever method for this exact purpose (Sussillo and Abbott, 2009). Most online learning methods iteratively modify the weights with small steps in the hope that the actual output eventually converges to the target output. FORCE learning operates with large modification of the weights to obtain an output very close to the target output, and this, from the onset of learning. However, it does not completely correct the weights, so that small deviations from the output are fed back to the recurrent layer. Because, the deviation from the desired output is kept small from the onset of learning, it provides the recurrent layer with a feedback resembling the target output. In addition, the network samples small deviations from the desired output, which makes network more robust to deviations after training. In other words, the amplitude of weight modifications gradually decreases while error is kept small, as opposed to classical iterative methods, in which weight modification is usually small, and output error gradually decreases. It elicits a higher tolerance to errors, and strengthens the attractor created with learning. Note that an extended version of this method has been successfully applied to the training of the connections within the recurrent layer. Recently, Hoerzer et al. (2014) developed a very similar

[9] Also denoted Thikonov regression.

method that uses more biologically plausible learning methods, i.e. Hebbian learning with reinforcement signal.

3.4 Recurrence for Variation in the Spatio-Temporal Domain

As mentioned in the first chapter, adaptive behavior includes the capacity to process and produce sequences. Indeed, activities within the PFC seem to reflect corresponding spatio-temporal representations. Yet, how these spatio-temporal representations arise is still largely unknown. Previous chapter reported the work that has been done on neural networks to explain the origin of distributed representations of arbitrary contingencies. A simple method is used, namely, a layer of randomly connected neurons.

However, the representations produced are limited to the spatial domain, i.e. the information available at one point in time. In this chapter we develop the idea that a similar simple mechanism can be used to generate arbitrary spatio-temporal representations. The reservoir paradigm which we reviewed is based on randomly connected recurrent networks. This paradigm allows for universal computational power with fading memory. In other words, reservoir can represent any combination of present and recent inputs along with the order in which they were fed to the network.

The representational power of these networks lies in the recurrent nature of their hidden layer. Yet, a striking feature of the connectivity of cortex is its local recurrent structure. In addition, this property is found to be most pronounced in the PFC. Furthermore, the readout layer can extract spatio-temporal representation of the recurrent layer with a linear readout. Transposed to the cortex, it implies that a single neuron could harness the representational power of PFC, even if the neural-population activity is highly dynamic.

With these elements in mind, we posit that the representational capacity of the cortex may be explained by its recurrent nature. If one ascribes the computational properties of reservoirs to recurrent networks of neurons in the PFC, arbitrary combinations of recent sensory, motor and internal information should be represented in cortical activity. Therefore, if a particular contingency can be explained by this information, we posit that a representation of this contingency exists in the distributed activity of the PFC. However, these representations are limited in time, and may be relatively weak in the face of noise. Next chapter will address mechanisms developed to overcome these issues.

Chapter 4

Explicit Context Representations

4.1 Introduction

The last chapter ended on the introduction of universal spatio-temporal representations by means of random recurrent connections. What must be kept in mind is that the temporal dimension is inherently and exponentially fading. This means that memory traces of inputs, and, consequently, their temporal relations, have a limited lifetime. Yet recordings in the prefrontal cortex (PFC) display explicit representations of task relevant information that seem to bridge past sensory inputs with current behavioral demand. Depending on the literature and the task at hand, this process may be described in cognitive terms as working memory (WM), rule maintenance, or context-dependent processing[1]. For simplicity's sake, in the current chapter we will refer to context as encompassing WM and rule maintenance. The reservoir paradigm in its original version may be limited to explain such representations.

A system that can process information must somehow create a bridge between past inputs and subsequent processing. That bridge would necessarily span delays larger than the fading memory capacity already present in the dynamics of the network. Because only selected information from inputs must influence future processing, the key concept of this cognitive function is that it is an intentional maintenance of selected features of input. The intentional component of this function implies that learning may play an important role in order to distinguish the relevant from irrelevant information. As we will see, it can lead to the formation of context representing relevant information for current processing. As opposed to the spatio-temporal representations in the reservoir that are implicit and universal, the solutions presented in this chapter employ explicit

[1] It must be noted that these terms are not precisely defined, literature uses them with different intended meaning. For example, one can argue that context-dependent processing refers to what a reservoir provides through its time-limited spatio-temporal representations.

stable representations of context, most of them with attractors.

We will first review a detailed model of working memory, then proceed with a model of context formation through concretion of attractors. In the second section, we will expand on two interesting types of model that involve context-dependent processing.

4.2 Attractors for Working Memory and Context Representation

4.2.1 Working Memory in Attractors

Because the first extensively studied correlate of WM is persistent activity, early neural network models have focused on dynamics that elicit this type of activity. The most evident solutions are attractors of the neural dynamics that reproduce different instances of persistent activity. In the simplest case, a fixed point[2] produce a static and fixed persistent activity, while continuous variables can be encoded in bump attractors[3].

The first use of an attractor network to model memory is the Hopfield network tat we presented in the second chapter. Despite its biological relevance in reproducing associative and content-addressable memory, its original implementation does not reproduce the maintenance aspect of WM, since the persistent activity is only possible through the constant activation of an input to the network. However, it set bases for new models of memory with active maintenance through attractors in RNN (Amit, 1989, 1995). One of the most popular biological model of WM in RNN has been developed by Compte et al. (2000). They postulate that recurrent excitation between pyramidal neurons mediated by NMDA receptors, a type of glutamatergic receptor in synapses, and mutual inhibition mediated by GABAergic interneurons, may be underlying persistent activity. This model reproduces the maintenance of a saccade direction in an occulomotor delayed-response task. To achieve maintenance of saccade direction during delay, a calibrated balance between excitation and inhibition elicits reverberating dynamics that stabilize in a cyclic bump attractor. Another version employing similar biological mechanisms implements fixed point attractors to model the maintenance of object in WM. Figure 4.1 illustrates the interaction between excitatory pyramidal and inhibitory interneurons used in this model.

This model is a popular instance of a rich domain (Wang, 2001; Durstewitz et al., 2000). Note that models have been developed along the same biological principles to explain decision making (Wang, 2008).

[2] Mathematically speaking, a fixed point is a 0-dimensional attractor.
[3] A 1-dimensional attractor, called ring attractor if it is cyclic.

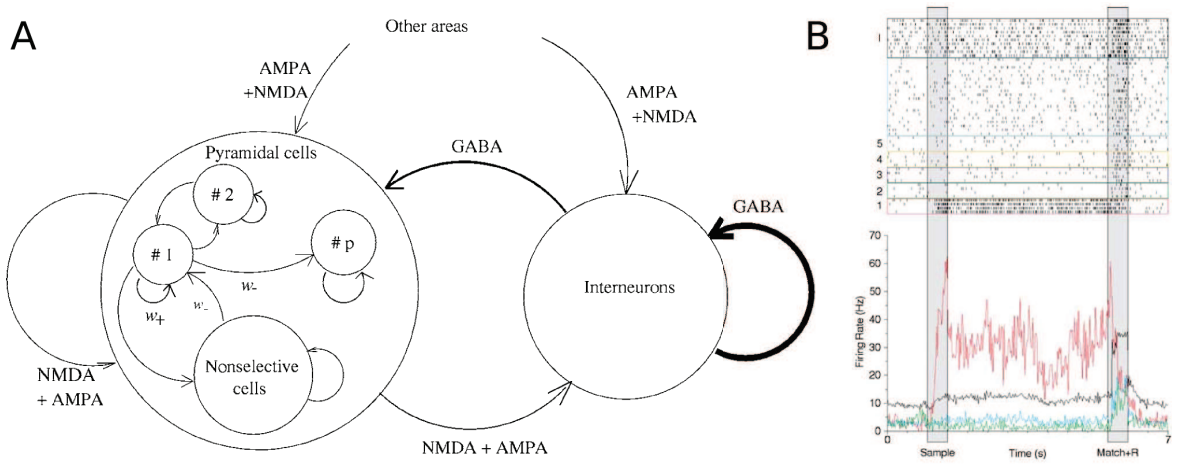

Figure 4.1: Biologically realistic model with an attractor network (Figure from Brunel and Wang (2001)). ***A***. *Schematic description of the Brunel and Wang's WM memory model. Distinct pools of pyramidal neurons (left hand circle) represent different objects to be held in memory during a delay. They project to GABAergic (inhibitory) interneurons which projects back to pyramidal neurons and to other interneurons, maintaining a balance excitation in the network. Projections from other areas onto pyramidal and GABAergic neurons selectively activates pools of pyramidal neurons and trigger the delay persistent activity.* ***B***. *Raster and average activity of persistently firing spiking neurons during the delay.*

4.2.2 Context Formation in Neural Networks

Attractors in neural networks have demonstrated their explanatory power concerning WM. A related subject is the formation of context representations that temporarily indicate the current rule in effect. Since the work on attractor led to promising explanations of temporary maintenance of information, studies have attempted to demonstrate how attractors can form and lead to the representation of sequences and context.

Neurophysiological recordings in the temporal cortex revealed neurons that are selective to stimuli that are close in the temporal order of sequences in which they belong (Miyashita and Chang, 1988). This observation led to the development of models using Hebbian synaptic plasticity to develop attractors based on the temporal contiguity of stimuli repeatedly presented in the same sequential order (Griniasty et al., 1993; Brunel, 1996). Note that, in terms of dynamical regime, the representation of sequence in these models and in the reservoir are opposite, because the reservoir represents sequences with transient dynamics.

Recently, Rigotti et al. (2010a) used this same method to implement the formation of context by incremental concretion of attractors. They simulated an appetitive and aversive trace conditioning experiment in which the contingencies between two uncondi-

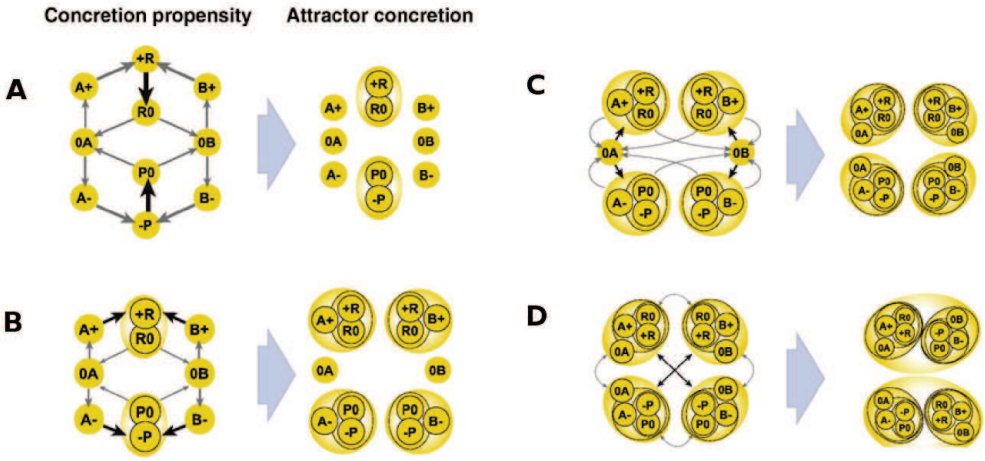

Figure 4.2: Model of context formation through attractor concretion (Modified figure from Rigotti et al. (2010a)). *A and B are the two CS, and R and P denote the reward and the punishment, together they constitute the inputs to the network. An associative layer represent the prediction of the network, +, - and 0 respectively denoting a predicted reward, predicted punishment and the neutral state, i.e. when the associative layer is inactive. The pair of characters represent the attractors that have already formed after the first step of learning in the context layer. For example A+ is the attractor active after stimulus A is followed by the prediction of a reward in the associative layer. In the four parts of this figure, the left hand panel illustrates in a graph the temporal succession of attractors due to the task design. Each node of the graph represents the already formed attractors while the arrows represent the possible transitions between attractors. The more frequent the transitions, the wider the arrows. The right hand panel shows the merging of attractors due to temporal contiguity of the most frequent transition.*

tioned stimuli (US) and their respective conditioned stimuli (CS) were reversed. Context formation is implemented in a two-step process. First, neurons transiently co-activated by events of the task see their mutual connections strengthened with Hebbian learning. It progressively produces attractors that segment the task in its successive steps. These steps are delimited in time by the events of the task. Secondly, the temporal continuity between task states triggers the merging of successive attractors. Hebbian learning is also used in this case, causing the attractors most frequently succeeding each other to iteratively merge until two major attractors remain which represent the two contingencies between US and CS (fig. 4.2).

4.3 Processing with Explicit Context Representations

4.3.1 Introduction

In the previous chapter, we addressed the representation of context in transient dynamics and the contextual processing of inputs through the interaction between current inputs and the reverberatory activity elicited by previous inputs. Here we introduce a different mechanism of contextual processing, where information about previous inputs is held in memory in an explicit form. In this case, continuous activity of units steadily represent the information to be held. While the transient dynamics contextual processing allows for temporal processing on a short time scale, steady and explicit contextual representations allow the system's processing to be influenced by inputs that were fed to the network on a much longer time scale. From a biological point of view, the transient dynamics exposed in the previous chapter might be inherent to the connectivity of cortical neuron sub-populations, in contrast to explicit contextual representations that may need to be learned.

4.3.2 Input Units Feeding Context to the Network

Feeding the context to the network as an input is the simplest method to influence the dynamics of a network so that it to processes inputs depending on a defined context. The input layer of the network will then be divided in regular input units that feed the features of each input pattern, and context units whose activation will vary to represent the context. For example, Cohen and Servan-Schreiber (1992) simulate deficits in schizophrenic patients with the Stroop task performed with a MLP that represents the contexts (or rules) "word reading" and "color naming" in the activation of two input/context units. In this case, backpropagation is used to train the network to perform the Stroop task.

4.3.3 Contextual Processing with Attractors

Trying to unveil the computational mechanism behind the contextual processing of sensory information, Mante et al. (2013) developed a model of the PFC with a RNN trained with backpropagation that was then compared with macaque monkey electrophysiology. The task modeled is a recent version of a long series of tasks on perceptual decision making (Newsome and Pare, 1988; Kim and Shadlen, 1999), which features moving dots. In this version, random dots have two properties, direction and color.

In the color context, the agent must determine which color is most represented in the dots. Conversely, in the motion context, it must pay attention to the most represented direction of the dots. These independent visual features are present whichever context is in effect. This task is a perceptual decision making version of the WCST introduced in the first chapter. One of the differences is that the difficulty of the present task can vary, the salience of the relevant perceptual feature can be modulated with a parameter denoted as coherence. Previous neural recordings and models with a similar task have suggested that the cortex accumulates over time evidence of the perceptual feature to process and that, when coherence is higher, decision is reached more rapidly.

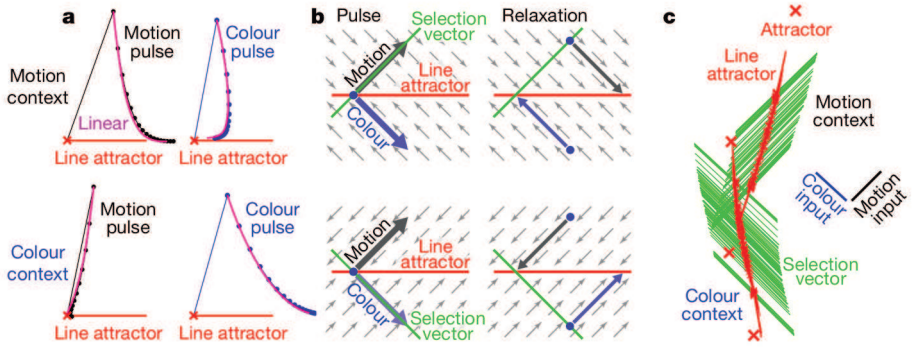

Figure 4.3: Context-dependent processing through linear attractors (Figure from Mante et al. (2013)). *A RNN is taught with backpropagation to perform a perceptual decision making task with two distinct rules. The training creates two line attractors whose axis represent the amount of evidence for a choice.* ***a****. To understand the dynamic of the context attractors, they observed the response of the network after a brief activation (pulse) of one sensory input neuron for 1 ms. In motion context, a motion pulse displaces the state of the population along the line attractor for choice while the colour pulse has no effect. In the colour context, the effects are opposite.* ***b****. The four graphs illustrate in two dimensions how the attractors integrate motion and color inputs with a vector field and a selection vector. The vector field shows in which direction the population state is moved if sensory input displaces the population state from the line attractor. The direction of the selection vector determines the type of sensory information integrated along the line attractor that represents the amount of evidence. Pulses of relevant sensory information are integrated (left) while irrelevant information is ignored (right).* ***c*** *Representation of both line attractors and selection vectors in two dimensions. Note that the selection vectors are parallel to the relevant sensory axis and perpendicular to the irrelevant sensory axis as shown in figure* ***b****.*

The present study innovated with an elaborated Principal Composant Analysis that revealed that each component of the sensory information actually reaches the PFC, whichever rule is in effect. Therefore, irrelevant sensory information for the current decision is still represented in the PFC. To explain how only relevant information con-

tribute to the choice, they implemented the task in a RNN (having the same architecture as a reservoir) fed with the two sensory information (motion and color), and with the context, which, in this case, is explicitly represented in the activity of two input neurons. The activity of one output neuron connected to the recurrent layer provides the answer of the network, which is trained with a recent implementation of the backpropagation algorithm (Martens and Sutskever, 2011). The training produces two linear attractors, each corresponding to a context (color or motion rule) (Fig. 4.3). When the color rule is in effect, only the color information is integrated along the choice axis, and, conversely, with motion rule, only motion is integrated.

4.3.4 Contextual Processing with Mixed Dynamics

Because reservoirs have a fading memory of past inputs, its spatio-temporal processing capabilities are limited in time. To circumvent this inherent limitation Maass et al. (2007) introduced learning inside a Liquid State Machine (LSM) for a few neurons only. It is equivalent as readout neurons that connect back into the recurrent layer. They demonstrated theoretically that this feedback mechanism produce a system with universal capabilities that can map time-varying inputs to a time-varying output potentially unlimited in time. This feedback mechanism produces high dimensional attractors, that maintain selected information and still allow for complex spatio-temporal processing. For example they demonstrate that these circuits can produce WM and integration of evidence dynamics with continuous attractors.

As of Echo State Networks, Pascanu and Jaeger (2011) have developed a model of working memory (WM) with a similar mechanism. Outlying neurons that represent states to be memorized feed their activity back to the recurrent layer. Although the approach is defined as bio-inspired, the network was developed for machine learning purposes. Nevertheless, the results are of interest for neural modeling purposes. The model implements a task of temporal image processing. The network is trained to count the number of opening brackets in a stream of characters, each closing bracket decreasing the count number. This count is held in a set of WM units schematically represented outside of the network (Fig. 4.4).

In addition, the output units are trained to predict the next character in the sequence. The probability of succession between character depends on the level of opening brackets. Consequently, to efficiently predict the next character, the output has to rely on the WM information fed back to the reservoir. WM units are fully connected and display fast attracting dynamics, switching between binary states that keep track of the opening level of detected brackets. In addition from the usual training of output connections, WM states are trained with the modification of connections within the WM layer, and

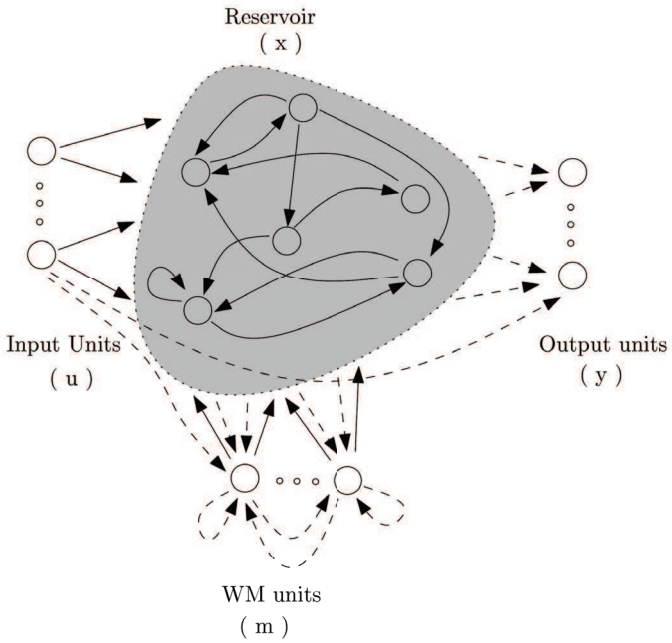

Figure 4.4: Architecture of Echo State Network (ESN) with working memory (WM) units (Figure from Pascanu and Jaeger (2011)). *WM units represent the opening level of detected brackets from a stream of characters presented in input. The output must predict the next character as a function of the input and the opening level of brackets. WM memory units differ from classical output units by having trainable connections between themselves (dashed arrows).*

between the reservoir and WM layer. After training the network, they observe what they refer to as input driven attractors because the continuous inputs fed to the network influence the attracting dynamics.

The nature of the dynamics implemented in this network is mixed in the sense that the network displays simultaneously attracting and transient dynamics. Attractors keep track of the opening level of brackets which is used by transient dynamics to process inputs. This attracting mechanism with feedback may be essential to explain contextual processing in the brain.

4.3.5 A Mechanism for Universal Representation in the Cortex

At the end of the previous chapter, we developed the argument that the cortex is endowed with the same representational power as reservoirs because of its recurrent structure. However, since reservoirs have a fading memory of recent inputs, another mechanism must explain the explicit representations that the cortex displays in its activity. Consequently, in the previous section, we presented a solution that involves readout neurons robustly representing contextual information with attractors and feed-

ing back their activity to the recurrent layer. This method allows for robust and lasting representations of information related to past inputs. We postulate that the cortex represents context with a similar feedback mechanism which allows for transient dynamics to encode complex contingencies influenced by this context.

Chapter 5

Hypothesis and Objectives

We posit at the end of the first chapter that representations of complex contingencies pre-exist the learning that eventually link them to a behavior. The necessary representations may not be directly available, but would be the result of an iterative process comparable to selectionism in psychology, whereby a cyclic process of variation, selection and retention produces representations of successively more complex contingencies. In the following sections we will present our hypothesis on how the cortex underlies the variation process, followed by the objectives of this thesis.

5.1 Hypothesis

For clarity, we use the term *contingency* to refer to a situation that can be defined by a set of past and present sensory, motor, and internal information denoted as *inputs*. The *context* of a contingency is the subset, composed by the non-recent past inputs, that participate in the definition of this contingency.[1]

Following the argument of the first chapter, complex behaviors that underlie adaptation can be learned because the cortex can represent complex contingencies. The main hypothesis of this thesis is that the recurrent nature of local cortical circuits endows the cortex with the inherent capacity to represent arbitrary contingencies that depend on current and recent inputs in its distributed activity. Consequently, we argue that:

- a single neuron can robustly extract any contingency from this complex distributed activity with a simple linear mechanism
- contingencies relying on context require explicit contextual representation that are fed back to recurrent cortical networks

[1] Of course, a broader definition of context would also include recent task variables. However, we exclude them from this strict definition to avoid confusion in the statement of our hypothesis.

- the combination of these mechanisms provide the cortex with a potentially universal representation of contingencies

5.2 Objectives

The hypothesis presented above is highly influenced by the work of Dominey and colleagues (Dominey et al., 1995; Dominey, 1995; Dominey and Ramus, 2000), the work of Maass and colleagues (Maass et al., 2002; Buonomano and Maass, 2009), and the work of Fusi and colleagues (Rigotti et al., 2013, 2010b). Together, these teams have demonstrated the spatiotemporal processing power of recurrent networks with random fixed connections, and stressed the importance of complex non-linear activities in the cortex. However, no systematic comparison between a reservoir and the activity of prefrontal neurons has been performed to show the spatio-temporal representations encoded in the PFC.

Our objectives are:

- To demonstrate, in a cognitive task, that simple neural networks with random recurrent connections can produce rich spatio-temporal combinations of inputs, and that they are sufficient to learn the task
- To confirm the presence of dynamic non-linear combinations of task variables in the activity of PFC neurons
- To assess if a simple linear decoder can extract the spatio-temporal representations of task variables from activity of the PFC, in a similar manner that a linear readout can extract dynamic representations of a reservoir
- To show, in a cognitive task, that feedback of persistent activity representing contextual information allows for robust maintenance of context through time thanks to attracting dynamics, while still allowing for spatio-temporal processing
- To confirm that the population activity of PFC neurons displays a corresponding mix of attracting and transient dynamics in the same cognitive task

Part II

Experiments

Chapter 6

Extracting Task Variables From Prefrontal Activity

Although the present experiment is dedicated to the assessment of decoders in order to extract task variables, the results obtained can inform our hypothesis on the nature of information representation in the cortex.

Several decoders are tested to extract exogenous and endogenous information from the frontal eye fields (FEF) of macaque monkeys. The former being related to the spatial position of a cue, and the later refers to the instructed position of a target.

The main contribution of the author of this thesis was to test reservoirs as decoders, from a signal processing perspective. As explained in the state of the art section, echo state networks have been used for signal processing of time series, and have demonstrated excellent performance compared to previous methods. Since activity in the PFC has demonstrated to be dynamic, especially after stimulus presentation, the reservoir was tested to assess its capacity in extracting task variables from a spatio-temporal processing perspective. However, as we will see, the temporal component of information processing capabilities of such a network was not useful because task variables could be extracted with a much simpler method.

OPEN ACCESS Freely available online

PLOS ONE

Comparison of Classifiers for Decoding Sensory and Cognitive Information from Prefrontal Neuronal Populations

Elaine Astrand[1], Pierre Enel[2], Guilhem Ibos[1], Peter Ford Dominey[2], Pierre Baraduc[1], Suliann Ben Hamed[1]*

1 Centre de Neuroscience Cognitive, UMR 5529 CNRS-Université Claude Bernard Lyon I, Bron, France, **2** Stem Cell and Brain Research Institute, INSERM U846-Université Claude Bernard Lyon I, Bron, France

Abstract

Decoding neuronal information is important in neuroscience, both as a basic means to understand how neuronal activity is related to cerebral function and as a processing stage in driving neuroprosthetic effectors. Here, we compare the readout performance of six commonly used classifiers at decoding two different variables encoded by the spiking activity of the non-human primate frontal eye fields (FEF): the spatial position of a visual cue, and the instructed orientation of the animal's attention. While the first variable is exogenously driven by the environment, the second variable corresponds to the interpretation of the instruction conveyed by the cue; it is endogenously driven and corresponds to the output of internal cognitive operations performed on the visual attributes of the cue. These two variables were decoded using either a regularized optimal linear estimator in its explicit formulation, an optimal linear artificial neural network estimator, a non-linear artificial neural network estimator, a non-linear naïve Bayesian estimator, a non-linear Reservoir recurrent network classifier or a non-linear Support Vector Machine classifier. Our results suggest that endogenous information such as the orientation of attention can be decoded from the FEF with the same accuracy as exogenous visual information. All classifiers did not behave equally in the face of population size and heterogeneity, the available training and testing trials, the subject's behavior and the temporal structure of the variable of interest. In most situations, the regularized optimal linear estimator and the non-linear Support Vector Machine classifiers outperformed the other tested decoders.

Citation: Astrand E, Enel P, Ibos G, Dominey PF, Baraduc P, et al. (2014) Comparison of Classifiers for Decoding Sensory and Cognitive Information from Prefrontal Neuronal Populations. PLoS ONE 9(1): e86314. doi:10.1371/journal.pone.0086314

Editor: Thomas Boraud, Centre national de la recherche scientifique, France

Received October 12, 2013; **Accepted** December 13, 2013; **Published** January 23, 2014

Copyright: © 2014 Astrand et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: EA was funded by the Centre national de la recherchescientifique and the Direction générale des armées. S. BH and the present study were funded by the ANR-05-JCJC-0230-01, a CNRS multidisciplinary program (2010) and a CNRS Biology-Maths-Computer science program (2012). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: benhamed@isc.cnrs.fr

Introduction

Decoding neuronal information is an important analysis tool in neuroscience both as a means to understand how neural information is distributed and multiplexed over large populations [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], and as a means to drive neuroprosthetic effectors [12], [13], [14], [15]. In this framework, classifiers are used to define the most probable state of a given variable (the position of a stimulus in space, the direction of the intended motor plan etc.), given the observed instantaneous simultaneous activity of a neuronal population. Above-chance decoding accuracy indicates that the neuronal population contains reliable information about the variable of interest, whether its individual neurons also do or not.

In order to optimize their prediction, all classifiers define a decision boundary in the space of the variable of interest (2-D space for stimulus position or movement goal, n-class discrete space for stimulus or movement classification), using a training set of data, i.e. a set of neuronal population activities matched with the actual experimental condition that they correspond to. The accuracy of the decoders is then evaluated on a testing set of data, corresponding to neuronal population activities from an independent sample. Accuracy is calculated as the percentage of correct predictions provided by the classifier. The shape and properties of the decision boundary varies across classifiers. Linear classifiers will set hyperplane boundaries while non-linear classifiers will set complex non-planar boundaries. Flexible decision boundaries will maximize the separation of the training neuronal population response as a function of the decoded variable, including irrelevant idiosyncratic noise patterns specific of this training data. This over-fitting of the decision boundary will result in a poor generalization on new testing data (see [16], [17]). In contrast, a too simple decision boundary, such as a hyperplane, may often fail to account for a non-linear encoding of the variable of interest by the recorded neuronal population.

Most classifiers have been developed in the fields of statistics and machine learning. As a result, their mathematical properties are well understood. Early studies have formalized the use of major classifiers to the readout of continuous variables (such as position in space, orientation etc.) from neuronal population activities [18], [19]. However, in the face of real data, the sensitivity with which information is extracted from neuronal activity will depend on several factors. In particular, a given neuronal population may not

encode with the same reliability and discrimination power all the variables it represents (e.g. a sensory information as compared to a cognitive information). As a result, classification sensitivity will depend both on the general response properties of the neuronal population being targeted and on the variable being decoded. The decoding sensitivity will also depend on the classifier being used as well as on the adequacy of the classifier with the experimental constraints. Several studies have used two or more decoders at reading out neuronal population activities (e.g. [20], [1], [2], [5]), without however pursuing a systematic comparison of their performance and how it is affected by the properties of the experimental data. In the following, we compare the readout performance of six commonly used classifiers operating on monkey frontal eye fields (FEF) spike signals, as a function of the size of the neuronal population, the number of training trials, and the balance in the data. The classifiers fall into three general decoder classes: probabilistic decoders, linear decoders and non-linear decoders. The classifiers we focus on are classifiers that have been used or proposed to decode neuronal population activities (non-exhaustive selection). These are a regularized optimal linear estimator, in its explicit formulation (*regularized OLE*, [12]) or in its linear artificial neural network approximation (ANN *OLE*, [1], [2]), a non-linear artificial neural network estimator (ANN *NLE*, [1], [2]), a non-linear naïve Bayesian estimator (*Bayesian*, [21]; please note that the naïve Bayesian estimation is formally equivalent to a Maximum likelihood classification) and a non-linear support vector machine classifier (*SVM*, [4]). A non-linear *Reservoir* recurrent network classifier (*Reservoir*, [22]) has also been tested because of its potential interest in decoding variables that have a specific organization in time. The general architecture and properties of these classifiers are described in the methods section.

We will compare how these decoders read out two distinct types of information available in FEF neuronal population responses. The first decoded variable corresponds to the position at which an initial stream of visual stimuli is presented. This information is exogenously driven by the environment (the presentation of the visual streams) and is robustly represented in the FEF [23], [24]. The second variable corresponds to the interpretation of the instruction held by the cue and the corresponding attention orientation signal. This information is endogenously driven in that it corresponds to the output of internal cognitive computations performed on lower level exogenous characteristics of the cue (here, position and color). Such endogenous attentional information is known to build up in the FEF [4], [25] and to influence lower visual areas, thanks both to feedback [26] and feedforward connections [27], [28].

In summary the present work pursues two objectives: 1) investigate whether endogenously driven neuronal information can be decoded with the same performance as exogenously driven neuronal information, and 2) identify the classifier that performs best at decoding neuronal information as a function of the experimental factors (neuronal population properties, subject's behavior and number of trials).

Methods

Ethical statement

All procedures were in compliance with the guidelines of European Community on animal care (European Community Council, Directive No. 86–609, November 24, 1986). All the protocols used in this experiment were approved by the animal care committee (Department of Veterinary Services, Health & Protection of Animals, permit number 69 029 0401) and the Biology Department of the University Claude Bernard Lyon 1. The animals' welfare and the steps taken to ameliorate suffering were in accordance with the recommendations of the Weatherall report, "The use of non-human primates in research". The study involved two *Rhesus maccaca* (a male, 10 kg, age 7 and a female, 7 kg, age 6), a standard in electrophysiological studies. The animals were housed in twin cages (2 m^2 by 2 m height in total). The twin cages could be separated in two individual cages or on the opposite, connected to form a unique housing for a pair of monkeys thus offering the monkeys a socially enriched environment. This last configuration was the norm. Twin cages communicated with a larger play cage ($4\times1.5\times2$ m^3) to which the monkeys were granted access on days on which they were not involved in experiments. Light was switched on and off at fixed hours (on: 7.30 a.m and off: 8 p.m), all year round. Monkeys had free access to food pellets. They were also given fresh fruits and nuts. All cages were enriched with mirrors, hanging ropes, water pools, balls and foraging baskets. No procedure that might cause discomfort or pain was undertaken without adequate analgesia or anesthesia. In particular, each monkey underwent a single surgical session under gas anesthesia (Vet-Flurane à 0.5–2%) during which a craniotomy was made over the left (resp. right) prefrontal cortex for monkey Z (resp. M) and peek recording chambers were implanted to allow access to the FEF with microelectrodes. Post-surgery pain was controlled with a morphine pain-killer (Tamgesic, 0.01 mg/kg i.m.) and a full antibiotic coverage was provided (long action Tamgesic 100, one injection during the surgery and one 5 days later, 0.1 mg/kg, i.m.). The general health status of the animals was monitored every day by competent and authorized personal. In agreement with the 3R 'reduction' recommendation, the two animals involved in the present study were enrolled later in another experiment.

Description of the neurophysiological database

Behavioral task. The data analyzed in the present work were collected while monkeys performed a cued target detection task based on a rapid serial visual presentation (figure 1, see also [25], [29]). It allowed to dissociate in time the processes related to the orientation of attention from those related to target detection [30]. In particular, the cue was a non-spatial abstract cue that informed the monkey in which hemifield it should direct its attention. Briefly, the monkey had to fixate a central point on the screen throughout each trial. Two streams of visual objects were presented, one in the visual receptive field of the neuron being recorded and the other in the contralateral hemifield. One of the streams included a cue which instructed with a certain probability the position of the target. The cue could be green (resp. red), predicting that the target would appear in the same (resp. other) stream. In the following, the green cue will be called a *Stay* cue and red cue a *Shift* cue. The monkey had to combine the information related to the physical attributes of the cue (its location and its color) to find out where the target was likely to appear. The monkey had to release a lever to report the presence of the target. The target appeared on 80% of the trials. The remaining 20% no target trials were catch trials that served to discourage the monkeys from making false alarms. In target trials, the target appeared either 150 ms, 300 ms, 600 ms or 900 ms following the cue. In 80% of these trials (64% of all trials), the target appeared in the instructed stream (valid trials). In the remaining 20% target trials (16% of all trials), it appeared in the opposite stream (invalid trials). The monkey was rewarded for releasing the lever 150 to 750 ms following target onset on valid and invalid trials and holding it on catch trials. Invalid trials were used to check that the monkey used the predictive information provided by the cue in order to guide its behavior. Sessions in which this was not the case were discarded from the analysis.

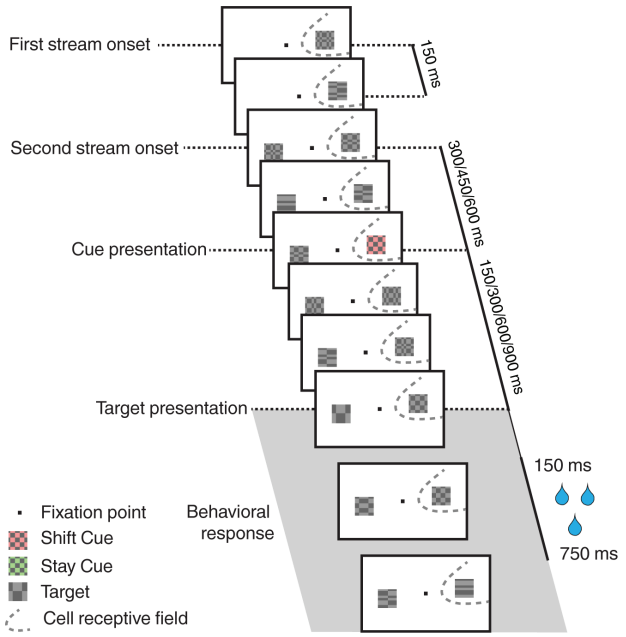

Figure 1. Task description. The experimental procedure is a cued-target detection based on a dual rapid serial visual presentation (RSVP) paradigm. The monkey is required to maintain its gaze on the central fixation point all throughout the trial. A first stream of stimuli, that is a succession of visual stimuli every 150 ms, is presented either within (as here) or opposite the fixation point from the cell's receptive field. Three hundred milliseconds later, a second stream appears opposite the first stream from the fixation point. Three hundred, 450 or 600 ms (here, 300 ms) following the second stream onset, a cue is presented within the first stream. This cue can be a green stay cue indicating to the monkey that the target has a high probability to appear within this very same stream or a red shift cue (as here), indicating that the target has a high probability to appear within the opposite stream. On 80% of the trials, the target is presented 150, 300, 600 or 900 ms from cue onset. On 80% of these target trials (64% of all trials), the target location is correctly predicted by the cue (valid target, as here). On 20% of these target trials (16% of all trials), the target location is incorrectly predicted by the cue (invalid target). On the remaining 20% of trials, no target is presented (catch trials), so as to discourage false alarms. The target is composed of just one horizontal and one vertical spatial cycle, while distractor items are composed of up to 6 horizontal and vertical spatial cycles. The monkey gets rewarded for responding by a bar release, between 150 and 750 ms following target presentation, and for holding on to the bar when no target is presented.
doi:10.1371/journal.pone.0086314.g001

Cell population. The spiking activity of 131 frontal eye field (FEF) neurons was recorded from two macaque monkeys. All procedures were approved by the local animal care committee in compliance with the guidelines of the European Community on Animal Care (cf. [25] for details). These cells were subjected to individual statistical analysis. Amongst them, a subset of neurons (n = 21) reliably encoded cue instruction while apparently providing no information about cue location or cue color (see [25] for details). These cells thus encoded the final position of attention, discriminating between cues instructing attention towards the receptive field (contralateral *Stay* cues and ipsilateral *Shift* cues) and cues instructing attention away from the receptive field (ipsilateral *Stay* cues and contralateral *Shift* cues). In the following, we will be comparing the performance of several classifiers at decoding the position of the initial visual stream of stimuli (exogenously driven visual information) to their performance at decoding the final position of attention instructed by the cue (endogenously driven cognitive information). The exogenous information was decoded using the entire FEF neuronal population while the endogenous information was decoded using either the entire population or the subset of cue instruction cells. This allows us to make two comparisons. Firstly, we can compare the decoding performance for exogenous versus endogenous information using the entire FEF neuronal population, secondly we can compare the decoding performance of endogenous information between the entire FEF neuronal population and the subset of cells that we previously identified as significantly modulated by the variable of interest.

Decoding procedure

Data pre-processing. For each cell and each trial, the spiking data was smoothed by averaging the spiking activity over 100 ms sliding windows (temporal resolution of 1 ms). This window width corresponds to a trade-off between performance and decoding speed, as narrower filtering windows result in a lower performance while wider filtering windows increase the delay of real-time decoding [29]. The 131 cells were combined to form a single neuronal population. To decode the first flow position, both correct and error trials were used, because the cells' response to this exogenous event does not depend on the monkey's engagement in the task. As a result, an average of 295 trials (s.d. = 107) was available per cell. In contrast, only correct trials were used to decode the position of attention, as error trials can arise from an improper orientation of attention. In addition, unless otherwise stated, trials in which the target appeared 150 ms after the cue were excluded from the data set to avoid a confound between cue- and target-encoding. As a result, fewer trials were available (mean = 112 trials, s.d. = 33). For each cell, 60 trials were randomly selected per condition (First flow on the left, First flow on the right, Attention instructed to the left and Attention instructed to the right). For most cells, these trials corresponded to a random subset of all the available trials per condition. For a minority of cells, some trials were randomly duplicated to achieve the requirement of 60 trials per condition. Since this can potentially induce an artificial inflation of decoding performance, we conducted random permutations following the exact same procedure as described in the data pre-processing section, in order to define the actual chance level; decoding performance was systematically compared to this chance level. Single trial responses were randomly combined across the entire neuronal population in order to create 60 virtual population responses to each event of interest. This procedure, defining a seed population activity, was repeated 20 times, thus defining 20 different population activity seeds (out of more than 131 to the power of 60 possible population activities, thus limiting the potential inflation induced by the duplication of some trials). Note that these population responses are free of the correlations that would be found in simultaneous recordings.

General cross-validation procedure. Visual and attention-related signals do not have the same temporal dynamics and their mean response peaks at different latencies from event onset. For both variables, the decoding was performed around this peak response. As a result, when decoding the position of the initial visual stream, we trained the classifiers on the smoothed activity observed at 125 ms following visual stream onset (i.e. on the 100 bin centered at 125 ms). When decoding the instructed position of attention, we trained the classifiers on the smoothed activity observed at 245 ms following visual stream onset (i.e. on the 100 bin centered at 245 ms). These timings correspond to the timing of the peak neuronal response to each specific event as estimated in Ibos et al. [25]. Due to a more complicated architecture, the

reservoir was trained using data from a time-window of 75 ms around these training references.

We trained the classifiers on 70% of the data (84 trials) and tested them on the remaining 30% of the data (remaining 36 trials) so that the testing is performed on a naïve set of trials, never experienced by the classifier. During *training*, the decoders were simultaneously presented with single-trial population activities (corresponding to the observed inputs) and the state of the decoded variable (corresponding to the associated outputs: Visual stream on the left or on the right or Attention instructed to the left or to the right). During *testing*, the decoders were presented with the successive test sets centered on a window of 100 ms around the time at which training was performed (i.e. one test set every 1 ms in this window) and produced their guess for the state of the decoded variable. The readout performance on each decoding run is then calculated by averaging the performance produced by the 100 successive testing sets (1 ms resolution) and corresponds to the percentage of trials on which the classifier provided the correct guess for the state of the decoded variable. This procedure was chosen to ensure that the final readout performance reflects a robust pattern of activity. This *training/testing* procedure was repeated for each data seed (i.e. 20 times in all, cf. data pre-processing section) to yield an average readout performance, using the exact same randomly constructed *training/testing* datasets for all decoders. Testing the decoders on a set of predefined seeds allows to discuss their readout performance independently of data variability.

Random permutation tests. Randomized permutation tests were performed for each classifier and for each analysis using the exact same procedure as above, after assigning, for each cell, randomized condition labels to each trial (using a random sampling with replacement procedure). This procedure, repeated 50 times, for each of the 20 data seeds, yielded the distribution of chance performance of each classifier. This distribution was thus constructed with 1000 data points. The readout performance of a given classifier was considered as significant when it fell in the 5% upper tail of its corresponding chance performance distribution (non-parametric random permutation test, $p<0.05$).

Classifiers

Optimal Linear Estimator (OLE). The linear regression (figure 2a–b) minimizes the mean square error for the following equation $\mathbf{C} = \mathbf{W}*\mathbf{R}$, where $\mathbf{R}$ is an n by t matrix of $\mathbf{R}_{ij}$, n being the number of cells in the neuronal population of interest, t the number of available trials and $\mathbf{R}_{ij}$ the neuronal response of cell i in the population, on trial j; $\mathbf{C}$ is a 1 by t vector, the sign of the elements of which describes the two possible classes taken by the binary variable of interest and $\mathbf{W}$ is a 1 by n vector corresponding to the synaptic weights that adjust the contribution of each cell to the final readout. This procedure defines a linear boundary between data points sampled from two independent distributions (figure 3a). As a result, such an estimator is optimal provided the neuronal output of the population activity is a linear sum of the inputs. This assumption appears to be a general property of neuronal populations (see [19], [31], [32], [1], [2], who suggest that neurons could form a set of basis functions encoding real-world variables). Such a linear decoding can be achieved in two ways:

Regularized Explicit function (R. OLE). The first approach is to inverse the above equation as $\mathbf{W} = \mathbf{C} *\mathbf{R}^{\dagger}$, noting $\mathbf{R}^{\dagger}$ the Moore-Penrose pseudo-inverse of $\mathbf{R}$. $\mathbf{R}^{\dagger}$ was determined on a subset of the data (Figure 2a, A_{train} = training dataset) and the resultant $\mathbf{W}$ matrix was applied to solve $\mathbf{C} = \mathbf{R} * \mathbf{W}$ on the rest of the data (A_{test} = testing dataset). As the Moore-Penrose pseudo-inverse leads to overfitting, we used a Tikhonov-regularized version of it: this solution minimizes the compound cost norm($\mathbf{W}*\mathbf{R} - \mathbf{C}$)+ λ*norm($\mathbf{W}$), where the last term is a regularization term added to the original minimization problem [33]. The scaling factor λ was chosen to allow for a good compromise between learning and generalization. Its precise value was optimized for each analysis as this value depended on the population size and number of training trials (see [34] for the λ optimization procedure).

Artificial Neural Network (ANN OLE). To estimate the penalty (or benefits) of training artificial neural networks, we compared the formal *OLE* solution described above with the performance of a one-layer feed-forward network with as many units in the input layer as in the FEF cell population of interest, one unit in the output layer reflecting the class of the binary variable of interest and a hyperbolic tangent transfer function (see Figure 2b, [1], [2]). Training was performed using a quasi-Newton back-propagation that defines the weight vector $\mathbf{W}$ which minimizes the square distance between the estimate of the state of the variable of interest and its actual value. To prevent overfitting, a regularization procedure was used. This procedure modifies the initially chosen network performance function (the mean of sum of squares of the network errors) by adding an additional regularization term. The regularization term consists of a weighted mean of the sum of squares of the network weights and biases. As a result, the modified performance function *msereg* becomes: $msereg = \lambda*mse + (1-\lambda)*msw$, where *mse* is the mean square error and *msw* is the mean square weight. The factor λ sets the performance ratio between the mean square error and the mean square weight. Here, equal weight was given to both the mean square error and the mean square weights ($\lambda = 0.5$) as this value yielded the highest decoding performances. The sign of the classifier output described the possible states of the variable of interest (-1 and 1).

Non Linear Artificial Neural Network Estimator (ANN NLE). The *OLE* described above cannot, by definition, capture non-linear processes, which might be at play in prefrontal cortical regions and/or during cognitive endogenous processes. We thus decided to implement a non-linear estimator. If the *ANN NLE* outperformed the *ANN OLE*, this would support the presence of non-linear neuronal information processes. The *ANN NLE* is implemented similarly to the *ANN OLE* above, except that a second layer is added to the network architecture in order to capture potential hidden non-linearities in the neuronal population response. This additional hidden layer has half as many units as the input layer (Figure 2c). Such a two-layer network architecture draws a non-linear boundary between data points sampled from two independent distributions (figure 3b).

Bayesian classifier. We used a Gaussian naïve Bayes classifier [35], [36] which directly applies Bayes' theorem (Figure 2d) to calculate the conditional probability that the population response, $\boldsymbol{R}$ is of class C_k: $P(C_k|\boldsymbol{R})$. Cells are "naïvely" assumed statistically independent. Bayes' theorem can be written for cell *n* as follows:

$$P(C_k|R_i) = \frac{P(C_k) * P(R_i|C_k)}{P(R_i)} \qquad \text{(eq.1)}$$

$P(R_i)$ can be ignored, since it is constant and independent of C_k. $P(C_k)$ is also constant across the different classes by design (the two classes are equi-probable). As a result

$$P(C_k|R_i) = \alpha P(R_i|C_k) \qquad \text{(eq.2)}$$

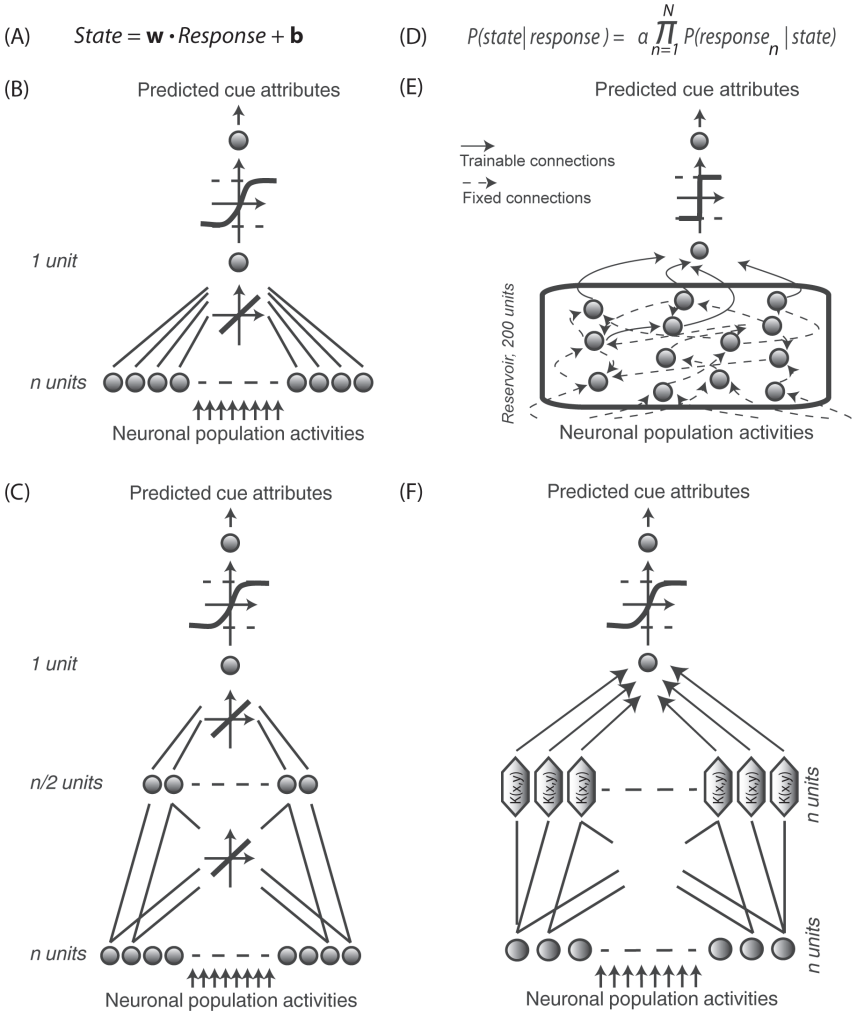

Figure 2. Decoders. (A) Regularized OLE, the training step is a simple regularized linear regression. (B) Optimal Linear Estimator (ANN OLE), implemented as a one-layer feedforward artificial neural network. The input layer has one unit per FEF cell and receives instantaneous population neuronal activities. The output layer contains 1 unit. Training involves optimizing the weights using a Levenberg-Marquardt backpropagation algorithm and a hyperbolic tangent transfer function. (C) Non-Linear Estimator (ANN NLE), implemented as a 2-layer feedforward artificial neural network. The network architecture only differs from the OLE by an additional hidden layer with n/2 units, n being equal to the number input units. (D) Bayesian decoder, applying Bayes' theorem to calculate the posterior probability that state i is being experienced given the observation of response r. (E) Reservoir decoding. The decoder has one input unit per FEF cell and one output unit. Fixed connections are indicated by dotted arrows and dynamical connections are indicated by full arrows. The reservoir contains 200 units. The recurrent connections between them are defined by the training inputs. A simple linear readout is then trained to map the reservoir state onto the desired output. (F) Support Vector Machine (SVM), the LIBSVM library (Chih-Chung Chang and Chih-Jen Lin, 2011) was used (Gaussian radial basis function kernel so as to map the training data into a higher dimensional feature space). The transformed data is then classified with a linear regressor and training is performed with a 5-fold cross-validation. For all decoders, the sign of the output corresponds to the two possible states of the variable being decoded.
doi:10.1371/journal.pone.0086314.g002

where α is constant across the different classes C_k. If the components R_i of $\mathbf{R}$ are independent,

$$P(C_k|R) = \alpha \prod_{i=1}^{n} P(R_i|C_k)$$

and the Bayesian classifier is optimal, in the sense that it intrinsically minimizes the misclassification rate. Indeed, misclassification is minimized if and only if the response $\mathbf{R}$ is assigned to the class C_k for which the $P(C_k|\mathbf{R})$ is maximum [37]. As a result, the Bayesian decoding procedure amounts to using $f_k(\mathbf{R}) = P(\mathbf{R}|C_k)$ as discriminant function. We estimated the conditional probability density of the neuronal response R_i of a given neuron $P(R_i|C_k)$, given a stimulus class C_k, as a Gaussian distribution, taking as parameters the mean and the standard deviation of the neuron's response across trials. The resulting Bayesian classifier draws a quadratic non-linear boundary between data points and takes into account the variance structure of the input distributions (figure 3c, distinct-variance Gaussian Bayesian model). This is equivalent to a discrete maximum likelihood method in that it calculates, for each trial, the probability of each class and chooses the class that presents the highest probability.

Reservoir Computing. We used a specific class of recurrent neural networks derived from *Reservoir computing*. In such a design, the dynamics of the neurons of the reservoir map the input onto a higher dimensional space, thus unveiling potential hidden

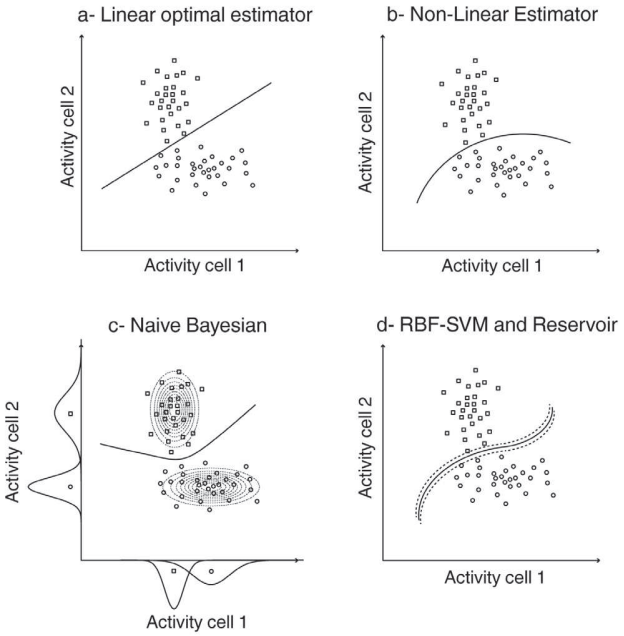

Figure 3. Decision boundaries for the different classifiers. Each plot represents the activity of a hypothetical cell 1 as a function of the activity of hypothetical cell 2, on successive trials, in response to a stimulus 1 (circles) or 2 (squares). a) Optimal linear estimator; b) non-linear estimator; c) naive Bayesian. The dotted ellipsoids (Bayesian) correspond to the probability-density fitted Gaussian distributions of the cells' activities for each stimulus; d) SVM with Gaussian kernel (RBF) and Reservoir. In the case of SVM, the dotted line corresponds to the margin around the decision boundary.
doi:10.1371/journal.pone.0086314.g003

contingencies. The simple readout process is then trained to map the overall state of the neuronal reservoir onto the desired output [38], [39]. Because of the higher dimensionality mapping achieved by the reservoir, such a recurrent neuronal network is expected to yield a better read out performance than a simple direct linear mapping (OLE) between the input and the desired output. In particular, it allows to segregate the data points sampled from two independent distributions thanks to a non-linear boundary that minimizes the mean square error in the higher dimensionality space the input data is projected on. Specifically, we used a recurrent neural network (RNN, figure 2e) with fixed connections and a readout layer that reads the activity of all neurons in the RNN [40]. All parameters specific to the reservoir were set with a grid-search procedure prior to the decoding experiments in order to optimize the decoding performance. This procedure consisted in testing the decoding performance of the reservoir over a large set of parameters and selecting those parameters that maximize correct classification. Due to heavy and time costly computations, these parameters (number of nodes, transfer function, scaling factor, input sparseness, reservoir sparseness, spectral radius, time constant and regularization parameter) were optimized only for full population- and trial-sizes. For all analyses, unless otherwise stated, the nonlinear optimal reservoir contained 500 analog nodes without transfer function. The fixed connections between the input units and the reservoir were randomly generated from a uniform distribution between 0 and 1 and scaled with a factor of $10^{-1.2}$ in order to balance how strongly the reservoir is driven by the input data. This optimal reservoir had no interconnections between its nodes. The nodes were initially set as leaky integrator, but optimization of their time constant revealed that the network performs better without leaky integration. As a result, such a reservoir is equivalent to a completely non-dynamic neural network using independent non-linear transformations to calculate the decoding performance. A Tikhonov regularization procedure was chosen in order to avoid overfitting. The readout layer performs an explicit linear regression between the activity of the neurons within the RNN and the desired output.

Reservoir with memory. Recurrent networks like the reservoir have been used to process temporal information such as time series. Here, we wanted to test whether the reservoir could extract temporal information embedded in the data and provide a stationary decoding performance that memorizes the decoded event for a longer period of time. To do this, new parameters were set in a grid-search manner (as described above) in order to optimize the decoding performance for a training window of 70 ms to 500 ms after cue onset. The non-linear dynamic reservoir contained 500 analog nodes with a hyperbolic tangent transfer function. The fixed connections between the input units and the reservoir were randomly generated from a uniform distribution between 0 and 1 and the scaling factor was set to $10^{-3.8}$. There were no interconnections between the nodes within the reservoir and the time constant was set to 55. These parameters created a non- dynamic reservoir that, because of the high time constant, uses previous time-steps to extract information. The readout layer performed an explicit linear regression between the activity of the neurons within the RNN and the desired output.

Support vector machine (SVM). The basic SVM can be considered as a non-probabilistic binary linear classifier that maps the inputs in space so as to maximize the separation between the inputs of the two classes ([41], figure 2f). The input data is nonlinearly mapped to a higher-dimensional feature space and then separated by a maximum margin hyperplane. Generally, this maximum margin hyperplane corresponds to a non-linear decision boundary in the input space, defined by the following equation (Eq3)

$$d(\vec{\mathbf{R}}_{.p}) = \mathrm{sign}\left(\sum_{j=1}^{t} C_j \alpha_j \langle \phi(R_j).\phi(R_p)\rangle + b\right) \qquad (eq.3)$$

where $d(\vec{\mathbf{R}}_{.p})$is the decision on the test neuronal population response $\vec{\mathbf{R}}_{.p}$; t is the total number of training trials; the class labels $Cj \in \{-1,+1\}$ and represent the states of the binary output variable during training; α_j represents a set of t constants that define the SVM optimal solution for the training set; the input data vector $\vec{\mathbf{R}}_{.j}$represents the population neuronal response on trial j. The decision boundary is fully defined by a subset of training samples, the so-called support vectors, but is never explicitly calculated. Mercer's theorem states that for each continuous positive definite function, $K(\mathbf{x}, \mathbf{y})$, there exists a mapping $\mathbf{\Phi}$ such that $K(\mathbf{x}, \mathbf{y})$ equals the dot-product, $\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y})\rangle$ for all $\mathbf{x}, \mathbf{y} \in R^n$. Mercer's theorem allows to learn the relationship between $\mathbf{x}$ and $\mathbf{y}$ in the feature space without an explicit estimation of the mapping function $\mathbf{\Phi}$, by simply using a kernel function; this makes the support vector machine efficient for operating in a high-dimensional feature space [42], [43]. The architecture of the SVM decoder we use here is presented on figure 2f (LIBSVM library, Gaussian kernel implantation, [44], http://www.csie.ntu.edu.tw/~cjlin/libsvm). Note that we used a SVM design with a Gaussian kernel, $K(x,y) = \exp(-\gamma||x-y||^2)$. Overall, because the input data is projected onto a higher-dimensional feature space, SVM allows segregating the data points sampled from two independent distributions thanks to a non-linear boundary (figure 3d). A grid search procedure (calculating decoding

performance over a range of cost and gamma SVM parameters) was performed, for each set of train data, prior to the decoding procedure, in order to find the SVM parameters that maximize decoding performance. This was done using a 5-fold cross-validation procedure so as to minimize over fitting. Specifically, each training set was randomly divided into 5 parts. One part was retained for testing the model while the other 4 parts were used for the training of the grid search procedure. This procedure was repeated 5 times so that each part is used exactly once to evaluate the selected parameters.

Results

Though the mathematical properties of the classifiers considered in the present work are well described, how they behave and how they differ when applied to real neuronal population activities has not been investigated this far. In particular, no study has directly questioned how their performance is affected by actual biological noise in the data, and how it differs between sensory and cognitive signals. In the following, we examine the performance of different classifiers and their dependency on several parameters that often turn out to be crucial in the context of single cell recording experiments. We first compare the decoders' performance as a function of the variable being decoded (visual/exogenous versus attentional/endogenous). We then evaluate the dependency of each decoder on the number of available training trials and the number of available cells. Last, we quantify the impact of unbalanced training samples, i.e. samples with unequal number of trials for each decoded class.

Who's best? Comparing readout performance across classifiers

A straightforward measure of how well a decoder extracts information from population neuronal activities is its readout performance, i.e. its correct classification rate. We thus compared the average performance of each classifier (*SVM, Reservoir, regularized OLE, Bayesian, ANN OLE and ANN NLE*) over 20 successive decoding runs (each performed on a distinct data set, cf. data seeds in the methods section) when decoding either the position of the first visual flow (figure 4a, light gray bars), or the instructed position of attention (figure 4a, dark gray bars) from the whole FEF population ($n = 131$). The different classifiers did not perform equally well and this, irrespectively of whether the position of the first visual flow or the instructed position of attention was being decoded (2-way repeated measure ANOVA, Variable x Classifier, Classifier main factor, $p<0.001$, figure 4a). A Bonferroni post-hoc analysis indicated that the *SVM*, the *regularized OLE*, the *Reservoir* and *the ANN OLE* significantly outperformed *the Bayesian and the ANN NLE* ($p<0.001$) both when decoding position of the first visual flow ($p<0.01$) and the instructed position of attention ($p<0.001$).

However the 95% confidence interval (as estimated by a non-parametric random permutation test, $p<0.05$, cf. methods) that served as a decision boundary for significantly above chance performance varied from one decoder to the other. We calculated, for each classifier, its performance relative to this 95% confidence upper limit (figure 4b). As was the case for the absolute readout performance, the relative readout performance also varied across classifiers (2-way repeated measure ANOVA, Variable x Classifier, Classifier main effect $p<0.001$, figure 4b). Here, a Bonferroni post-hoc analysis indicated that only *the SVM* and *the regularized OLE* significantly outperformed the other classifiers ($p<0.001$) both when decoding the position of the first visual flow ($p<0.05$) and the instructed position of attention ($p<0.001$, accompanied here by *the reservoir* and *the ANN OLE*). This difference between the absolute performance and the present relative performance analyses was due to the higher 95% confidence limit of the *Reservoir* and *the ANN OLE* as compared to that of the *SVM* and the *regularized OLE*.

Who's best? Comparing the readout performance for endogenously driven vs. exogenously driven neuronal information

The next question we sought to answer is whether the performance of classifiers on exogenous information is predictive of their performance on endogenous information. We thus compared the performance of the different classifiers at decoding, from the whole FEF population, either the spatial position at which the first stream was presented (exogenous, figures 4a–b, light gray bars) or the position at which attention was instructed by the cue (endogenous, figures 4a–b, dark gray bars). All decoders (*SVM, Reservoir, regularized OLE, Bayesian, ANN OLE or ANN NLE*) provided both a better absolute and relative readout of the exogenous variable as compared to the endogenous variable (2-way ANOVA, Variable x Classifier, Variable main factor, $p<0.001$, figures 4a–b). Specifically, the average absolute decoding performance of first stream position over all decoders (mean $= 93.0\%$, s.e. $= 5.2\%$) was 16 percent higher than the average absolute decoding performance of the instructed position of attention (mean $= 77.0\%$, s.e. $= 1.6\%$). Likewise, the average relative decoding performance of first stream position over all decoders (mean $= 33.8\%$, s.e $= 1.8\%$) was also 16 percent higher than the average relative decoding performance of the instructed position of attention (mean $= 17.7\%$, s.e $= 1.8\%$).

Most of FEF neurons encode visual information, while only a small proportion of cells encode the instructed position of attention (16%, [25]). This could account for the higher performance obtained at decoding the exogenous information as compared to the endogenous information. Alternatively, this difference could be due to a noisier encoding of endogenous variables by cortical neurons as compared to how exogenous information is encoded (or more broadly speaking, to different cortical encoding schemes as a function of the variable being considered). In order to address this issue, we performed two additional analyses: 1) we evaluated the decoders' performance at reading out the instructed position of attention from a subset of FEF cells characterized by a statistically significant cue-instruction related response ($n = 21$), and 2) we evaluated the *SVM's* performance at decoding the first visual stream position from a random selection of 21 visual cells. We then compared the performance a) between the two conditions (population size hypothesis), and b) between the first condition and when using the whole neuronal population (population selectivity hypothesis).

Population size hypothesis. In order to test whether population size fully accounts for the difference in performance between the readout of first visual stream position and the readout of the instructed position of attention, we proceeded as follows. We identified, within the whole FEF population, the visually responsive neurons ($n = 111$, significant visual modulation within 150 ms from first stream onset for 30 ms out of 25 ms, t-test, $p<0.05$). We randomly selected 21 visual neurons from this pool of 111 visual neurons and we calculated the average performance of the *SVM* at reading out the first visual stream position from this small population over 20 successive decoding runs. This procedure was repeated 20 times so as to have an estimate of the influence of the cell sampling on the readout performance. Such a procedure yields an absolute average readout performance of 79.5% (s.e. $= 0.2\%$; relative mean performance $= 22.6\%$, s.e. $= 0.19\%$

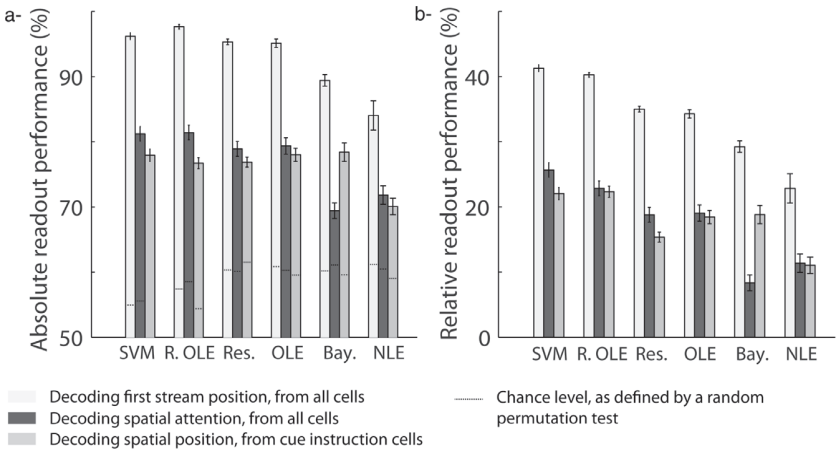

Figure 4. Comparison of mean performance at reading out first stream position and spatial attention across classifiers. A) Absolute readout performance. The dashed lines indicate the chance level for each condition, as estimated by a random permutation test ($p<0.05$). B) Readout performance, relative to chance level. The flow position is decoded using all cells in the population (light gray). Spatial attention is decoded using all cells in population (dark gray) or using only cells with significant individual attention-related responses (intermediate gray). The mean readout performance and the associated standard error around this mean are calculated over 20 decoding runs. SVM = support vector machine, Res. = reservoir, R. OLE = regularized OLE, Bay. = Bayesian, NLE = ANN non-linear estimator, OLE = ANN optimal linear estimator.
doi:10.1371/journal.pone.0086314.g004

figure 5, light gray bar). Both the absolute and relative average readout performance of attention position from the cue-instruction selective FEF cells fell within the range of readout performance of first flow position from a random small FEF population(absolute performance: $p = 0.09$, mean = 77.9%, s.e. = 0.90%; relative performance: $p = 0.53$, mean = 22.0%, s.e = 0.92%, figure 5, dark gray bar). The smallest readout performance of first visual stream position obtained from the different random samples of FEF visual cells was 65.9% while the highest performance was 92.9% (figure 5, dotted line on light gray bar). This demonstrates that for small populations, performance is highly dependent on the population sample. This applies to the decoding of first visual stream position and most probably also to the decoding of the instructed attention location.

Population selectivity hypothesis. The readout performance at decoding the instructed position of attention was estimated from a subset of cells, individually encoding the final cue instruction (21 cells). As for decoding the instructed position of attention from the entire FEF population, all decoders did not perform equally well (one-way ANOVA with repeated measures, decoder main factor, $p<0.001$, figures 4a–b, medium gray bars). A Bonferroni post-hoc analysis indicated that *the SVM, the regularized OLE, the reservoir*, *the ANN OLE* and *the Bayesian* classifiers outperformed *the ANN NLE* (absolute performance: $p<0.001$, figure 4a; relative performance: $p<0.05$, figure 4b). In addition, decoding the instructed position of attention from the whole FEF population or from a selected subset of cells did not affect the readout performance of all decoders in the same way (two-way ANOVA, significant interaction between the two populations and decoder main factors, $p<0.001$). A Bonferroni post-hoc analysis revealed that this population effect is specific to the *Bayesian decoder* both for the absolute and the relative performances ($p<0.001$, figure 4a–b), the absolute readout performance of this classifier being 9.0% higher when the decoding is performed on the selected subset of cells than when it is performed on the entire the FEF population.

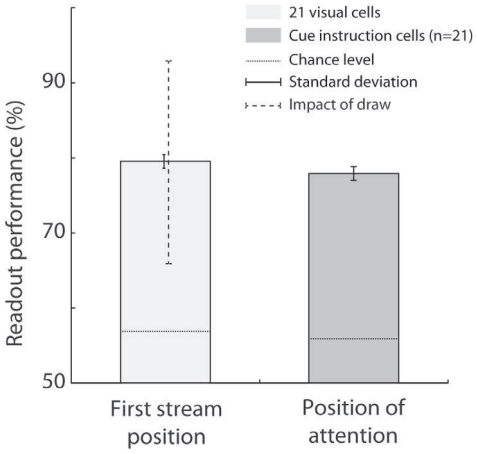

Figure 5. Comparison of decoding flow onset (light gray) with 21 visual cells versus decoding spatial position of attention (dark gray) with the 21 cells with significant individual attention-related responses. The mean read-out performance across 20 runs is showed with standard deviation around this mean. The dotted line corresponds the maximum- and minimum performance across 20 draws of 21 visual cells out of 111. The SVM classifier was used. The mean readout performance and the associated standard error around this mean are calculated over 20 decoding runs. Chance level is defined using a random permutation procedure ($p<0.05$).
doi:10.1371/journal.pone.0086314.g005

Trade-off between population size and population response sampling

Two parameters are expected to drastically influence readout performance: population size (i.e. the number of cells which are simultaneously being recorded from) and population response sampling (i.e. the number of trials on which the training is performed). In the following, we consider sequentially the impact of each parameter in conjunction and then independently so as to gain a better understanding of the contribution of each of these

two parameters onto decoding performance. The *ANN NLE* was excluded from all further analysis due to its extremely time costly computations (~6 hours per data seed/run) combined with a relatively poor readout performance (Regularized OLE/Bayesian: less than 1 second per data seed per run; ANN OLE: less than 2 seconds per data seed per run; SVM/Reservoir: less than 3 seconds per data seed per run; note that these time estimates are both dependent on the type of processor being used and on the optimization of the computation scripts).

Population size and trial number trade-off. In order to further explore the trade-off between trial number and population size when decoding the instructed position of attention from randomly selected FEF cells, we performed an additional analysis in which we co-vary both parameters simultaneously. This analysis is performed on the best performing classifiers, namely, the regularized OLE (figure 6a), the SVM (figure 6b) and the reservoir (figure 6c). On all plots, we indicate both the 65%, 70% and 75% performance iso-contours (figure 6, black contours) and the 95% confidence limits for significant readout (figure 6, gray contours). Confirming our previous observations, the regularized OLE achieved the best readout performance at all population sizes and training trial number combinations. In particular, a 75% absolute performance rate was achieved with as few as 60 cells and as little as 40 training trials. The SVM came next, followed by the Reservoir, although the latter appears to outperform the former for small trial numbers and small population size.

Population size. The readout performance at decoding the instructed position of attention from the entire FEF population steadily increased as a function of population size for all decoders (Figure 7a). For populations of less than 25 randomly selected FEF neurons, *SVM*, *Reservoir*, *regularized OLE* and *ANN OLE* provided equivalent readout success rates, outperforming *the Bayesian* classifier. As the number of neurons in the population increased, *the SVM*, *the regularized OLE* and *the reservoir* improved their performances similarly whereas *the ANN OLE* improved with a slower rate. The *Bayesian* was trounced by all the others and the impact of increasing the population size onto its readout performance was the lowest.

Absolute readout performances above the upper 95% confidence limit are indicated, in figure 7a, by a thicker line. It is interesting to note that *the SVM* and *the regularized OLE* had an absolute performance significant with as few as 4 random FEF cells. The *reservoir* achieved significant readout performances with 9 cells, whereas *the Bayesian* required 22 cells in the population.

When decoding was performed on the subset of attention selective FEF cells (n = 21, figure 7b), the overall effect of population size on readout performance was equivalent across decoders, except for the fact that the *regularized OLE* improved with a slightly slower rate as the population size increases (Figure 7b). As expected by their high attention-related information content, adding an attention-cell to the population induced an average increase of 0.76% on the readout performance (Figure 7b). This is to be contrasted with the impact of a randomly selected cell onto the overall population performance (0.17% increase in readout performance, figure 7a).

Trial number. As for population size, the readout performance at decoding the instructed position of attention from the entire FEF population steadily increased as a function of the number of available trials on which to train the decoders (Figure 7c). However, all decoders did not behave equally in the face of trial number. In particular, the regularized OLE outperformed all other decoders at all values of training trial number. This classifier actually reached significant decoding rates with as few as 10 trials (thick green line, figure 7c). The performance of the Reservoir, SVM and ANN OLE decoders became statistically significant around 20 training trials and stabilized for 30 trials or so (thick lines, figure 7c). While the SVM achieved the best readout performance amongst these three, the Bayesian decoder was outperformed by all the other classifiers at all training trials number and required more than 35 trials to achieve significant readouts.

When the decoding of instructed position of attention was performed on the subpopulation of attention-selective FEF cells, the impact of number of trials was drastically reduced (figure 7d). Indeed, the *regularized OLE*, the *SVM* and the *reservoir* achieved significant readout rates and are close to their maximum decoding performance with as few as 15 cells. The rise to maximum performance was slower for the *ANN OLE* and the *Bayesian* classifiers, and here again, this latter decoder required more data samples to achieve significant readouts (more than 30 trials).

Training sample balance

In an online-decoding environment, training is ideally performed on a fixed number of past trials in reference with the testing time-point. The assumption that these fixed trials equally

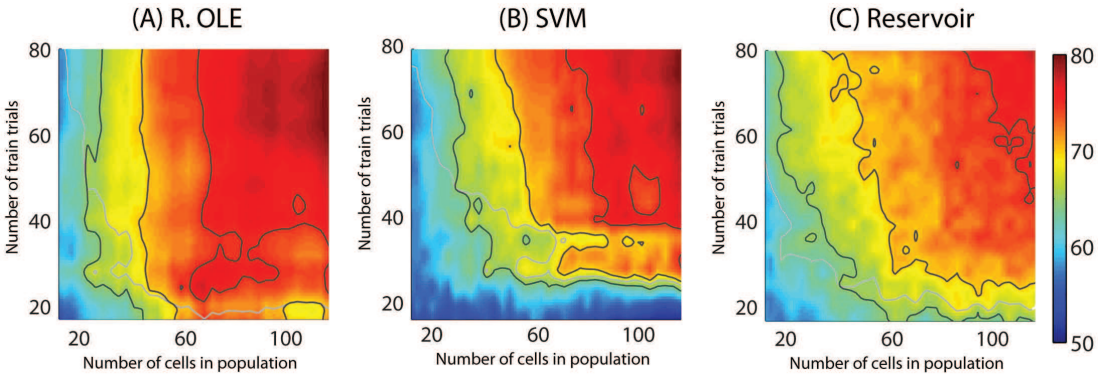

Figure 6. Decoding of spatial attention from the whole FEF population activities as a combined function of number of trials and cells with (A) Regularized OLE, (B) SVM and (C) Reservoir decoders. The black contour lines correspond, from yellow to dark red regions, to 65, 70 and 75% of readout performance. The gray contour lines corresponds to chance level as calculated, at each point, by a random permutation test ($p<0.05$). Smoothing with Gaussian kernel of 7. The readout performance is an average readout performance on 10 decoding runs. The maximum possible number of training trials is 84 trials. The y-axes are truncated at 80 trials.
doi:10.1371/journal.pone.0086314.g006

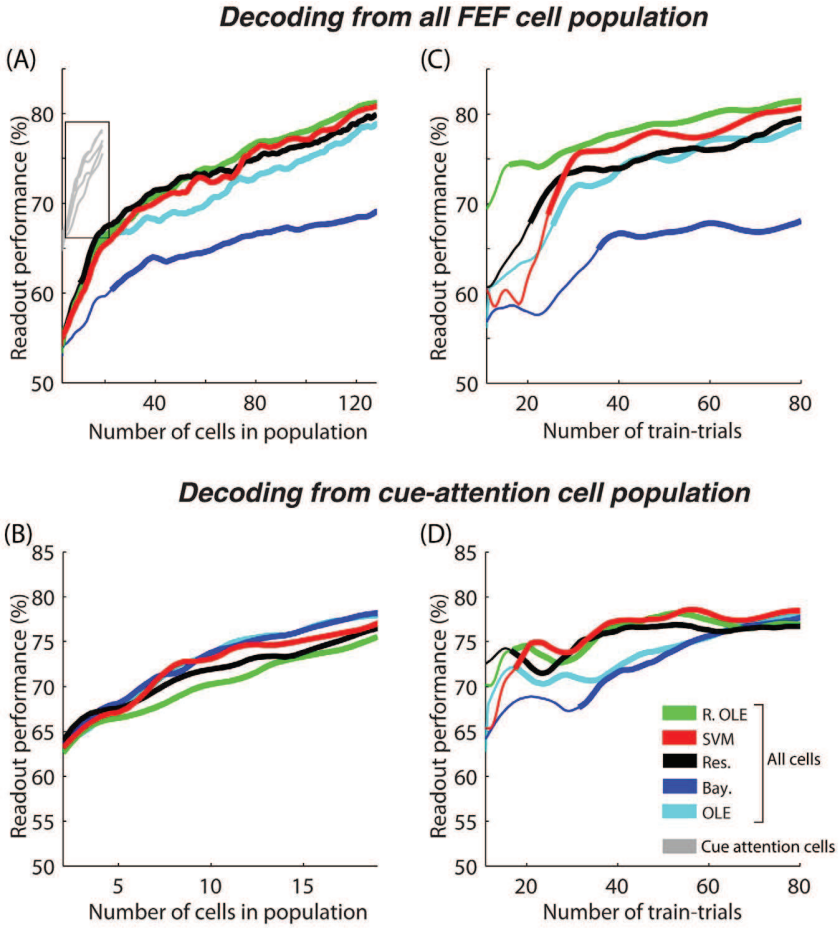

Figure 7. Decoding spatial attention (A–B) as a function of cell population size and (C–D) number of trials available for training. In (A) and (C), decoding is performed on the whole FEF cell population while in (B) and (D), decoding is performed only on the attention-related cells - presented also in gray in (A). The mean readout performance is calculated over 20 decoding runs. Thick lines indicated values that are significantly above chance as calculated using a random permutation test ($p<0.05$). SVM = support vector machine, Res = reservoir, Ex. OLE = explicit OLE, Bay. = Bayesian, NLE = ANN non-linear estimator, OLE = ANN optimal linear estimator.
doi:10.1371/journal.pone.0086314.g007

represent condition 1 (here, attention instructed to the left) and condition 2 (attention instructed to the right) might actually be violated, in particular due to a potential bias in the performance of the subject, having a higher performance for one condition over the other. Here, we explored the impact of such an imbalance in the number of training trials for the two states of the variable of interest (figure 8). The overall picture is that this imbalance incurs a drop in average readout performance. This drop in performance increased as the imbalance between the number of trials for the two conditions increased. The rate at which the performance decreased highly depended on the classifier. The *Bayesian* and the *ANN OLE* performed best with a respective performance drop rate of 3% and a 5% for a 50% imbalance in the data set (i.e. when one class has half as many trials as the other class). Furthermore, the *Bayesian* and the *ANN OLE* were the only classifiers for which performance remained above the upper 95% confidence limit at 50% imbalance. In comparison, the *SVM* had an 18% performance drop rate, the *regularized OLE*, a 28% performance drop rate and the *reservoir, a 30%* performance drop rate. There thus appears to be a trade-off between decoding performance in ideal settings and resistance to actual real data biases as considered here.

Memory

All previous decoding procedures relied on the estimation of the readout performance from population activities averaged over successive 100 ms windows, irrespectively of the response that was produced by the population at previous time points. However, recent evidence suggests that reverberating activities in local neuronal populations allows to maintain as well as to accumulate information in time [9], [45]. The specific *Reservoir* architecture allows us to directly assess the impact of information maintenance and accumulation over time by simply presenting the network with training data sampled over a longer time interval (70–500 ms after cue onset–figure 9, dark gray curve- versus 212–283 ms–figure 9, light gray curve) while still testing over successive 100 ms intervals (dark and light gray curves respectively, figure 9). In this analysis the classifier is tested on all time points ranging from 70 to 500 ms after cue presentation and each readout performance corresponds to the exact performance for that time point (i.e. in contrast with the previous measures, we do not average the readout performance over a 100 ms window). In this analysis, trials in which the target appeared 150 ms or 300 ms after cue onset have both been excluded to avoid the potential confound between cue and target-related activities. Readout performances above the upper 95%

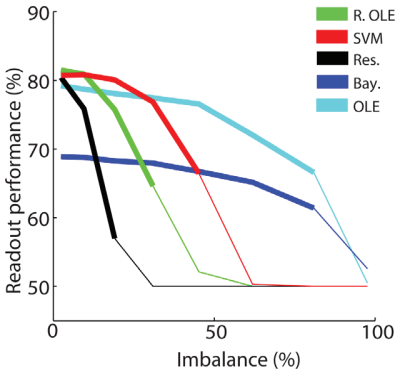

Figure 8. Impact of imbalance in the training set. The y-axis represents the difference between the readout performance of a balanced data set (same number of trials for each condition) and that of an unbalanced data set (more trials in condition 1 than in condition 2). The x-axis represents the degree of imbalance in training trial number between the two conditions. The mean readout performance and the associated standard error around this mean are calculated on 20 decoding runs. Thick lines indicated values that are significantly above chance as calculated using a random permutation test ($p<0.05$). SVM = support vector machine, Res = reservoir, R. OLE = regularized OLE, Bay. = Bayesian, NLE = ANN non-linear estimator, OLE = ANN optimal linear estimator.
doi:10.1371/journal.pone.0086314.g008

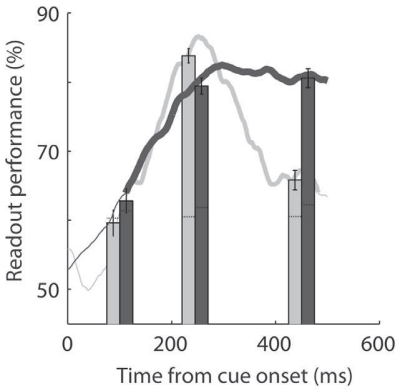

Figure 9. Impact of memory on Reservoir decoding performance on reading out the spatial position of attention. The light gray curve and bars corresponds to a reservoir training on a window of 75 ms around 245 ms after cue onset (as in all previous figures). The dark gray curve and bars corresponds to a reservoir training a larger time window (from cue onset at 0 ms to 700 ms post-cue). Decoding is performed on all FEF cell population activities. The bars show the mean readout performance and the associated standard error around this mean obtained by testing activities in a time window of 100 ms around the time reference point for training (245 ms after cue onset, N = 20 decoding runs). The curves show the mean readout performance and the associated standard error around this mean for each time point. Thick lines indicated values that are significantly above chance as calculated using a random permutation test ($p<0.05$).
doi:10.1371/journal.pone.0086314.g009

confidence limit are represented with a thick line. As expected, taking into account a longer period of time when training the *reservoir* network resulted in an increased decoding performance throughout the post-cue period, that was sustained at a distance from the cue (400–500 ms post-cue, dark gray curve). Taking into account the temporal structure of the signals however lead to a 5% drop in readout performance at the time of maximum attention-related population activity (245 ms following cue onset). As a result, this decoding approach is only interesting when the ability to track the information over time is more important than achieving maximum decoding performance.

Discussion

Our results suggest that endogenous information such as the orientation of attention can be decoded from the FEF with the same accuracy as exogenous visual information. In addition, all classifiers did not behave equally in the face of population size and heterogeneity, the available training and testing trials, the subject's behavior and the temporal structure of the variable of interest. In most situations, the regularized optimal linear estimator and the non-linear Support Vector Machine classifiers outperformed the other tested decoders.

Decoding of endogenous information as compared to exogenous information

Our decoders achieve, on average, a 19% higher performance at decoding exogenous information (here, the position of the first visual stream) from a heterogeneous FEF neuronal population, as compared to endogenous information (here, the position of attention instructed by the cue). These observations are in line with a previous study also showing a higher accuracy at decoding the position of a visual cue (*SVM* classifier, 100% accuracy, [4]) as compared to decoding the position of attention away from cue presentation (*SVM* classifier, 89% accuracy), from a heterogeneous FEF population. In the current study, we further show that this advantage at decoding exogenous over endogenous information is constant across both linear and non-linear classifiers. This could be due to the fact that FEF contains more visual-selective than attention-selective cells (*Cell selectivity hypothesis*). Alternatively, it could be that visual information is encoded in the FEF with a higher reliability than attention-related information (*Response reliability hypothesis*). While we cannot favor one possibility over the other, both are worth considering.

Cell selectivity hypothesis. The frontal eye fields are known to have strong, short latency visual responses [24], due to direct as well as indirect anatomical projections from the primary visual cortex V1 [46], [47]. Early studies report that up to 47% of FEF neurons are visually responsive [48] while up to 80% of pre-saccadic FEF neurons are also visually responsive [23]. In the dataset used in the present work, FEF neurons were recorded on the basis of their responsiveness to the key events of the cued-target detection task. Eighty-four percent of these neurons had significant neuronal responses to first visual stream onset (111 visual neurons out of a total of 131 neurons).The frontal eye fields are also known to be at the source of covert attention signals [49], [26]. And indeed, FEF neurons have been shown to encode spatial attention signals. The proportion of such FEF neurons varies from one study to another, most probably due to the specificities of the behavioral task being used. For example, in classical cued-target detection tasks that allow to manipulate spatial attention, the spatial mapping between the cue and the subsequent covert attentional orientation changes. The cue can be a spatial cue, indicating that attention should be held at the location where it is presented. In this case, there is a direct mapping between the location of the cue and the instructed position of attention and about half FEF neurons are shown to represent this latter information (40.8% in [50]; 51.8% in [4]). The cue can be a symbolic cue that requires to be interpreted so that the instructed location of attention can be extracted, for example, a central cue that instructs attention to the right if of a specific type (e.g. red or right pointing arrow), and to

the left if of another type (e.g. green or left pointing arrow). In this case, the spatial location of the cue is irrelevant to define the final position of the cue, while its identity is fully informative. Gregoriou et al. [51] report that 44.7% of FEF neurons were modulated by spatial attention in such a task. A more complex situation is the one used in Ibos et al. [25], in which the spatial location and the color of the cue are non-informative if considered separately, but fully informative if combined. This complex transformation most probably accounts for the lower proportion of attentional neurons available in the present dataset (16%, 21 out of 131, [25]). Overall, the proportion of visual and attentional FEF neurons thus appears to vary from one study to another, depending on the specific tasks being used and the associated recording biases.

Focusing on the present dataset (84% of visual cells and 16% of attention-related cells), the better readout performance at decoding the position of the first visual stream from the entire FEF population as compared to the decoding of the instructed position of attention could be due to the fact that more cells contribute to the encoding of this visual event. Constrained by our FEF neuronal sample, we cannot increase the proportion of FEF attention selective cells to match that of visually selective cells. However, we can select amongst the visually-selective cells a random sub-sample of neurons matching the number of attention-selective cells. As described in the cell drop-out analysis, decreasing the size of the neuronal population being decoded from is expected to have a drastic impact on the readout performance. This is indeed what is observed (test performed selectively with the *SVM* classifier, figure 5), though the decoding accuracy highly depends upon the visually-selective cells composing the random sub-sample: in 20 successive draws of a sub-sample of 21 visually-selective cells, performance varied from as low as 65.9% to as high as 92.9%. The readout performance at decoding the instructed position of attention from the attention-selective cells lies within this range. This suggests that the decoding accuracy of visual information and attention information are comparable and that another sample of attention-selective cells could have led to either higher or lower performances than what we describe here. Extrapolating over this observation, it should thus be possible to achieve spatial attention allocation readout performances equal to those obtained for first visual stream onset position, provided more attention-selective cells are included in the neuronal population. This will need to be confirmed experimentally.

Response reliability hypothesis. The observed differences in performance at decoding first visual stream position versus the spatial attention allocation could be due to the fact that the encoding of endogenous variables is more susceptible to trial-to-trial variability due to intrinsic factors such as motivation or fatigue. The encoding of a sensory stimulus (as first visual stream onset, here) is expected to be less affected by these intrinsic factors unless its detectability is highly degraded. Supporting this hypothesis, Cohen et al. [52], [53] show that, on a single trial, the degree to which a neuronal V4 population encodes spatial attention varies and is predictive of the overt behavioral performance on that very same trial. In Farbod Kia et al. [29], we demonstrate that, in the present task, part of the error trials arise from a miss-encoding of attention orientation. Here, the run-to-run variability in the decoding accuracy, each run consisting of a different training/testing set of trials, reflects the trial-to-trial variability with which a given variable is encoded by the neuronal population. The decoding accuracy for the spatial position of attention has a higher standard error than the decoding accuracy for first stream position. This could be due to a genuine difference in the trial-to-trial variability with which these two types of information are encoded. It is however worth noting that, though the cue-to-attention mapping required from the monkeys in the present dataset is complex, the *SVM* achieves a readout performance of 81.2% at decoding the spatial allocation of attention from the entire FEF population. This performance is relatively close to that achieved with the same classifier at decoding the same information during a simpler task involving a direct spatial mapping between cue position and attention allocation (89%, [4]) from an FEF population composed of a higher proportion of attention-selective cells (51.8%, in [4], vs. 16% in the present study). This indicates that the proportion of attention-selective cells in the neuronal population is not the only determinant of performance and response variability needs to be considered. Information redundancy across attention-selective cells should also be taken into account. In the current study, as well as in the Armstrong et al. study, single-neuron recordings were achieved in independent sessions. Decoding from simultaneously recorded neuronal population activities in a single animal is expected to uniformly improve readout performance for all decoders due to a decrease in overall (inter-subject and inter-session) data variability. However, the response of simultaneously recorded neurons also shows an important degree of correlation [54]. The impact of these correlations on the total information conveyed by such a neuronal population is controversial [55], [56], [57], [58], preventing a direct estimate of their net effect on the decoding accuracy reported here. This needs to be borne in mind when considering the present study.

Overall, our study suggests that endogenous information such as covert attention orientation can be decoded from an appropriate neuronal population with similar accuracy as exogenous information such as the position of visual stimulus. Interestingly, and in line with our present work, Gunduz et al. [59] show that the spatial position of attention can also be decoded from larger distributed neuronal populations in humans, as recorded from a parieto-frontal ECoG matrix, with a performance of up to 48% (chance = 33.3%, decoding being performed on the whole band signal spectrum). This decoding accuracy is to be compared to the performance at decoding attentional engagement (84.5%, chance = 50%) and motor engagement (92.5%, chance = 50%). Rotermund et al. [60] decode the spatial position of attention, in non-human primates, with a maximum accuracy ranging between 93% (left/right hemisphere spatial attention allocation) and 99% (spatial attention allocation to two close by positions within the same hemisphere), from a large distributed neuronal population, as recorded from an epidural ECoG matrix placed over the striate and extra-striate visual cortex. Altogether, these different studies and ours strongly support the idea that endogenous cognitive information content can be decoded from population neuronal activities.

The optimal classifiers

A general observation from our study is that the *SVM*, the *Regularized OLE*, the *Reservoir* and *the ANN OLE* unambiguously outperform *the Bayesian* and *the ANN NLE*. A link is often made between reservoir computing and kernel machines [61], [62], in particular because both techniques map the input data into a higher-dimensional feature space. In the case of the *Reservoir*, this mapping is performed explicitly by the reservoir neurons whereas the *SVM* uses the so-called "kernel-trick" to avoid this costly explicit computation. The *Regularized OLE* and *the ANN OLE* differ significantly from these two classifiers because they only use a simple hyperplane to separate the input data (i.e. they can only classify linearly separable data). Even though these four classifiers outperform the other classifiers, there are several other factors that also need to be considered.

Temporal structure in decoded feature. A major difference of reservoir computing is that it can depend on the recent history of the input. Such a *Reservoir* allows to process information that is explicitly coded in time. In contrast, the state of the other classifiers only depends on the current input [63]. As a result, using the *Reservoir* classifier is a better choice when decoding variables with a specific temporal organization as is often the case with spatial attention that moves around in time. Indeed, in such a behavioral context as the one described here, attention needs to be sustained in time from cue interpretation up to target detection. When this temporal aspect is taken into account by training the *Reservoir* on single trial population responses sampled over a longer post-cue interval (70–500 ms rather than 207–283 ms), the decoding accuracy for the spatial attention orientation is remarkably maintained over time. However, if the objective is to achieve highest decoding performance, than simpler decoding schemes appear to be more appropriate than *Reservoir* decoding.

Decoding speed-accuracy trade-off. Although the *SVM*, *Regularized OLE, Reservoir* and *ANN OLE* perform equally well in an optimal situation, it is important to note that the *regularized OLE* appears to be more resilient to a limited number of trials. Moreover, when both the number of available trials and cells in the population are limited, *the regularized OLE* outperforms *the SVM, reservoir* and *ANN OLE*. Last, when decoding speed becomes critical, the *Regularized OLE* approach is the fastest.

Information within the neuronal population. Here, we describe that the *SVM*, *the Regularized OLE, the Reservoir* and *the ANN OLE c*lassifiers outperform the other classifiers when decoding a given feature from a heterogeneous population containing both feature-selective neurons and non-selective neurons. This represents an advantage in an online decoding perspective, as it indicates that optimal readout performance can be achieved without a prior selection of the neuronal population contributing most to the feature of interest. If, for specific purposes, this selection becomes crucial, it can be performed statistically, using for example a single value decomposition approach (SVD, as in [12]).

The subject's behavior. Another critical aspect to take into consideration is the behavior of the subject which can also influence the choice of classifier. Indeed, if the subject presents a difficulty to perform the task correctly and is for example biased for one state of the feature of interest, then this produces an imbalance in the training set that can lead to a decrease in the performance. All classifiers do not behave equally in the face of this imbalance. The *Bayesian* and *the ANN OLE* decoders appear to be quite resilient to this factor, while the *SVM*, the *Regularized OLE* and the *Reservoir* are strongly affected by an imbalance beyond 10 to 40%. While imbalance in the training data sample affects the decoding performance of the *SVM* and of the *Regularized OLE*, we have shown that these two classifiers are quite resilient to a drop in trial number. As a result, they can still be considered as optimal in the case of biased behavior, provided the training is performed on a balanced subset of the data.

Number of feature states to be decoded. Support vector machines were originally designed for binary classification [41] and there is a lot of ongoing research on how to effectively extend them to multiclass decoding. Up to now several methods have been proposed where a multiclass *SVM* is constructed by using many binary *SVM* classifiers. Generally, this results in a more computationally expensive classifier [64]. The *Regularized OLE*, the *ANN OLE* and the *ANN NLE* are by essence continuous classifiers (as their output can take any value in a one-dimensional, two-dimensional or n-dimensional space) but they can also be extended to multiclass decoding by constructing several binary classifiers. The *Reservoir* can easily be implemented in a multiclass decoding problem thanks to an architecture that has the same number of output neurons as the number of classes. Each output neuron then represents one class, and the output neuron with the highest activation is chosen as best guess on a given trial. It can also be extended to a continuous n-dimensional decoder, reading out for example the position of a given variable in space, thanks to two output cells representing respectively the x- and y-coordinates. The naïve *Bayesian* classifier also naturally extends to multiclass decoding since it calculates the probability of each class given a certain response and then chooses the class with the highest probability. It can be extended to a continuous n-dimensional feature space within the Gaussian process regression framework [65].

Author Contributions

Conceived and designed the experiments: SBH. Performed the experiments: GI. Analyzed the data: EA PE GI. Contributed reagents/materials/analysis tools: PE PB PFD. Wrote the paper: EA SBH.

References

1. Ben Hamed S, Page W, Duffy C, Pouget A (2003) MSTd neuronal basis functions for the population encoding of heading direction. J Neurophysiol 90: 549–558.
2. Ben Hamed S, Schieber MH, Pouget A (2007) Decoding M1 neurons during multiple finger movements. J Neurophysiol 98: 327–333.
3. Musallam S, Corneil BD, Greger B, Scherberger H, Andersen RA (2004) Cognitive control signals for neural prosthetics. Science 305: 258–262.
4. Armstrong KM, Chang MH, Moore T (2009) Selection and maintenance of spatial information by frontal eye field neurons. J Neurosci 29: 15621–15629.
5. Gu Y, Fetsch CR, Adeyemo B, Deangelis GC, Angelaki DE (2010) Decoding of MSTd population activity accounts for variations in the precision of heading perception. Neuron 66: 596–609.
6. Meyers EM, Qi X-L, Constantinidis C (2012) Incorporation of new information into prefrontal cortical activity after learning working memory tasks. Proc Natl Acad Sci USA 109: 4651–4656.
7. Meyers EM, Freedman DJ, Kreiman G, Miller EK, Poggio T (2008) Dynamic Population Coding of Category Information in Inferior Temporal and Prefrontal Cortex. J Neurophysiol 100: 1407–1419.
8. Barak O, Tsodyks M, Romo R (2010) Neuronal Population Coding of Parametric Working Memory. J Neurosci 30: 9424–9430.
9. Crowe DA, Averbeck BB, Chafee MV (2010) Rapid sequences of population activity patterns dynamically encode task-critical spatial information in parietal cortex. J Neurosci 30: 11640–11653.
10. Kadohisa M, Petrov P, Stokes M, Sigala N, Buckley M, et al. (2013) Dynamic Construction of a Coherent Attentional State in a Prefrontal Cell Population. Neuron, 80(1):235–46.
11. Stokes MG, Kusunoki M, Sigala N, Nili H, Gaffan D, et al. (2013) Dynamic coding for cognitive control in prefrontal cortex. Neuron 78: 364–375.
12. Markowitz DA, Wong YT, Gray CM, Pesaran B (2011) Optimizing the decoding of movement goals from local field potentials in macaque cortex. J Neurosci 31: 18412–18422.
13. Pesaran B, Pezaris JS, Sahani M, Mitra PP, Andersen RA (2002) Temporal structure in neuronal activity during working memory in macaque parietal cortex. Nat Neurosci 5: 805–811.
14. Thomson EE, Carra R, Nicolelis MAL (2013) Perceiving invisible light through a somatosensory cortical prosthesis. Nat Commun 4: 1482.
15. Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, et al. (2012) Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature 485: 372–375.
16. Bishop C (2007) Pattern Recognition and Machine Learning. New York, NY, USA: Springer.
17. Duda R, Hart P, Stork D (2000) Pattern Classification. 2nd ed. New York: John Wiley and Sons.
18. Seung HS, Sompolinsky H (1993) Simple models for reading neuronal population codes. Proc Natl Acad Sci USA 90: 10749–10753.
19. Salinas E, Abbott LF (1994) Vector reconstruction from firing rates. J Comput Neurosci 1: 89–107.
20. Zhang K, Ginzburg I, McNaughton BL, Sejnowski TJ (1998) Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. J Neurophysiol 79: 1017–1044.
21. Ma WJ, Beck JM, Latham PE, Pouget A (2006) Bayesian inference with probabilistic population codes. Nat Neurosci 9: 1432–1438.

22. Jaeger H (2001) The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, German National Research Center for Information Technology.
23. Bruce CJ, Goldberg ME (1985) Primate frontal eye fields. I. Single neurons discharging before saccades. J Neurophysiol 53: 603–635.
24. Bullier J, Schall JD, Morel A (1996) Functional streams in occipito-frontal connections in the monkey. Behav Brain Res 76: 89–97.
25. Ibos G, Duhamel J-R, Ben Hamed S (2013) A functional hierarchy within the parietofrontal network in stimulus selection and attention control. J Neurosci 33: 8359–8369.
26. Armstrong KM, Fitzgerald JK, Moore T (2006) Changes in visual receptive fields with microstimulation of frontal cortex. Neuron 50: 791–798.
27. Barone P, Batardiere A, Knoblauch K, Kennedy H (2000) Laminar distribution of neurons in extrastriate areas projecting to visual areas V1 and V4 correlates with the hierarchical rank and indicates the operation of a distance rule. J Neurosci 20: 3263–3281.
28. Pouget P, Stepniewska I, Crowder EA, Leslie MW, Emeric EE, et al. (2009) Visual and motor connectivity and the distribution of calcium-binding proteins in macaque frontal eye field: implications for saccade target selection. Front Neuroanat 3: 2.
29. Farbod Kia S, Åstrand E, Ibos G, Ben Hamed S (2011) Readout of the intrinsic and extrinsic properties of a stimulus from un-experienced neuronal activities: towards cognitive neuroprostheses. J Physiol Paris 105: 115–122.
30. Ibos G, Duhamel J-R, Ben Hamed S (2009) The spatial and temporal deployment of voluntary attention across the visual field. PLoS ONE 4: e6716.
31. Pouget A, Sejnowski TJ (1994) A neural model of the cortical representation of egocentric distance. Cereb Cortex 4: 314–329.
32. Pouget A (1997) Spatial transformations in the parietal cortex using basis functions. Journal of Cognitive Neuroscience 9: 222.
33. Tikhonov AN (1963) Solution of incorrectly formulated problems and the regularization method. Doklady Akademi Nauk SSSR: 501–504.
34. Refregier P, Vignolle J-M (1989) An Improved Version of the Pseudo-Inverse Solution for Classification and Neural Networks. EPL 10: 387.
35. Mitchell T (1997) Machine Learning. McGraw Hill.
36. Mitchell T, Hutchinson R, Niculescu R, Pereira F, Wang X, et al. (2004) Learning to Decode Cognitive States from Brain Images. Machine Learning: 145–175.
37. Duda RO, Hart PE (1973) Pattern classification and scene analysis. New York: Wiley.
38. Jaeger H (2007) Echo state network. Scholarpedia 2: 2330.
39. Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Computer Science Review 3: 127–149.
40. Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D (2007) An experimental unification of reservoir computing methods. Neural Netw 20: 391–403.
41. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20: 273–297.
42. Aizerman M, Braverman E, Rozonoer LI (1964) Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control: 821–837.
43. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory. COLT '92. New York, NY, USA: ACM.pp. 144–152.
44. Chang C-C, Lin C-J (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2: 27:1–27:27.
45. Standage D, Paré M (2011) Persistent storage capability impairs decision making in a biophysical network model. Neural Netw 24: 1062–1073.
46. Stanton GB, Bruce CJ, Goldberg ME (1995) Topography of projections to posterior cortical areas from the macaque frontal eye fields. J Comp Neurol 353: 291–305.
47. Schall JD, Morel A, King DJ, Bullier J (1995) Topography of visual cortex connections with frontal eye field in macaque: convergence and segregation of processing streams. J Neurosci 15: 4464–4487.
48. Mohler CW, Goldberg ME, Wurtz RH (1973) Visual receptive fields of frontal eye field neurons. Brain Res 61: 385–389.
49. Moore T, Fallah M (2004) Microstimulation of the frontal eye field and its effects on covert spatial attention. J Neurophysiol 91: 152–162.
50. Thompson KG, Biscoe KL, Sato TR (2005) Neuronal basis of covert spatial attention in the frontal eye field. J Neurosci 25: 9479–9487.
51. Gregoriou GG, Gotts SJ, Desimone R (2012) Cell-type-specific synchronization of neural activity in FEF with V4 during attention. Neuron 73: 581–594.
52. Cohen MR, Maunsell JHR (2010) A neuronal population measure of attention predicts behavioral performance on individual trials. J Neurosci 30: 15241–15253.
53. Cohen MR, Maunsell JHR (2011) Using neuronal populations to study the mechanisms underlying spatial and feature attention. Neuron 70: 1192–1204.
54. Zohary E, Shadlen MN, Newsome WT (1994) Correlated neuronal discharge rate and its implications for psychophysical performance. Nature 370: 140–143.
55. Abbott LF, Dayan P (1999) The effect of correlated variability on the accuracy of a population code. Neural Comput 11: 91–101.
56. Romo R, Hernández A, Zainos A, Salinas E (2003) Correlated neuronal discharges that increase coding efficiency during perceptual discrimination. Neuron 38: 649–657.
57. Hung CP, Kreiman G, Poggio T, DiCarlo JJ (2005) Fast readout of object identity from macaque inferior temporal cortex. Science 310: 863–866.
58. Ecker AS, Berens P, Tolias AS, Bethge M (2011) The effect of noise correlations in populations of diversely tuned neurons. J Neurosci 31: 14272–14283.
59. Gunduz A, Brunner P, Daitch A, Leuthardt EC, Ritaccio AL, et al. (2012) Decoding covert spatial attention using electrocorticographic (ECoG) signals in humans. Neuroimage 60: 2285–2293.
60. Rotermund D, Ernst U, Mandon S, Taylor K, Smiyukha Y, et al. (2013) Toward High Performance, Weakly Invasive Brain Computer Interfaces Using Selective Visual Attention. The Journal of Neuroscience 33: 6001–6011.
61. Schmidhuber J, Wierstra D, Gagliolo M, Gomez F (2007) Training recurrent networks by Evolino. Neural Comput 19: 757–779.
62. Shi Z, Han M (2007) Support vector echo-state machine for chaotic time-series prediction. IEEE Trans Neural Netw 18: 359–372.
63. Hermans M, Schrauwen B (2012) Recurrent kernel machines: computing with infinite echo state networks. Neural Comput 24: 104–133.
64. Hsu C, Lin C (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13: 415–425.
65. Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press.

Chapter 7

Dynamical Mixed Selectivity in Reservoir Computing and Primate Prefrontal Cortex

We want to show that randomly connected recurrent networks can elicit rich dynamics that underlie all the representations necessary to perform a cognitive task. To demonstrate this, we used a simple recurrent network with untrained recurrent connections, the reservoir. Because the representational power of such a network is limited in the temporal domain, we implemented a feedback mechanism that represented context explicitly. After training a reservoir to perform a cognitive task, we analyzed and compared its unit and population activity to single neuron recordings from the dorsal anterior cingulate cortex of macaque monkeys which were trained to perform the same task.

The present experiment focuses on the representational power of randomly generated recurrent networks, therefore, the learning mechanism is secondary in this particular experiment. As a consequence, the learning method was chosen for its efficiency and has low biological plausibility.

Title:

Dynamical mixed selectivity in reservoir computing and primate prefrontal cortex

Abbreviated title:

Dynamical aspects of mixed selectivity

Authors:

Pierre Enel[1,2], Emmanuel Procyk[1,2], René Quilodran[1,2,3], Peter Ford Dominey[1,2]

1 - Stem Cell and Brain Research Institute, INSERM U846, 18 ave Doyen Jean Lepine, 69675 Bron Cedex, France

2 - Université de Lyon, Université Lyon I, Lyon, France

3 - Escuela de Medicina, Departamento de Pre-clínicas, Universidad de Valparaíso, Hontaneda 2653, Valparaíso, Chile

Acknowledgement: The present work was funded by European research projects FP7 ORGANIC, EFAA and WYSIWYD. EP is funded by the Agence Nationale de la Recherche ANR-06-JCJC-0048 and ANR- 11-BSV4-0006, and by the labex CORTEX ANR-11-LABX-0042. The authors declare no competing financial interests.

Abstract

A characteristic aspect of primate cognitive function is the ability to adapt to novel situations. Determining what is most pertinent in these situations is not always possible based only on the current sensory inputs and often depends on the interaction between current inputs, recent inputs, internal states, and behavioral outputs. In this context, a legitimate question is how cortical dynamics represent these complex contingencies. A recent area of research referred to as reservoir computing has demonstrated that recurrent neural networks have the property of recombining present and past inputs into a higher dimensional space thereby providing a pre-coding of an essentially universal set of combinations which can then be selected and used arbitrarily for their relevance to the task at hand. Such non-linear recombinations have been referred to as mixed selectivity in recent studies which demonstrated their importance for adaptive behavior. In the current research, we train a reservoir to perform a complex cognitive task, and compare its neural activity to dorsal anterior cingulate cortex (dACC) neurons previously recorded in two rhesus monkeys performing the same task. The systematic comparison of single unit and population activities revealed that the reservoir inherently displays a dynamic form of mixed selectivity and can also learn attractor dynamics that are likewise observed in the dACC neurons, thus emphasizing the reservoir-like characteristics of dACC. The direct applicability of the reservoir to an arbitrary cognitive task argues in favor of the corresponding universal coding capabilities in such recurrent networks including the PFC.

Introduction

Prefrontal cortex (PFC) plays a major role in behavioral flexibility (Miller and Cohen, 2001). A pervasive feature of PFC is neural activity that is not selective for a particular behavioral variable but that instead reflects a complex combination of these variables. Though observed in early PFC studies while animals performed tasks involving multiple variables (Watanabe, 1992; Sakagami and Niki, 1994; Asaad et al., 1998), these "mixed selectivity" activities have only recently become a specific research focus (Rigotti et al., 2010a, 2010b, 2013). Mixed selectivity and the resulting capacity to represent arbitrary combinations of inputs may be crucial in adapting to new contingencies (Rigotti et al., 2013). Rigotti et al. (2010b) demonstrated that randomly connected networks produce mixed selectivity of current inputs. Such networks have been explored in reservoir computing.

Reservoir computing refers to neural network models in computational neuroscience (Maass et al., 2002) and machine learning (Jaeger, 2001), characterized by a sparsely connected recurrent network of neurons (spiking or analog), with fixed connection weights (excitatory and inhibitory). Via the recurrent connections, inputs are projected into a high dimensional dynamic space inherently sensitive to combinatorial structure of inputs but also to their serial and temporal structure, which constitute temporal information (Jaeger and Haas, 2004; Buonomano and Maass, 2009). These representations are read out from reservoir neurons via modifiable connections. In the first of such models the reservoir corresponded to PFC, and the modifiable readout connections to the corticostriatal projections (Dominey, 1995; Dominey et al., 1995). Maass developed a related approach with spiking neurons and demonstrated their non-linear computational capabilities (Maass et al., 2002). In machine learning, Jaeger demonstrated how such systems have inherent signal processing capabilities (Jaeger and Haas, 2004). When these networks are exposed to inputs with multiple dimensions (e.g. target identification, serial order, match/nonmatch) neurons represent non-linear mixtures of these dimensions (Dominey et al., 1995; Rigotti et al., 2010b). Interestingly, spatiotemporal reservoir properties are found in cortex (Kilgard and Merzenich, 1999; Brosch and Schreiner, 2000; Bartlett and Wang, 2005; Nikolić et al., 2009), as stimuli presented in the past influence the processing of subsequent stimuli.

Here we tested whether key properties of reservoir networks could be matched with observed neural coding in the frontal cortex. We directly compared mixed selectivity properties, population dynamics and attractor characteristics observed in reservoir neurons with primate anterior cingulate cortex neurons from monkeys performing a problem solving task that involves search and repetition behaviors (Quilodran et al., 2008). The systematic comparison revealed that the reservoir displays a dynamic form of mixed selectivity and can learn population attractor dynamics that are likewise revealed in the cortical neurons, thus illustrating the reservoir characteristics of PFC. Strengthening contextual information by feeding it back to the reservoir turned out to develop attractors that corresponded to the different contexts. Dynamic mixed selectivity inherent to reservoir networks can allow for a form of universal coding of arbitrary pertinent combinations of inputs and internal states, and this intrinsic capability can be enhanced via adaptation through learning.

Materials and Methods

Problem solving task

In order to compare the neural activity in the recurrent network model with that of the behaving primate cortex, we tested both systems using a problem solving task that was originally developed by Procyk and Goldman-Rakic (2006) to investigate shifts between exploration and exploitation behavior (see Quilodran et al., 2008 for detailed description). Two rhesus monkeys had to find by trial and error which among four targets presented on a touch screen was rewarded by fruit juice (Fig. 1A). At the onset of a trial, monkeys fixated a central fixation point and held their hand on a lever displayed on the screen below the fixation point (Fig. 1B). After a delay period of 1.5 seconds, 4 targets appeared on the screen. The animals made a saccade to one target and fixated it for 0.5 seconds until the lever disappeared, giving the GO signal to touch the chosen (fixated) target. Feedback was preceded by a 0.6-second delay, followed by a 2-second delay ending at the beginning of next trial. The search phase included incorrect trials (INC) and the first rewarded trial (COR1) during which animals explored the targets. The following 3 correct trials (COR) allowed the monkeys to repeat the rewarded choice and constituted the repetition phase. Occasionally (10% of cases) repetition lasted for 7 or 11 trials to prevent the animals from anticipating the end of the repetition phase. A signal to change appeared at the end of the last repetition trial to indicate to the animal that a new target was going to be rewarded. A search phase and its following repeat phase are referred to as a problem. In only 10% of cases the same target was rewarded in two consecutive problems. After training, monkeys performed the task in a nearly optimal fashion. In each search, they avoided previously explored targets that were not rewarded, and correctly repeated the rewarded choice. Likewise, they generally avoided repeating the previously rewarded target in the subsequent problem. The average number of trials in search was 2.4 ± 0.15 for first monkey and 2.65 ± 0.23 for second monkey (knowing that the same target is not rewarded two problems in a row, expected number of trials in optimal search is ~2.2) and in repetition 3.14 ± 0.7 and 3.4 ± 0.55 for first and second monkey respectively (the optimal-repetition trial number is above 3, as some problems had more than 3 rewarded repetition trials).

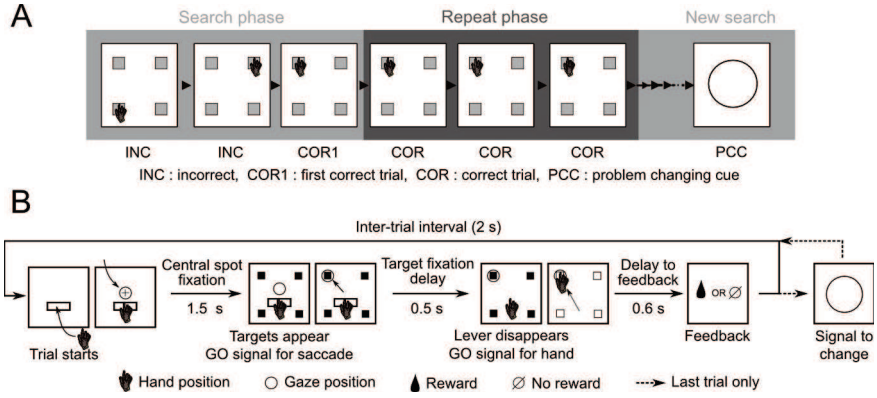

Figure 1. Problem solving task. **A**. *An example problem. Monkeys searched through the targets by trial and error to find which target was rewarded. Incorrect trials (INC) and first rewarded trial (COR1) constitute the search phase. In this example, the upper left target is rewarded. The set of subsequent rewarded trials is defined as the repeat phase, made up of correct (COR) trials. At the end of the repeat phase a problem changing cue indicated to the monkey that a new problem was starting, thus a new target was rewarded.* **B**. *Single trial events. Trials started with hand on lever and fixation of a central point. A 1.5s delay ensued before targets appeared and fixation point disappeared, providing the GO signal. The monkey made a saccade to the chosen target and fixated it for 0.5s. The GO signal for hand movement was given by lever removal and the monkey touched*

the fixated target. All targets go blank at the time of touch and disappeared at feedback. Feedback was preceded by a 0.6s delay and followed by a 2s inter-trial interval.

Recurrent neural network model

We developed a recurrent neural network model using reservoir computing (RC) to perform the cognitive task in order to generate predictions that could then be tested with dACC monkey data. According to the RC principle, a fixed, large, random reservoir (recurrent neural network RNN) is excited by input signals, and the desired output is combined from the excited reservoir signals by a trainable readout mechanism (a simple linear regression in the most simple versions). The RC principle has been independently discovered in cognitive neuroscience (temporal recurrent networks, Dominey et al., 1995; Dominey and Ramus, 2000), in computational neuroscience (liquid state machine, Maass et al., 2002), and in machine learning (echo state networks, Jaeger, 2001). Models have been recently developed along the RC principle to reproduce cognitive functions like working memory (Pascanu and Jaeger, 2011) and language comprehension (Tong et al., 2007; Hinaut and Dominey, 2013). Two versions of the model were used in order to obtain the results of this paper: the original version and a second version implementing a simple contextual memory. The initial version was used in single neuron analyses in the first part of the results while the contextual memory version was introduced later to show the benefits of context encoding (Pascanu and Jaeger, 2011; Mante et al., 2013).

In both versions, a recurrent network of firing rate neurons received task inputs and were fully connected to a readout layer, the output of the model (Fig. 2A). Reservoir recurrent connections provide rich dynamics formed by nonlinear recombinations of inputs that evolve through time. Readout neurons activate to represent model's target choice, and feed back the choice through readout-reservoir connections. Reservoir-readout connections are the only modifiable connections of the model.

Single neuron dynamics

Neurons were simulated as leaky-integrator firing-rate units. Inputs were integrated over time with the following equation:

$$x(t+\Delta t)=\left(1-\frac{\Delta t}{\tau}\right)x(t)+\frac{\Delta t}{\tau}\left(W_{\text{res}}\, r(t)+W_{in}\, u(t)+W_{fb}\, z(t)\right) \quad (1)$$

where $x(t)$ denotes the membrane potential vector of reservoir neurons, Δt the time step (25 ms), τ the time constant of the leaky integration (375 ms or 15 time steps), W_{res} the reservoir internal-weight matrix, $r(t)$ the firing rate vector of the reservoir neurons, W_{in} is the input weight matrix, $u(t)$ the input neuron vector, W_{fb} the readout to reservoir weight matrix and $z(t)$ the readout neuron vector. At each time step the firing rate $r(t)$ of reservoir neurons was computed as the hyperbolic tangent of its membrane potential $x(t)$ generating a nonlinearity in the dynamics of the neuron:

$$r(t)=\tanh\left(x(t)\right)$$

The readout unit activity was defined as the weighted sum of the reservoir-neuron firing rate:

$$z(t)=W_{out}\, r(t)$$

where W_{out} is the readout-weight matrix.

Neural network architecture

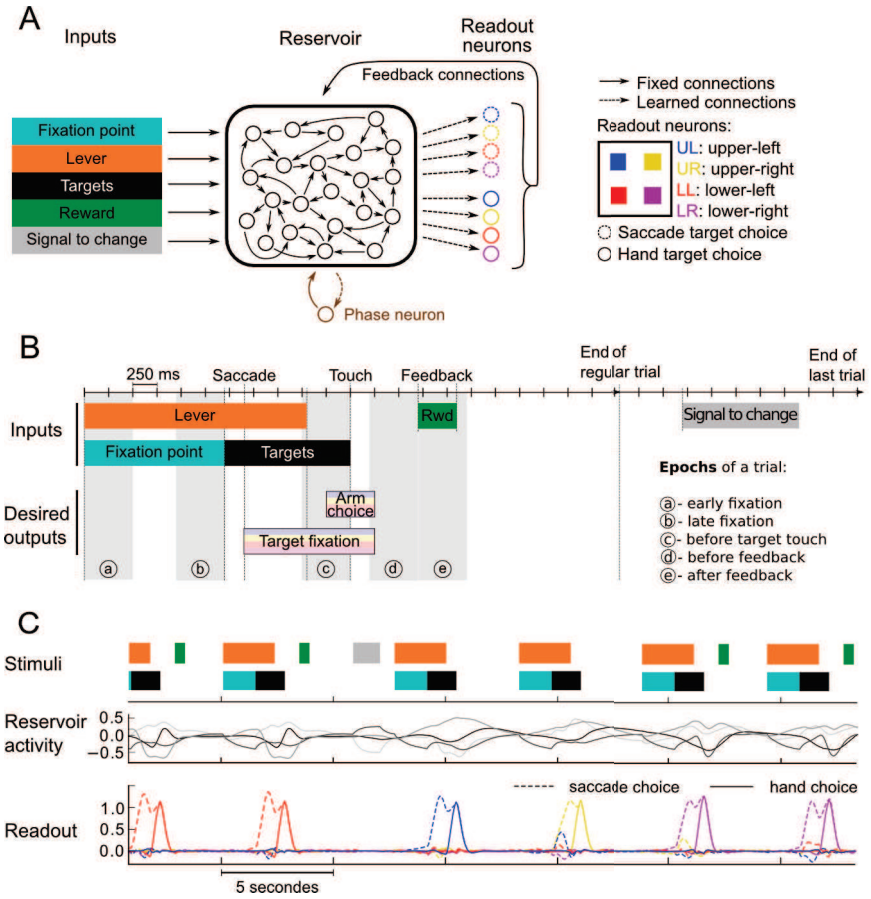

Figure 2. Model architecture, modeled task and test-activity example. **A**. *Model architecture. A recurrent network of 1000 randomly connected neurons called the reservoir received input from 5 units representing the presence of the fixation point, the lever, the targets, the reward and the signal to change. Output choice of the network was represented in two sets of 4 readout neurons corresponding to target fixation and arm touch respectively. Connections between the reservoir and readout (dashed arrows) were modified through learning to reproduce the behavior given by sequence of correct input/output examples. A contextual memory version of the model included a context neuron (in brown) that represented phase information (search or repeat).* **B**. *Time course of a modeled trial. A trial started with the activation of the lever and fixation point. Fixation point neuron deactivated concomitantly with targets appearance which was the GO signal for saccade to a target after a reaction time. Fixation of the target followed and was represented in the activation of one readout neuron among the four dedicated to target fixation. The lever input deactivation was the GO signal for arm touch that was represented in the activation of the readout neuron, after a reaction time, that represented the target chosen with the saccade in the second set of four readout neurons. Touch event occurred at the middle of arm choice and was the start of a 0.6 second delay to feedback. Feedback was simulated as the activation of the reward input for correct trials and the absence of activation in incorrect trials. A 2.15 seconds inter-trial interval started at the onset of feedback and ended at the onset of the next trial except for the last trial of problem (COR4: fourth correct trial) in which the inter-trial interval was extended to 4.25 seconds and the signal to change activated for 1.2 seconds. (Rwd: reward)* **C**. *Example of the network performing the task after learning to explore the targets with a circular search. Upper panel: A sequence of stimuli neurons activation. Middle panel: Activity of 4 example reservoir neurons. Lower panel: Readout of the network showing the successive choices of the model. The example shows the end of a repetition period where red target was rewarded. After signal to change input was activated (grey block), a new search for the rewarded target began with the exploration of blue target, then yellow and finally purple which was rewarded and then repeated.*

We implemented an RC model where a reservoir of 1000 recurrently connected neurons was fully connected to a readout layer. Learning took place only between the reservoir neurons and the readout units, at the level of the readout weights. Weights between reservoir neurons (internal weights) and between input and reservoir neurons (input weights) were stochastically generated and fixed. Input weights were generated with a uniform distribution in the interval [-1, 1] with a 0.1 probability of connection. Internal weights followed a Gaussian distribution (mu=0, sigma=1) with a 0.1 probability of connection between each pair of neurons. These were scaled so that the largest absolute eigen-value of the weight matrix – commonly referred to as the spectral radius – was equal to 0.9. This ensured a dynamical regime allowing for sustained activity in the recurrent network without saturation. Activity in the network thus developed and integrated successive stimuli inputs so that activity at each time point represented the combination of previous and current inputs (reservoir computing principle).

Input neurons represented the major external features of the task (Fig. 2A). They included 5 inputs, each represented by one neuron: the fixation point, the lever, the targets, the reward and the signal to change. Each of these neurons had a 0.1 chance of connecting with each reservoir neuron. Weights were generated following a uniform distribution in the interval [-1, 1]. The model generated outputs corresponding to oculomotor saccades and arm touches to the spatial targets corresponding to the monkeys' behavioral output and time course of the task events. A first set of 4 readout neurons represented the four possible target choices for eye saccades and a second set of 4 readout neurons represented arm touches. The highest activated neuron for each of the two sets represented the model's choice and both neurons were required to represent the same target in each trial. In the contextual memory version of the model, an additional readout neuron was trained to represent the phase (search vs repetition). In both versions, all reservoir neurons were connected to the readout neurons and constituted the only modifiable connections of the network. The readout neurons were connected back to the reservoir neurons to feed the choice information back to the recurrent neurons with a 0.1 chance of connection. These connections were generated prior to learning and remained fixed for the duration of the experiment. Connection weights were drawn from a uniform distribution between -1 and 1 for the choice outputs and between -0.3 and 0.3 for the contextual readout neuron in the contextual memory version of the model.

Simulation protocol

We trained the model to learn a task that reproduces the major features of the actual task performed by monkeys (Fig. 2B). Timing of these elements closely matched the actual monkey task in order to compare evolution of activity in dACC and reservoir neurons. Fixation point and lever were each simulated as the activation of their corresponding input neurons. They provided GO signals as they switched off for saccade to and fixation of a target, and for touching this target respectively. The readout neurons corresponding to these choices were trained to activate at their respective GO signal after a reaction time of 250 ms and deactivated after a 250 ms reaction time following touch. Following the fixation point, an input neuron represented the presence of the targets on screen and is deactivated after touch. Activation of the arm touch neuron started before the actual touch event to allow the neuron representing arm choice to reach full activation before switching off the targets input and to simulate the preparation and movement itself. Feedback was simulated with a reward input neuron that activated when a choice was correct. At the end of a problem, a fifth input neuron was activated to represent the signal to change indicating the start of a new problem. In the contextual memory version, the context neuron representing the phase was trained to activate when the signal to change input neuron was being activated, and to remain active for the duration of the search phase, until the first reward.

Each trial lasted 5550 ms (222 time steps), except for the last correct trial (COR4) that ended with the presentation of the signal to change and lasted 8050 ms (322 time steps). The task

was taught to the reservoir with supervised learning using a matched set of <stimuli, desired output> pairs made of 600 problems. Readout neurons were trained to represent choice by activating to value 1 at periods of choice while remaining silent the rest of the time, thus acting like binary neurons. The training procedure employed a slightly modified version of the FORCE learning method developed by Sussillo and Abbott (2009). With the FORCE method, learning of connection weights between reservoir and readout neurons is based on an on-line process of weight adjustment that allows for sampling of the readout error by the system. Weights are corrected so that a small fraction of the readout error is fed back to the reservoir. Readout weights are successively modified to produce the target output while sampling deviations in the reservoir activity that result from readout feedback with a slight discrepancy between actual and desired output. Hence, the system learns to produce a stable readout even in the face of readout errors that are propagated to the reservoir. We used the recursive least-squares algorithm in combination with the FORCE learning principle to modify readout weights, as described in Sussillo and Abbott (2009):

$$W_{out}(t)=W_{out}(t-\Delta t)-\frac{e(t)P(t)r(t)}{1+r^{T}P(t)r(t)}$$

Where $e(t)$ is the error before weights are modified and is defined as the difference between actual and desired output. The error of readout neuron i is defined as follows:

$$e_i(t)=W^{T}_{out\,i}(t-\Delta t)r(t)-d_i(t)$$

where $W_{out\,i}$ is the weight vector between the reservoir neurons and the readout neurons and $d_i(t)$ is the desired output. $P(t)$ can be assimilated to the matrix of all learning rates for each pair of reservoir and readout neurons and is modified as follows:

$$P(t)=P(t-\Delta t)-\frac{P(t-\Delta t)r(t)r^{T}(t)P(t-\Delta t)}{1+r^{T}(t)P(t-\Delta t)r(t)} \quad \text{with} \quad P(0)=I$$

where I is the identity matrix.

To allow for better convergence of the weights, we modified the feedback from the readout to the reservoir generated with the original FORCE learning method. We blended the actual output, produced after weight modification according to the FORCE principle, with a clamped feedback i.e. a delayed version of the desired output. The proportion of clamped feedback and actual output varied smoothly and steadily during training, starting with only clamped feedback and ending with actual output. The signal $f(t)$ was fed back to the reservoir and replaced $z(t)$ in equation (1):

$$f(t)=\frac{t}{L}z(t)+\frac{L-t}{L}c(t)$$

where L is the full duration of training and $c(t)$ the clamped feedback that is a 325 ms (13 time steps, determined through optimization) delayed version of the desired output. This delay greatly improved learning in our experiment. With a delayed desired output as clamped feedback, readout neurons had to learn to activate at the onset of fixation and arm choice without the correct and expected readout activity that would have been fed back to the reservoir with FORCE-learning fast adaptation of the weights. Likewise, when readout neurons should deactivate at the end of fixation and arm choice, the reservoir neurons still received the clamped activity resembling a readout that was not deactivated. Similar to the FORCE learning principle, this method allowed the learning algorithm to sample a higher number of time steps with discrepancies between actual and desired readout around the activation and deactivation of the readout neurons.

Testing procedure

In order to assess the trained model's behavioral performance, a sequence of 200 problems was provided as input to the reservoir and the output choices were evaluated. The maximally activated neurons for saccade and hand choices had to match, and thus represented the model's choice. Trials where saccade and hand choices did not match were counted as errors.

Performance was assessed on the basis of three rules: (1) do not repeat an unrewarded target choice; (2) repeat rewarded target choice once found; (3) while searching for the rewarded target, do not choose the target rewarded on the previous problem. Performance of the model was measured according to these rules. Trials that did not respect one of the three rules counted as an error. Error rate was defined as the number of trials that did not respect the rules over the total number of trials. In order to balance the length of the search period, the number of search trials was generated for each problem in advance. Thus, no target was predefined as rewarded, rather, after a predefined number of search trials (from 1 to 3), the reward was given and the behavioral output of the model was assessed according to the above described rules (for a similar method used to test human subjects, see Amiez, Sallet, Procyk, & Petrides, 2012).

The model was tested after training with three different behavioral strategies that complied with the rules. First, using a random search where the targets were explored in a different order at each problem. Second, using an ordered search where targets were explored following the same target sequence at each problem while avoiding the previously rewarded target in the sequence. In other words, the search always started with the same target, except if it was rewarded on the previous problem, and followed the same sequence, again, avoiding the target rewarded on the previous problem. Third, using a circular search where targets were explored in infinite repeating circle. As an example, let's define the repeating sequence upper-left (UL), upper-right (UR), lower-right (LR), lower-left (LL), UL, UR, LR and so on. If for a given problem, the rewarded target is UR, the search of the next problem will start with the next target in the sequence, namely, LR and continue with LL and UL until it finds the rewarded target.

Reservoir neuron analyses reported here are based on the activity of networks that learned to explore targets with the circular search. Results did not differ when the model was trained with the ordered search. Figure 2C illustrates the activity of reservoir and readout neurons corresponding to a sequence of inputs once the task has been taught with a circular search.

Neural data

Quilodran et al. recorded 546 neurons in the dorsal bank of the cingulate sulcus of two rhesus monkeys and analyzed them along with local field potential for their correlation with the behavioral shift (Quilodran et al., 2008). The present article reports on a new and separate reanalysis of this dataset to support findings obtained with modeling. All reanalyzes of these data were based on firing rate estimates of the recorded neurons. Subsets of this pool of neurons were selected depending on the requirements of the analyses. The number of neurons per analysis is specified in each case in the related method description.

Model and monkey single unit analysis

Mixed-selectivity analysis was performed by using the same methods for both reservoir neurons from the model (neurons from the recurrent network) and dACC neurons. The analysis focused on specific 500 ms trial epochs. Epochs used were: early fixation (0 – 500 ms from fixation onset), late fixation (-500 – 0 ms to targets appearance), before touch (-500 – 0 ms to target touch), before feedback (-500 – 0 ms to feedback) and after feedback (0 – 500 ms from feedback) (Fig. 2B). Firing rates of reservoir neurons were averaged within these periods, thus obtaining a single firing rate value for each epoch. Average activity of dACC neurons for each epoch of each trial was estimated as the number of spikes within these epochs. Epoch, along with phase (search vs. repetition) and choice (UL, UR, LR, LL) constitute the three factors used in single neuron analysis with 5, 2 and 4 possible levels respectively (40 conditions total). The dACC neuron pool for the mixed selectivity analysis was a subset of 85/546 dACC neurons selected for having at least 15 trials per condition. All reservoir neurons were included in the analysis. A three-way ANOVA was

conducted on the activity of each neuron with factors Epoch x Phase x Choice. A neuron was considered significant for a factor or an interaction between factors if its p-value was inferior to 0.05. Interaction effects between phase and choice are considered here as an indicator of mixed selectivity which is defined by the interaction of these variables in their contribution to the firing rate of a single neuron. Thus in this present study we use the term "mixed selectivity" to refer exclusively to its non-linear component. Moreover, we introduce the terminology "dynamic mixed selectivity" to refer to mixed selectivity patterns that interact with epoch and correspond in our experiment to the interaction between epoch, phase and choice variables in the ANOVA analyses.

Reservoir neurons of the model were also analyzed for their response to the first reward in a problem to compare with results obtained in Quilodran et al. (2008). For that purpose, firing rate activities of single reservoir neurons were averaged over the time window 300 ms to 800 ms after feedback onset and then pooled in incorrect (INC), first correct (COR1) and correct (COR) trials. Pairwise t-test with false discovery rate correction (p-value < 0.05) was used to quantify the number of reservoir neurons that fired significantly more in COR1 trials than in INC and COR trials.

Model and monkey population analysis

Population analyses were performed on neural activity from full trials. Firing rate of each dACC neuron was first estimated with a Gaussian kernel (standard deviation = 100 ms) convolved through time every millisecond, eliciting a firing rate estimate at each millisecond. Activities were time normalized to accommodate for trial-time variations with the following method. The average duration of periods between key events of the task was calculated and allowed us to determine the number of time bins of a specified size (see below) within each period. The activity of each neuron was then divided in the number of time bins. For each neuron, estimated firing rate within these time bins was averaged to elicit a single average firing rate value per time bin. The events used to normalize time were: lever touch, targets appearance, target touch, feedback, next-trial lever touch.

A state space analysis was performed, allowing visualization of population activity trajectories, with a principal component analysis (PCA). Activities of each successive trial in a problem were averaged at the level of single neurons over each trial type: INC1 and INC2 were the first and second unrewarded search trials, COR1 the first rewarded trial, and COR2-4 the repeat trials with the presentation of the signal to change at the end of the COR4 trial. INC3 trials were ignored due to lack of trials in dACC data, but results with few trials show that INC3 trajectory was very close to INC2 trajectory and did not add relevant information to this analysis. A subset of 184 dACC neurons were selected for having at least 10 trials of each type. The activity of each dACC neuron was time normalized using the above described method with 100 ms time bins and then mean normalized over all bins for each trial type. Similarly, reservoir neuron activity was averaged over all trials of a trial type for each time point. PCA was performed on the data matrix where columns correspond to individual neurons and rows represent the concatenated time points of each trial type. Each cell in the matrix was the mean normalized average firing rate of one neuron at one time bin for one trial type. All reservoir neurons were included in this analysis.

We performed cross-temporal pattern analysis (CTPA) on population activity of reservoir and dACC neurons to assess the dynamic nature of the representation of phase (see Stokes et al. (2013) for the original method). Search activity included INC but not COR1 trials that lie at the transition between search and repeat phases. Repeat activity encompassed all COR trials except COR1. Activity of COR4 trials until the regular end of a trial were also included (Fig. 1 and 2B for trial time course). All reservoir neurons were included in the analysis. For dACC data, a subset of 389/546 neurons was selected for having at least 20 trials in each of the two phases. The activity of each dACC neuron was time normalized using the above described method with 20 ms time bins.

For each dACC neuron in the population subset, every other trial was selected to be part of the training data. The remaining trials constitute the test data, for both search and repeat activities, thereby obtaining $search_{train}$, $search_{test}$, $repeat_{train}$, $repeat_{test}$ activities. Interleaved selection of trials prevents a possible bias due to the drift of single unit activity within a recording session of dACC neurons. For the model, the first half of the search and repeat trials of the modeled activity were selected to compose the train subsets and the second half as the test subsets. Search and repeat activities were then contrasted using the averaged activity over all the trials for each neuron: $trainData = avg(search_{train}) - avg(repeat_{train})$, $testData = avg(search_{test}) – avg(repeat_{test})$. Thus, trainData and testData were vectors where each element was the contrasted averaged activity of each neuron for train and test data respectively. We used a Pearson correlation coefficient r between the train and test vectors to quantify the similarity of phase representation. This procedure was repeated using different time points for train and test and successively assessed the similarity between phase representations at different time points within a trial. An assumption free permutation test was performed on dACC analysis results to derive the statistical significance of the correlation coefficient r that elicited lower values than the model. P-value for each training/testing time-point pair was obtained with 10000 randomly-generated permutations. Significance threshold was set to p-value = 0.001.

A decoding method was used to assess the capability of a linear readout to extract a continuous phase signal from the dynamic activity of the reservoir and dACC neurons. In the absence of a context neuron, a readout unit without feedback to the reservoir was trained to activate similarly to the phase context neuron, i.e. to start firing when the signal to change was given to the network, firing continuously during search phase and deactivate when the first reward was given during COR1 trials. With this method, phase information extraction has to rely only on the activity of the reservoir neurons. Similarly, phase state was decoded from dACC population activity with time normalization (20ms bins) over all time bins of a trial after training a ridge regression on full trial activities. The search trials included all INC trials, and the repeat phase was composed of COR2-3 trials, COR1 and COR4 trials lying at the transition between search and repeat. A subset of 290/546 neurons were selected for having at least 20 trials in each category (search/repeat). The 20 trials of each category were divided in a test set (2 trials) and a cross-validation set (leave one out) that was used to derive the ridge parameter of the linear regression (Tikhonov regularization to avoid overfitting). The operation was repeated with 10 different test sets in order to assess the performance of the continuous decoder over a wide range of train and test trials. For each fold of validation in each fold of test, a linear readout model was derived from the linear regression between the full trial length (all time bins considered as an observation) and the desired output which was 1 for search trials and 0 for repeat trials. Validation and test output of the decoder was classified as correct if it was superior to 0.5 in search trials and inferior to 0.5 in repeat trials. Performance rates were computed as the number of time bins from all test trials combined incorrectly classified over the total number of these time bins. Every cross-validation returned the same ridge parameter (10^5) as eliciting the best performance with the validation data. Results of this analysis were therefore all obtained from test data with this ridge parameter. Finally, we trained the ridge regression readout on all the trials of each category and tested the decoder on all the time bins of 20 COR1 and 20 COR4 trials. Correct output of the COR1 test trials was defined as superior to 0.5 for time bins before the feedback and inferior to 0.5 for time bins from 300ms after the feedback to the end of the trial, while correct output of COR4 test was defined as inferior to 0.5 for time bins before the signal to change and superior to 0.5 for time bins from 1s after the signal to change to the end of the trial.

Results

Model performance

As expected, it was impossible for the model to learn to perform the task with a random search (41.64% ± 10.91% of suboptimal choices over 30 simulations, 1000 reservoir neurons) (see method for full description of performance calculation). In the absence of a pattern in the trained search sequence, the model could not produce a coherent output. In contrast, the model performed the task almost perfectly with the circular search (0.11% ± 0.43% of suboptimal choices over 30 simulations, 1000 neurons) and the ordered search (5.92% ± 5.40% of suboptimal choices over 30 simulations, 1000 neurons). As a comparison, the rate of suboptimal choices over all trials in monkeys assessed with the same method are 0.53% and 2.21% for each monkey, respectively.

Neural coding of task-related and behavioral variables

Two pertinent task-related variables in this task are the target choice, and an internal variable, corresponding to the phase (search or repetition). To explore the variance in activity explained by these two variables, we systematically tested each neuron of reservoir (total of 1000 neurons) in 30 independent simulations with the circular search using ANOVA, a commonly used parametric test in single-unit electrophysiology experiments (Mansouri et al., 2006; Cai and Padoa-Schioppa, 2012). Tests were performed at 5 different epochs: early fixation, late fixation, before touch, before feedback and after feedback. The majority of reservoir single units displayed significant main effects for choice and phase (three-way ANOVA, Epoch x Phase x Choice; mean ± std: Phase 97.6 ± 0.53% of a total of 1000 neurons; Choice 99.8 ± 0.13%). Table 1 (at the end of the article) reports results of the three-way ANOVA for all factors and their interactions. In order to see the coding of variables at specific epochs, two-way ANOVAs were conducted with factors Phase and Choice at each epoch (p-value < 0.05) and show more coding (Table 2, at the end of the article). These results did not differ with the ordered search training.

Focusing on pre and post feedback epochs, Quilodran et al. (2008) showed that some dACC neurons respond differentially depending on the phase of a problem. For the purpose of comparison with the model, we generalized this approach by assessing the difference in activity of the 85 dACC neurons that had at least 15 trials per condition with the same method used for the model: 59 neurons displayed a significant effect for phase (69%) and 57 a main effect for choice (67%).

Thus, single units within the recurrent reservoir network and the dACC encoded task related variables including target choice, and task phase, as revealed by significant main effects in ANOVA. Neither of these variables can be directly derived from the current external input. Rather, the target choice and phase depend on the history of previous inputs and responses, and thus indicate that reservoir and dACC neurons have access to this information. This access to previous inputs and states in reservoir activity, referred to as fading memory, is due to the recurrent dynamics and is part of the defining properties of reservoir computing (Jaeger, 2001; Maass et al., 2002). While such task related activity invites a straightforward explanation of cortical function, as noted above, recent research suggests that cortex does not uniquely rely on such simple coding schemes, and also displays more complex mixtures of selectivity to task related variables (Rigotti et al., 2013).

Mixed selectivity

As seen above, analysis of neural coding in terms of single task variables can provide an explanation of neural coding that appears simple. However, the activity of PFC neurons has often been described as complex, reflecting different combinations of task-related variables (Watanabe, 1986, 1992; Barone and Joseph, 1989; Sakagami and Niki, 1994; Asaad et al., 1998). This phenomenon has been the focus of several recent studies which revealed its importance for

cognitive tasks (Rigotti et al., 2010b, 2013). The analysis of Choice x Phase interactions can reveal whether reservoir neurons display mixed-selectivity properties (three-way Epoch x Phase x Choice ANOVA, Choice x Phase interaction, p-value < 0.05; see methods). Nearly all reservoir neurons showed mixed selectivity effects (99.9 ± 0.1% out of 1000 neurons, over 30 simulations) as revealed by the Choice x Phase interaction.

While mixed selectivity has been described in PFC neurons, no study has yet to our knowledge systematically and specifically explored it in the dACC. We thus sought to test the prediction that this same mixed selectivity, as revealed by the Choice x Phase interaction, should be present in dACC neurons in animals trained to perform the same task as the reservoir model. Our reanalysis of Quilodran et al. data indeed revealed that 33 out of 85 dACC neurons (39%) selected for the analysis displayed mixed selectivity. Figure 3 shows example activities for model and dACC with mixed selectivity effects at epoch before touch. Rigotti et al. (2010) showed that neurons of recurrent networks display complex recombination of current input that are similar to mixed-selectivity activities observed in PFC. These recombinations would allow external units connected to the recurrent network to detect the conjunction of several inputs. Recurrent networks of the reservoir type have the double property of recombining inputs and maintaining information about inputs across time (Dominey and Ramus, 2000; Jaeger, 2001; Maass et al., 2002). This is exemplified in the current experiment as the recombination of phase and choice information expressed in mixed selectivity that cannot be explained by a simple linear combination of the two variables contributions, and dependent on the history of previous inputs.

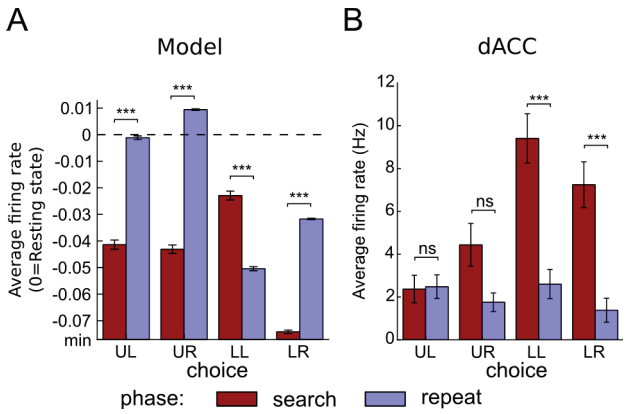

Figure 3: Reservoir model and dACC neurons showed mixed selectivity patterns between phase and choice variables. *Both figures show histograms of activity of a single unit averaged over all trials of a condition at epoch 'before touch'. (Error bars represent SEM). All statistical tests displayed on this figure correspond to two-sided pair-wise t-test with false discovery rate correction (p-value < 0.05).* **A.** *Model single activity significant for phase-choice interaction (ANOVA, Phase x Choice interaction, p-value < 10^{-15}) showing statistically higher activities for repeat phase when UL, UR and LR are chosen and the opposite pattern for LL target.* **B.** *A dACC neuron activity significant for phase choice interaction (ANOVA, Phase x Choice interaction, p-value < 10^{-5}) that is selective for phase only when lower left and lower right targets are chosen.*

Dynamic mixed selectivity

Modulation of selectivity through time is a well described feature of PFC neuronal activity, whether the neurons show selectivity at a specific time in a trial or shift selectivity within a trial (Quintana and Fuster, 1999; Mansouri et al., 2006; Cai and Padoa-Schioppa, 2012). We will refer to this pattern of selectivity that changes over time as dynamic selectivity. Model reservoir neurons display clear dynamic selectivity, with nearly all the neurons showing modulation of

selectivity across epochs for both phase and choice task variables as revealed by Epoch x Phase and Epoch x Choice interactions (mean ± std: 99.2 ± 0.4% for Epoch x Phase interaction and 99.7 ± 0.3% for Epoch x Choice interaction for 1000 reservoir neurons across 30 simulations, three-way ANOVA, Epoch x Phase x Choice, p-value < 0.05, Table 1). Figure 4A shows an example of such dynamic selectivity for phase. Single variables were encoded dynamically which suggests that mixed selectivity could be encoded in a dynamic fashion as well.

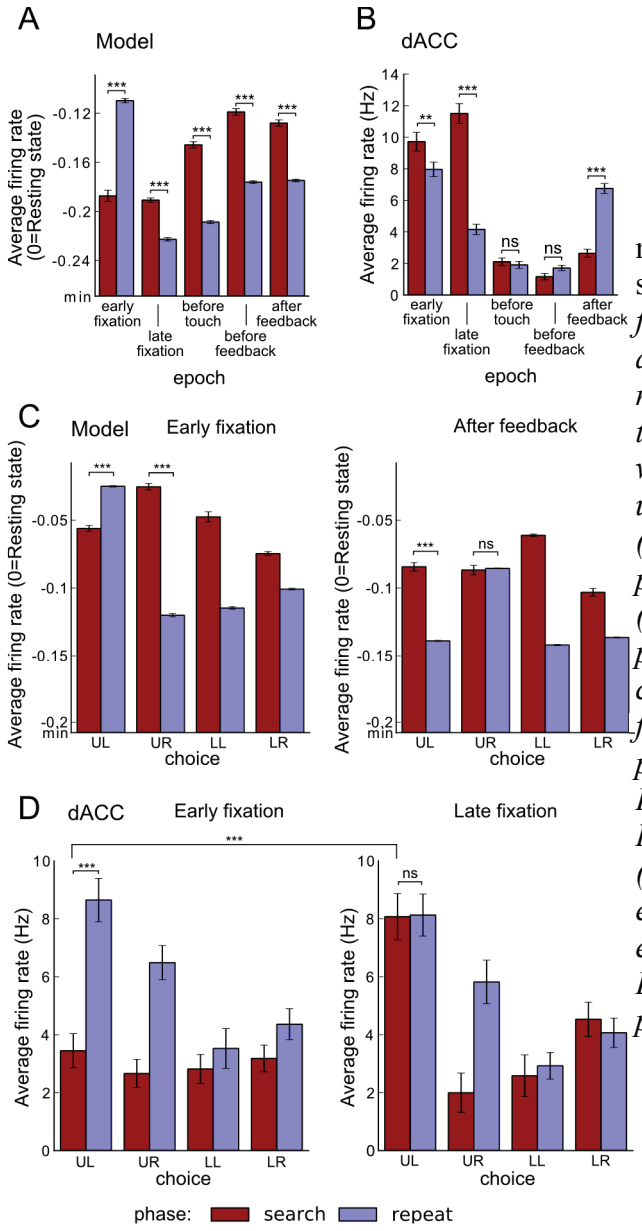

Figure 4. Single task variables and mixed selectivity are dynamically encoded, as shown here by single unit activity examples. A*ll figures show histograms of activity of a single unit averaged over all trials of a condition. Error bars represent SEM. All statistical tests displayed on this figure correspond to two-sided pair-wise t-test with false discovery rate correction.* **A**. *Reservoir unit selectivity for phase depends on epoch (ANOVA, Epoch x Phase interaction, p-value < 10^{-15}).* **B**. *Same as* **A** *for a dACC neuron (ANOVA, Epoch x Phase interaction, p-value < 10^{-15}).* **C**. *Reservoir unit activity for all conditions of phase and choice at epochs early fixation and after feedback showing two different patterns of mixed selectivity (ANOVA, Epoch x Phase x Choice interaction, p-value < 10^{-15}). Reservoir activity for phase depends on choice (left), and this dependence varies depending on epoch (right).* **D**. *Same as* **C** *for a dACC units at epochs early fixation and late fixation (ANOVA, Epoch x Phase x Choice interaction, p-value < 10^{-3}).*

Pursuing this dynamic aspect, in the following, a pattern of mixed selectivity that changes over time is referred to as dynamic mixed selectivity. We consider dynamic mixed selectivity when there was a significant three-way interaction between task epoch, phase and choice. Dynamic mixed selectivity was observed in the majority of model neurons (99.4 ± 0.4% of 1000 neurons, three-way ANOVA, interaction between Epoch x Phase x Choice, p-value < 0.05, Table 1 and Fig. 4C).

In order to determine if this property of the reservoir was equally observed in the primate data we performed the three-way ANOVA and examined the two way and three way interactions. Dynamic selectivity was similarly found in the dACC neurons selectivity (66 out of 85

neurons (78%) for Phase x Epoch interaction, and 56 out of 85 neurons (66%) for Choice x Epoch interaction, Table 1 and Fig. 4B), as well as dynamic mixed selectivity, as revealed by the Epoch x Phase x Choice interaction (16 out of 85 neurons, 19%, Table 1 and Fig. 4D). This prevalence of forms of dynamic selectivity suggests that the characterization of neurons as showing mixed selectivity features through the interactions of two variables excluding time can be extended to include the dynamical nature of these mixed-selectivity representations.

Unlike dACC, the vast majority of the neurons in the model were significant for each test. A possible explanation lies in the absence of noise in the whole network thereby producing very stable activities in each condition. Indeed, preliminary results showed that input noise slightly lowers the number of significant neurons in each test.

Reservoir networks are by nature dynamic, and representations within single reservoir neurons are themselves highly dynamic. While it was demonstrated that the presence of mixed selectivity in the activity of single neurons is important (Rigotti et al., 2013), and that this feature may be necessary to represent contingencies that are not represented in neurons firing only for single variables (Rigotti et al., 2010b), it is relatively non-intuitive to understand the role of dynamic mixed selectivity in producing complex but stable behaviors, leading to questions concerning how any pertinent information can be extracted from such unstable coding. Universal representation of all contingencies through dynamic mixed selectivity implies that a particular pattern of mixed selectivity in the activity of a single neuron will be transient. Such transient representations may be convenient for the system at a specific time, and this suggests that their stable representation through time may require adaptation within the network.

Task-relevant mixed selectivity

In the problem solving task the first reward is the key transient signal to stop exploring and concentrate on a single target, i.e. to initiate the repeat phase. Quilodran and colleagues found that this transition signal is represented in the activity of a sub-population of dACC neurons. These feedback-related neurons activated only after the feedback of the first rewarded trial (COR1, see Fig. 5A), and thus represented the conjunction of reward and search phase. Their activity was significantly different from incorrect (INC) and correct trials in repeat phase (COR) (12% and 24%, of 146 and 88 feedback related neurons in first and second monkey respectively, see Quilodran et al., 2008). This is an example of mixed selectivity that is highly relevant for the task, as a signal to shift from the search phase to the repetition phase, each of which has specific and distinct behavioral requirements.

In order to determine if neurons in the reservoir displayed this task relevant mixed selectivity COR1 response, we compared the average firing rate during the post feedback period (300ms to 800ms after the feedback) and found that $4.2\% \pm 3.2\%$ (mean $\pm$ std, 30 simulations, example in Fig. 5B) of the neurons had a significantly higher firing rate for the first rewarded trial than for the unrewarded and other rewarded trials (one-sided pairwise t-test with false discovery rate correction, p-value < 0.05). This result shows that the network encoded this mixed-selectivity feature of the task. While the number of neurons encoding the COR1 information in the reservoir was reduced with respect to the monkey (4.2% vs 12% and 24% respectively), its mere presence opens the possibility for extraction and strengthening for more explicit processing.

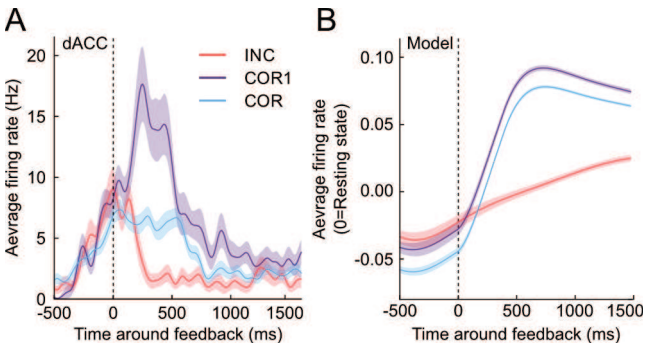

Figure 5. Example activities of reservoir and dACC neurons showing higher activity for feedback in COR1 trials. *Single neurons activities are averaged over incorrect (INC), first correct (COR1) and other correct (COR) trials and centered on the feedback (shaded areas correspond to SEM). A. Average activity of a dACC neuron showing a strong increase in activity after feedback for COR1 trials only (pairwise t-test with false discovery rate correction, COR1 > COR corrected p-value < 10^{-3}, COR1 > INC corrected p-value < 10^{-5}, Gaussian smoothing of firing rate with 35 ms standard deviation). B. Reservoir neuron from the model showing a significantly higher activity after feedback for COR1 trials (pairwise t-test with false discovery rate correction, COR1 > COR corrected p-value < 10^{-6}, COR1 > INC corrected p-value < 10^{-15}).*

Effects of contextual memory

While the reservoir displays a high percentage of mixed selectivity neurons, the number of COR1 neurons was reduced with respect to those found in the monkey. Because of the behavioral significance of the COR1 response (i.e. the transition from search to repeat), it is possible that the COR1 response in the monkey was increased by learning or adaptation that allows the system to represent the task phase more explicitly.

In this context, we observed that the reservoir required a high number of neurons (on the order of 1000) to perform the task optimally. Phase information in the network (i.e. whether the system is in search or repetition phase) was the consequence of the reward, yet this short reward input had a limited influence on the reservoir dynamics due to fading memory (Bernacchia et al., 2011). In a modified version of the model (contextual memory version), we trained one additional output neuron to represent phase. Like the other output neurons, this neuron was connected to the reservoir neurons by modifiable connections, and fed back its activity to the network as well (see Fig. 2A). It was trained to activate steadily only during the search phase, hence mimicking the tendency of numerous dACC neurons to display higher activity in search versus repeat phase (Quilodran et al., 2008). We refer to this as a "context" neuron as it encoded the phase, which provided a form of context. After learning, the context neuron activated following the signal to change input at the end of a problem, and fired steadily until the first reward. Note that the activation of the phase neuron was learned and was produced solely through learning of the readout connections to this phase neuron, just like the other output neurons. The only difference with respect to other output neurons is that rather than coding an actual behavioral response, it coded an internal state variable, the phase (search or repetition).

Interestingly, feeding this internally-generated contextual information back into the reservoir greatly reduced the number of suboptimal choices made by the model. The contextual memory version of the model performed very well with less than half the number of neurons in the reservoir layer originally required (Fig. 6). The initial version of the model compensated for the lack of explicit contextual memory with a higher number of neurons that provided extended memory of the reward (White et al., 2004; Verstraeten et al., 2007; Ganguli et al., 2008). In the contextual memory version of the model, since phase information was explicitly encoded with the

context neuron, neurons responding to the combination of phase and reward should be more numerous. Indeed, the percentage of neurons having a significantly higher activity for COR1 trials than for INC and COR trials was much higher with the context neuron feeding back phase information (13.2% ± 1.2%, p-value < 10^{-15}), approaching the proportions seen in the primate.

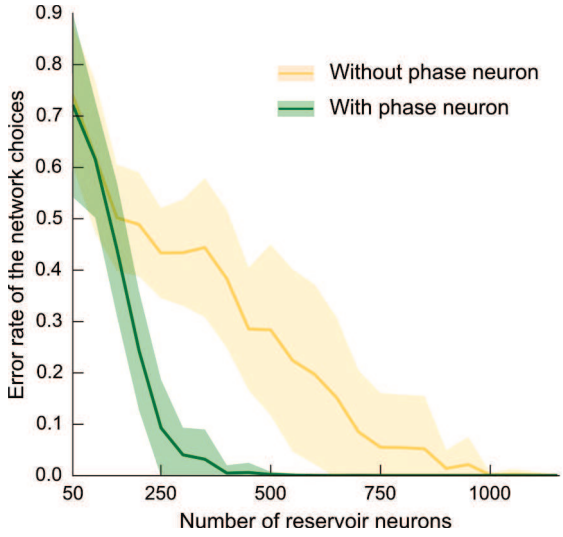

Figure 6. Error rates of the model as a function of the number of neurons in the reservoir layer. *The model required less neurons to perform well when phase information was explicitly represented in the trained context neuron. Error rates were computed as the ratio between the number of suboptimal trials (choices that likely postpone the reward) over the total number of trials. The reservoir required ~400 neurons to perform the task optimally with phase information explicitly encoded in a readout neuron, vs. ~1000 neurons that were required without the context neuron.*

While previous results illustrated the intrinsic capacity of reservoir network to encode complex combinations of previous and current inputs, here we demonstrate how information already present in the network can become explicitly encoded through learning. As a consequence, this strengthening of the internal representation allowed for combinations of internal and external inputs to be more widely represented in single unit activities, revealed by the increase in COR1 neurons, and for a significant performance increase, revealed by the reduced number of neurons required to solve the task.

Context is represented in attractors

So far we have explored single-unit activity to investigate the presence of mixed selectivity within the reservoir and dACC, therefore supporting the idea of ubiquitous dynamic mixed selectivity within PFC. However, population analysis can unveil dynamical aspects that are difficult to assess at the level of single units. In our case, the constant activation of the contextual phase neuron during search phase and its complete inactivation during the repeat phase may be sustained thanks to a population activity specific to each phase that was itself modified by the context neuron. This circular dependency with stable activities suggests an attracting dynamics that would depend both on the phase neuron and the reservoir population. Indeed, the principle of the learning algorithm used in this experiment (FORCE learning) is to converge on weights that shape attractors of the neural dynamics in order to produce a stable and robust dynamics (Sussillo and Abbott, 2009).

In order to reveal this specific dynamical property of the population activity we employed a PCA analysis to visualize the state trajectories of the reservoir neurons population in

order to assess if search and repeat population activity was well separated. Full trial activities were averaged for each neuron of the reservoir population over each trial of a problem: INC1 and INC2 were the first and second unrewarded search trials (INC3 trials were ignored due to lack of trials), COR1 the first rewarded trial, COR2-4 the repeat trials with the presentation of the signal to change at the end of COR4 trial (see methods for further details). Reservoir neuron population activity (without the context neuron) projected onto the first 3 dimensions show the cyclic nature of all the trajectories, representing the common path of successive events in each trial (51% of explained variance over 3 first dimensions, Fig. 7A). Moreover, search and repeat trajectories were only slightly separated after feedback but then collapse slowly and seemed to merge at the beginning of next trial. Since phase information was solely given by the reward, trajectories were most separated after this event but are indistinguishable at the middle of the next trial. We then performed the PCA on reservoir neurons in the contextual memory version of the model. In this version of the model, phase information was explicitly encoded in the network dynamics as illustrated in the PCA results that show two well separated sets of cyclic trajectories representing the search and repeat phases which suggest attracting dynamics (42% of explained variance over 3 first dimensions, Fig. 7B). The signal to change at the end of a problem deviated the end of COR4 trajectory towards the start of search trials as the contextual phase neuron activation was triggered. Likewise, the first reward during COR1 trial acted as signal in the transition from the search to the repeat trajectories as the context neuron shut off. Previous modeling and machine learning studies have shown how information representations could be maintained through attractors to develop a form working memory and participate in contextual processing of inputs (Durstewitz et al., 2000; Wang, 2001; Pascanu and Jaeger, 2011). These theories are supported by experimental results that also suggest rule representation and cognitive states through attractors (Durstewitz et al., 2010; Balaguer-Ballester et al., 2011; Mante et al., 2013).

We then performed the same PCA analysis with dACC data to determine if dACC similarly represented phase state in attractors and to compare the qualitative features of the trajectories with those of the model. In a similar manner, each neuron's activity was averaged over all the trials of the same type (a subset of 184/546 dACC neurons were selected, see methods). Projection of the dACC population activity onto the first two dimensions of PCA (51.4% of variance explained over 2 first dimensions, Fig. 7C) revealed similar cyclic trajectories representing the common path for all trials. However, these dimensions displayed slight trajectory differences between search and repeat trials, especially around feedback. This separation between search and repeat phase was much stronger with dimension 3 (42.5% of variance explained variance over dimension 1 and 3, Fig. 7D). dACC neural population activity differed most after feedback and until the end of the trial which is explained by the presence of a reward only in the repeat trials. Yet, trajectories were still separated between the start of a trial and the feedback. This strongly suggests that the population dynamics maintained separate states underlying behavioral context through attractors. As with the model trajectories, COR1 and COR4 trials displayed clear transitions between search and repeat trajectories. Our analyses support attracting dynamics as a plausible mechanism to implement contextual memory in neural networks to represent contextual information (Pascanu and Jaeger, 2011; Mante et al., 2013).

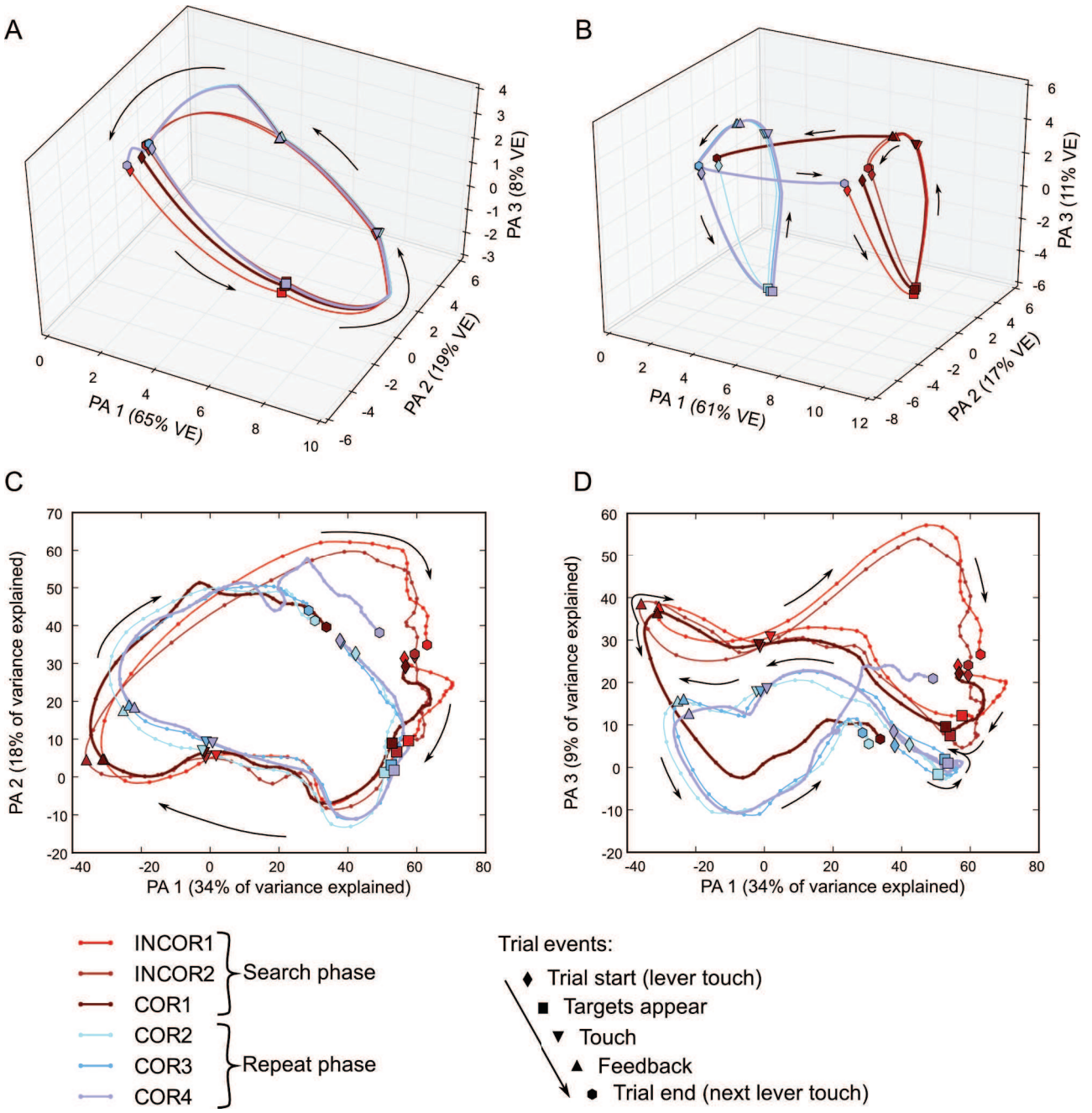

Figure 7. Reservoir and dACC population PCA-projected trajectories of successive trials in a problem show attractor-like dynamics. *Trajectory colors represent trials of each phase, reddish colors for search and blueish colors for repetition.* **A**. *Model population trajectories of average activity without context neuron (dimensions 1, 2 and 3; 93% of variance explained). All trajectories show a similar cyclic shape showing the successive events of a trial. Search and repeat trajectories are only separated after feedback and seem to collapse into a single trajectory at the middle of the next trial.* **B**. *Same as* **A** *with context neuron version of the model (90% of variance explained). Search and repeat trajectories form two well separated cyclic trajectories resembling two distinct attractors. COR1 and COR4 trials were the transition trials between the two sets of trajectories.* **C**. *dACC neural population trajectories of average activity for successive trials of a problem on dimension 1 and 2 of the PCA (51% of variance explained with both dimensions). Points represent successive 100ms of averaged activity. Trajectories show a cyclic pattern representing the path of activity in a trial. Search and repeat trajectories seem to overlap mostly between target appearance and touch events.* **D**. *Same as* **C** *with dimension 1 and 3 instead (42% of variance explained).*

Search and repeat trajectories are well separated suggesting an attractor like encoding of phase at the population level. (PA: principal axis, VE: variance explained)

Dynamic mixed selectivity yields local temporal structure

The single unit analyses revealed that mixed selectivity changes in time – that is, it is dynamic. The PCA analysis further revealed that at the population level, the model and dACC populations behave as dynamical systems, switching between two attractors. Here we sought to characterize the local temporal continuity of neural representations despite their dynamic nature. A method comparable to autocorrelation - cross-temporal pattern analysis (CTPA) - was recently used to demonstrate the dynamical aspect of information representation in neural population activity (Meyers et al., 2012; Stokes et al., 2013). This method allowed us to assess if phase information was represented in the population activity as a whole, and determine the dynamic character of its representation. In our case, the contrast of activity between search and repeat trials from one time point was correlated with the contrasted activity from all other time points of a trial (see methods for details). When the correlation between two time points is high, phase was represented in a similar pattern of population activity at these two time points. The correlation coefficients resulting from all cross-correlations are represented on a heat map where training and testing time are represented on the vertical and horizontal axis respectively, and the diagonal corresponds to training and testing at the same time point. Slow variations in the activity pattern elicit extended correlation around the diagonal, while dynamic coding in the population elicits narrow bands along the diagonal line. The continuous presence of correlation along the diagonal thus corresponds to a form of continuity in time for the neural representation of phase. In the analysis, search trials encompassed all INC trials. COR1 trials were not included as search trials for this analysis because transition from search to repeat occurred within these trials at the time of feedback as revealed by the PCA analysis. All remaining COR trials composed the repeat trials.

We investigated the dynamic nature of phase representation within the reservoir for both versions of the model and then compared it with dACC. The version of the model without context neuron represented phase as the propagation of the reward input in the network eliciting a large band of performance around the diagonal (Fig. 8A). We can observe that the coding of phase started with the feedback, increasing and then slowly diminishing as the distance from the feedback increases and finally becomes impossible to retrieve with the CTPA method at the end of the trial, though it was still present when the choice is made (between Targets ON and Touch in Fig. 8A). Indeed, phase information in the middle of trials is present in the network as confirmed by phase decoding with a regularized linear-regression (results not shown). Over time within a trial, the memory of the phase (as signaled physically by the reward) faded, corresponding to the characteristic fading memory of reservoir systems (Jaeger, 2001).

The contextual memory version of the model allowed a stable representation of phase as illustrated by near complete maximum correlation between all time points (Fig. 8B). The correlation measure narrows around the feedback but unlike the unmodified version of the model, correlation expanded instantaneously after the reward input switch off, revealing a very stable representation. Every point along the diagonal of the cross-temporal analysis show high correlation, characteristic of a phase representation throughout time over trials. This result is expected since phase neuron constantly fed the reservoir with phase information.

The same method was applied to 389/546 dACC neurons that were selected for having at least 20 search and 20 repeat trials (see methods). The analysis shows a large band of significant correlation that narrows concomitantly with feedback (Fig. 8C). From the start of the trial to feedback, phase representation evolved smoothly as shown by the statistically significant correlation coefficients for pairs of training/testing points separated by up to 1.5 seconds. Similarly to the model, representation of phase within dACC population was more dynamic at the time of

feedback presentation as illustrated in the narrow band of correlation that expands after feedback as if the representation of phase stabilized. Stokes et al. (2013) showed a similar pattern of dynamic coding at the time of target presentation that stabilized to represent context. We can consider that the dACC neurons in Fig. 8C correspond to an intermediate between the two versions of the reservoir without (Fig. 8A) and with (Fig. 8B) contextual memory where phase encoding in a stable readout activity is an extreme case of persistent activity.

These results indicate that there was a continuity of phase coding within the neural network that was similar in the reservoir and the dACC. In the original version of the model, phase information derived from the propagation of the reward feedback which first elicited a dynamic representation that stabilized and then faded in time. The introduction of a feedback unit that learned to represent phase allowed this representation to become stable over time.

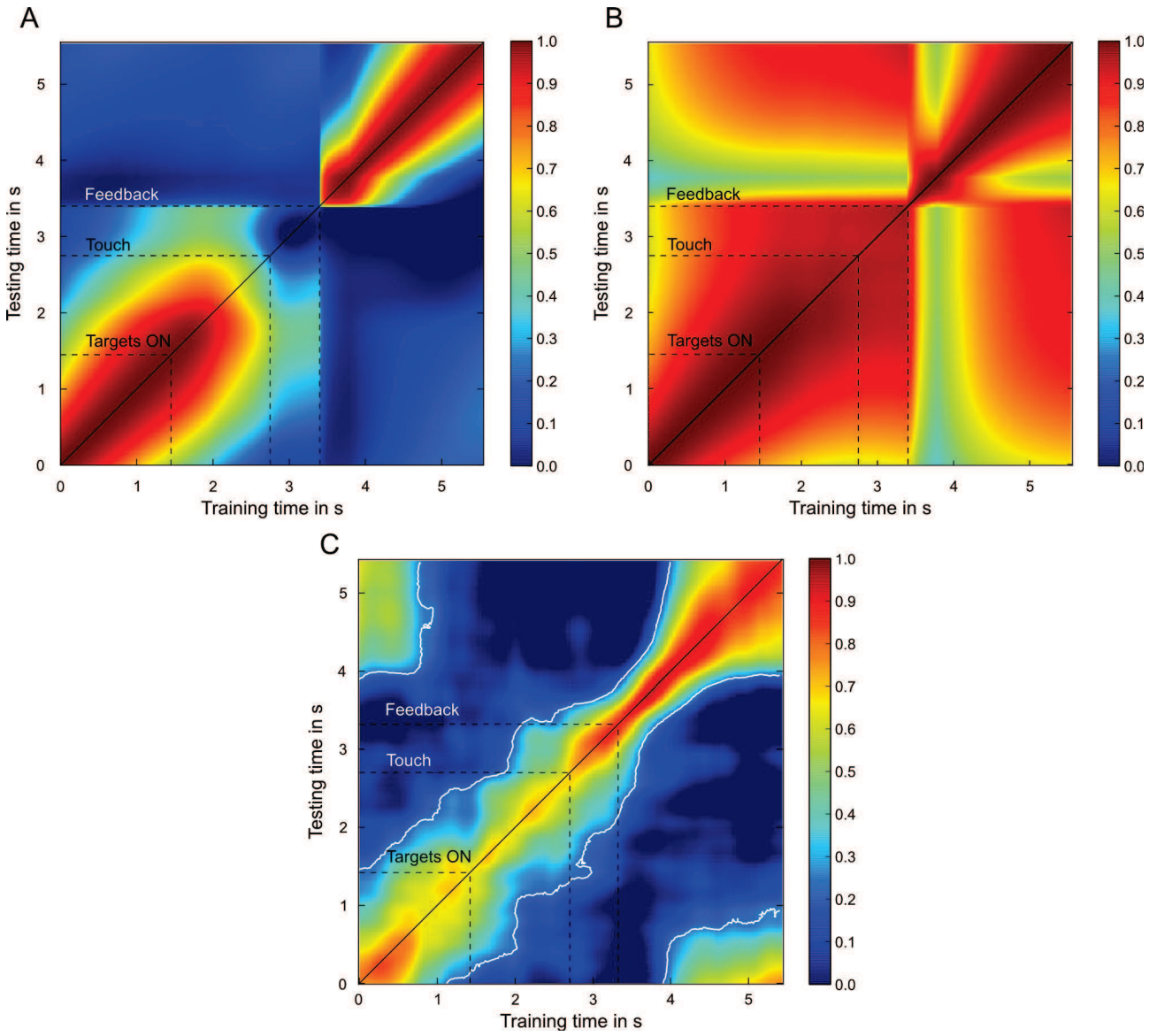

Figure 8. Cross-temporal pattern analysis (CTPA) of phase representation in model and dACC population activity. *This analysis shows the dynamic nature of phase representation through correlation of activity contrasted between search and repeat. The diagonal represents the correlation coefficient between activities from the same point in time. Narrow bands of correlated contrast around the diagonal are typical of dynamic representation while wider spread correlations signals a more static representation.* **A**, *CTPA of reservoir population from original model version. Analysis results display a diagonal of performance with a gap in the middle of trials, between touch and feedback. CTPA method could not retrieve phase information at this period of trials thereby showing the weakness of phase representation in the activity of reservoir neurons.* **B**, *Same as* **A** *with a contextual memory version of the model. Phase representation within reservoir population was extremely stable as shown by the highly correlated contrasts between every pair of time points except after feedback. Reward had a strong influence on the representation of phase which is observed here as cross of lower correlation centered on the diagonal at the time of feedback.* **C**, *Same as* **A** *and* **B** *on time normalized dACC data. White line represents the contour of significant values (permutation test with 10^4 permutations, p-values < 0.001). Representation of phase in dACC population evolved smoothly during the course of a trial. Phase was well represented through the full length of a trial but was more dynamic at the time of feedback which was the event specifying phase. Phase representation in dACC seems to be best described as an*

intermediate between the original and the context neuron version of the model.

Dynamic mixed selectivity can generate stable output

We have shown that neurons in the model displayed complex and dynamic activities that are not easily interpretable as to their contribution to stable and controlled output. That is, the mapping of this complex time-varying activity onto coherent behavior is not immediately evident. However, both monkeys and the model (both original and contextual memory versions) were quite efficient at performing the task. This implies that in both, there was a mechanism that can extract coherent task-relevant representations from dynamic mixed selectivity. Recent studies suggest that complex non-linear combinations of inputs create rich activities in recurrent networks from which relevant information is easily extracted using simple linear regression (Buonomano and Maass, 2009; Rigotti et al., 2013). This decoding of complex dynamics by linear readout is by definition the construction of a RC network that projects inputs into a higher dimensional space which ease the process of separating (classifying) inputs.

Here, we attempted to determine whether a linear readout could be used to generate stable outputs across time from dynamic and complex activity. For this purpose we trained a readout unit, with the method used for other readouts units, to represent the phase state without feeding it back to the reservoir. Thus, the readout unit had to extract phase from a reservoir population that did not receive this readout as a feedback. The learning procedure converged on weights that activated this special readout neuron throughout the search phase even though phase representation was dynamic. Figure 9A illustrates the target choice readout activity of the model along with the readout unit extracting phase after the task has been learned (results were replicated on 30 simulations). The output of the phase readout unit was steady during search phase and activated and deactivated sharply with key events (signal to change and first reward), thereby demonstrating the precision that can be obtained if adequate mechanisms read out the complex dynamic mixed activity generated in a reservoir network. When compared with Figure 8A, even during the task period (before the feedback) when the CTPA values of the model without context neuron are reduced, the trained context neuron stably represented the context (i.e. search or repetition phase).

A simple linear readout could extract phase state steadily from the reservoir population activity that seemed dynamic. Astrand et al. (2014) recently showed that one of the best decoder for task related variables on a population of prefrontal neurons is a simple regularized linear-regression which elicits performance similar or superior to those using complex machine learning methods like Support Vector Machine while being simpler and less expensive computationally. We thus set to reproduce the continuous decoding results with dACC activity by training a readout through ridge regression. The readout was trained and then tested on the activity of full search (INC) and repeat (COR2-3) trials to see if it could continuously decode phase state (290/546 neurons having at least 20 trials of each category, cross-validation, see methods for details). This method allowed us to assess if a single readout, that can be considered as a downstream neuron connected to the dACC population, can continuously extract phase information. On average, over all time bins of every search and repeat trials tested, the decoder made only 3.0% ± 2.1% of errors (mean ± std). We then tested this decoder trained on all INC and COR2-3 trials with COR1 and COR4 as test trials to explore the transition between search and repetition in decoding phase with the linear readout. Over all time bins of all COR1 trials (20 trials), the percentage of misclassified time bins was 5.8% ± 5.4% (Fig. 9B). Likewise, the error was very low for COR4 trials (20 trials) with a 8.1% ± 8.8% error rate (Fig. 9C). These results demonstrate the success of a simple linear readout to continuously extract phase information from a dACC population activity that was globally dynamic. As suggested by the model results, cortical recurrent connections may expand inputs into high dimensional and dynamic activities that could be simply read-out linearly by a downstream neuron (Buonomano and Maass, 2009).

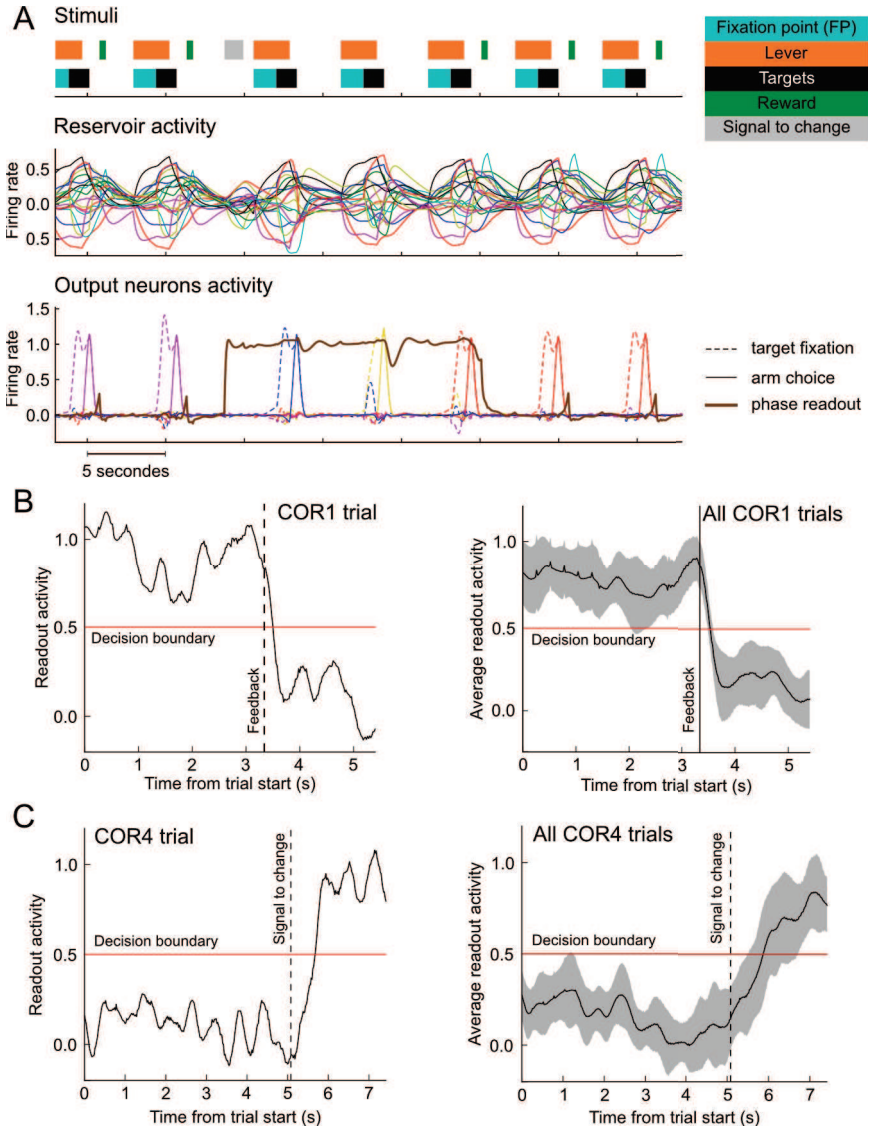

Figure 9. Continuous phase readout from reservoir and dACC activity. **A**. *Example stimuli sequence, corresponding reservoir activity and model output during test showing the stability of phase readout. Upper panel shows a selection within the stimuli input sequence fed to the reservoir during test. Color blocks represent activity in of the reservoir input neurons simulating the different features of the task. Middle graph shows the activity of 20 randomly picked reservoir neurons illustrating the dynamic nature of activity in the reservoir. Lower graph shows the evolution of the readout neurons that represent target fixation (dashed lines) and hand touch (solid lines) along with the activity of a unit reading out phase (brown line). This special readout unit was not connected back to the reservoir and could not rely on the attractors created with a feedback connection. This portion of stimuli sequence starts with the last trial of a problem. The signal to change is given to the model (grey block) activate the phase readout unit which fires steadily during the following three trials corresponding to the search phase showing the stable output that can be obtained from the dynamic reservoir activity. The third target exploration is rewarded (COR1 trial) and ends the search phase. At COR1 trial, when the reward neuron activates, the phase readout unit shuts off thereby signaling the start of the repeat phase. The rewarded target choice (red lines) is then repeated in subsequent trials.* **B**. *Left graph illustrates phase readout of a single COR1 trial. Output of the readout was considered correct if superior to the decision boundary (0.5) for time bins before the feedback (transition from search to repeat) and inferior to the decision boundary from 300ms after the feedback to the end of the trial. In this example the readout made no errors. Right graph*

shows the average readout output across all COR1 trials. Shaded area corresponds to the standard deviation. **C**. *Same as* **B** *for COR4 trials. In these trials transition from repeat to search happened at the signal to change. Readout output was considered correct if inferior to the decision boundary before the signal to change and if superior to the decision boundary from 1s after the signal to change onset until the end of the trial. Transition from repetition to search seemed slower than the transition from search to repetition.*

Discussion

Rigotti et al. (2010) recently postulated that randomly connected recurrent networks could account for a form of universal combinations of inputs to the system. One of the principal characteristics of these networks is the mixed selectivity to task-related factors that is observed in the recurrent neurons. This same mixed selectivity in single cortical units is also crucial for the successful performance of these tasks by trained primates (Rigotti et al., 2013). In many tasks involving executive function, the context provided by previous inputs will be required for meeting current behavioral demands. Past inputs that influence future behavior create a context that forms temporal bridges between past events and the current situation. Context should be internally represented and contribute to the mixed selectivity combinations and universal representation of contingencies that will include past events. Thus, mixed selectivity must be dynamic.

The first study to directly demonstrate context encoding in a sequence learning task in the monkey was undertaken by Barone and Joseph (1989). PFC neurons were selective for spatial target location and their sequential rank, displaying a form of dynamic selectivity. During this same period, the sequence coding properties of recurrent networks was explored (Pineda, 1987; Cleeremans et al., 1989; Pearlmutter, 1989; Williams and Zipser, 1989). Elman (1990) employed error back-propagation learning in a recurrent network, and demonstrated that the network could adapt to the statistical structure of the training data (Elman, 1991). To eliminate the complexity of training recurrent connections, time was cut off after one cycle through the recurrent network, thus yielding the simple recurrent network SRN. This simplification diminished the SRN's ability to handle temporal dynamics.

To retain the temporal dynamics of recurrent networks, we developed a model of prefrontal cortex as a recurrent network in which the recurrent connections were fixed, and learning occurred in the output connections – corresponding to physiologically motivated cortico-striatal plasticity (Dominey, 1995). Simulating the sequence learning task of Barone and Joseph (1989), we thus observed the same sequence context effects (combining target and sequence rank) in our recurrent neurons (Dominey et al., 1995). This temporal recurrent network was the first expression of the reservoir computing framework, where a network of recurrently connected elements with fixed connections projects to a set of output neurons with trainable connections. This concept was subsequently elaborated by Maass et al. (2002) with the liquid state machine, and Jaeger (2001) with the echo state network. Maass and colleagues examined how these recurrent networks can perform complex computations, and he also importantly demonstrated that the primate cortex displays coding properties characteristic of these reservoir networks (Buonomano and Maass, 2009; Nikolić et al., 2009).

In the current study, we trained a reservoir network to perform a cognitive task that requires contextual memory, both for applying the correct search and repetition strategies and for keeping track of when to switch strategies. We systematically examined task related coding of target choice and task phase over different epochs of trials. We observed all possible main effects, and two-way and three-way interactions of these effects revealing dynamic mixed selectivity within the network. Target choice and phase interact, and this interaction is further influenced by time or epoch. We then confirmed that primate dACC displays these dynamic mixed selectivity effects. The inputs are recombined in a higher dimensional space that contain potentially all possible recombinations and may be a mark of the representational power of such networks. In this respect, recurrent networks are sometimes compared to Support Vector Machine which is state of the art machine learning (Lukoševičius and Jaeger, 2009). To support the hypothesis that the cortex relies on such mechanisms Rigotti et al. (2013) demonstrated that high dimensional neural activity is required for monkeys to successfully perform.

In this dynamic mixed selectivity framework, all combinations of recent and current inputs

can be represented by a reservoir network, allowing it to have readily available representations of all contingencies. Learning then provides a means to isolate and strengthen behaviorally relevant activities that may be initially weak. Indeed, our reservoir model represented the important COR1 information in small differences of activities compared to the clear dACC COR1 activity. We can postulate that prior to learning some dACC neurons may have had small differences in activity as observed in the reservoir neurons, and that learning subsequently amplified this difference to express COR1 information in a more explicit and robust way, so that it could be distributed within the network and transmitted to other areas. Mante et al. (2013) similarly demonstrated how a recurrent network could be trained to perform a context-selective task, and that the resulting system had attractor properties that were appropriately adapted to the task. Thus, in general we can consider that a universal coding capability inherent to randomly connected recurrent networks provides the possibility to represent all task-related contingencies, and that learning and adaptation enable a focus on and adaptation to the ecologically pertinent contingencies. A developmental form of universal coding and subsequent adaptation has been observed for the vowel space in the auditory system that is initially universal and then focused to the native language in early development (Kuhl et al., 1992).

Interestingly, due to fading memory, context information (related to feedback) present in the network has a limited life time in the version of the model without context neuron. It may be amplified to bridge inputs separated through long intervals. Our model experimentation illustrates how the phase information can be robustly encoded in the network through learning. Indeed, we showed at the population level that explicitly representing phase in a steadily active neuron elicits the formation of an attracting dynamics that represent phase information in a stable state. Seminal work from Amit (1989) developed the hypothesis that attractors of the neural dynamics could account for sustained activity in the prefrontal cortex that would underlie working memory. This theory has since received wide attention and remains one of the most accepted mechanism behind working memory (Wang, 2001) and other cognitive functions including decision making (Wang, 2008).

Population level analyses inspired by Stokes et al. (2013) and Meyers et al. (2012) showed how highly volatile and time-limited representation of phase can be turned into a stable, and long lasting representation through learning. Again, based on these observations in the model, we performed the same analyses in dACC and observed the same attractor-based population coding of phase, contributing to the characterization of dACC population dynamics. Our model of context learning demonstrates the capacity of a quite simple architecture to develop attracting dynamics robustly representing phase through time. In our context neuron version of the model, phase representation seemed extremely stable compared to what dACC activity showed. We suppose that the dACC dynamics is best described as a stable transient dynamics (Durstewitz and Deco, 2008; Rabinovich et al., 2008; Rabinovich and Varona, 2011) that lie between the purely input driven representation of phase in the original version of the model, and phase context neuron version which showed strong attracting dynamics.

The dynamic mixed selectivity observed at the single unit level renders the task of interpreting single neuron activity difficult. Yet our model illustrates how populations of complex activities can contribute to stable output activity and, as a consequence, to stable behavior. Based on the parallel established between the model and the dACC we can consider that dynamic mixed selectivity in dACC must be interpreted at the population level.

By definition, an adaptive system when faced with a novel and unpredicted situation will be capable of extracting pertinent regularities from the environment, in order to behave appropriately. In complex environments, the notion of behaviorally pertinent is not always directly available from the current sensory input. Instead, it is based on the interaction between the current sensory input and the (more or less) recent history of inputs, internal states, and outputs of the system. One could conceive different strategies for creating a system capable of adapting to such conditions. We

believe that analyzing the relation between reservoir networks and PFC helps to understand how primate prefrontal cortex addresses this problem. Through the projection of input into a higher dimensional space where they interact with internal states, PFC provides a pre-coding of an essentially universal set of combinations of input and internal states; these can then be selected and used in application to the arbitrarily (but structured) situations that the system will encounter. Through learning and adaptation, these pertinent representations can be further refined for the specific needs of the system.

Our model describes how a recurrent network can represent universal combinations of inputs and internal states. In the future, a more advanced version of this model would explore the biological plausible mechanisms by which the system amplifies itself the relevant information like COR1 to express them more robustly in order to face the inherent noise present in inputs. With such a mechanism, new information is robustly coded in the network and can then participate in new combinations, thereby bringing representation of contingencies of internal states that are not directly accessible from the sensory inputs.

References

Amiez C, Sallet J, Procyk E, Petrides M (2012) Modulation of feedback related activity in the rostral anterior cingulate cortex during trial and error exploration. Neuroimage 63:1078–1090.

Amit DJ (1989) Modeling Brain Function: The World of Attractor Neural Networks, Cambridge . New York.

Asaad WF, Rainer G, Miller EK (1998) Neural activity in the primate prefrontal cortex during associative learning. Neuron 21:1399–1407.

Astrand E, Enel P, Ibos G, Dominey PF, Baraduc P, Ben Hamed S (2014) Comparison of classifiers for decoding sensory and cognitive information from prefrontal neuronal populations. PLoS One 9:e86314.

Balaguer-Ballester E, Lapish CC, Seamans JK, Durstewitz D (2011) Attracting dynamics of frontal cortex ensembles during memory-guided decision-making. PLoS Comput Biol 7:e1002057.

Barone P, Joseph J (1989) Prefrontal cortex and spatial sequencing in macaque monkey. Exp Brain Res 78:447–464.

Bartlett EL, Wang X (2005) Long-lasting modulation by stimulus context in primate auditory cortex. J Neurophysiol 94:83–104.

Bernacchia A, Seo H, Lee D, Wang X-J (2011) A reservoir of time constants for memory traces in cortical neurons. Nat Neurosci 14:366–372.

Brosch M, Schreiner CE (2000) Sequence sensitivity of neurons in cat primary auditory cortex. Cereb cortex 10:1155–1167.

Buonomano D V, Maass W (2009) State-dependent computations: spatiotemporal processing in cortical networks. Nat Rev Neurosci 10:113–125.

Cai X, Padoa-Schioppa C (2012) Neuronal encoding of subjective value in dorsal and ventral anterior cingulate cortex. J Neurosci 32:3791–3808.

Cleeremans A, Servan-Schreiber D, McClelland JL (1989) Finite State Automata and Simple Recurrent Networks. Neural Comput 1:372–381.

Dominey P, Arbib M, Joseph J (1995) A model of corticostriatal plasticity for learning oculomotor associations and sequences. J Cogn Neurosci 7:311–336.

Dominey PF (1995) Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. Biol Cybern 73:265–274.

Dominey PF, Ramus F (2000) Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant. Lang Cogn Process 15:87–127.

Durstewitz D, Deco G (2008) Computational significance of transient dynamics in cortical networks. Eur J Neurosci 27:217–227.

Durstewitz D, Seamans JK, Sejnowski TJ (2000) Neurocomputational models of working memory. Nat Neurosci 3 Suppl:1184–1191.

Durstewitz D, Vittoz NM, Floresco SB, Seamans JK (2010) Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. Neuron 66:438–448.

Elman JL (1990) Finding Structure in Time. Cogn Sci 14:179–211.

Elman JL (1991) Distributed representations, simple recurrent networks, and grammatical structure.

Mach Learn 7:195–225.

Ganguli S, Huh D, Sompolinsky H (2008) Memory traces in dynamical systems. Proc Natl Acad Sci U S A 105:18970–18975.

Hinaut X, Dominey PF (2013) Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. PLoS One 8:e52946.

Jaeger H (2001) The“ echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology

Jaeger H, Haas H (2004) Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304:78–80.

Kilgard MP, Merzenich MM (1999) Distributed representation of spectral and temporal information in rat primary auditory cortex. Hear Res 134:16–28.

Kuhl P, Williams K, Lacerda F, Stevens K, Lindblom B (1992) Linguistic experience alters phonetic perception in infants by 6 months of age. Science (80-) 255:606–608.

Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 3:127–149.

Maass W, Natschläger T, Markram H (2002) Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput 14:2531–2560.

Mansouri FA, Matsumoto K, Tanaka K (2006) Prefrontal cell activities related to monkeys' success and failure in adapting to rule changes in a Wisconsin Card Sorting Test analog. J Neurosci 26:2745–2756.

Mante V, Sussillo D, Shenoy K V., Newsome WT (2013) Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature 503:78–84.

Meyers EM, Qi X-L, Constantinidis C (2012) Incorporation of new information into prefrontal cortical activity after learning working memory tasks. Proc Natl Acad Sci U S A 109:4651–4656.

Miller EK, Cohen JD (2001) An integrative theory of prefrontal cortex function. Annu Rev Neurosci 24:167–202.

Nikolić D, Häusler S, Singer W, Maass W (2009) Distributed fading memory for stimulus properties in the primary visual cortex. PLoS Biol 7:e1000260.

Pascanu R, Jaeger H (2011) A neurodynamical model for working memory. Neural Netw 24:199–207.

Pearlmutter BA (1989) Learning State Space Trajectories in Recurrent Neural Networks. Neural Comput 1:263–269.

Pineda F (1987) Generalization of back-propagation to recurrent neural networks. Phys Rev Lett 59:2229–2232.

Procyk E, Goldman-Rakic PS (2006) Modulation of dorsolateral prefrontal delay activity during self-organized behavior. J Neurosci 26:11313.

Quilodran R, Rothé M, Procyk E (2008) Behavioral shifts and action valuation in the anterior cingulate cortex. Neuron 57:314–325.

Quintana J, Fuster J (1999) From perception to action: temporal integrative functions of prefrontal

and parietal neurons. Cereb cortex 9:213–221.

Rabinovich M, Huerta R, Laurent G (2008) Transient dynamics for neural processing. Science (80-) 321:48–50.

Rabinovich MI, Varona P (2011) Robust transient dynamics and brain functions. Front Comput Neurosci 5:24.

Rigotti M, Barak O, Warden MR, Wang X-J, Daw ND, Miller EK, Fusi S (2013) The importance of mixed selectivity in complex cognitive tasks. Nature.

Rigotti M, Ben Dayan Rubin D, Morrison SE, Salzman CD, Fusi S (2010a) Attractor concretion as a mechanism for the formation of context representations. Neuroimage 52:833–847.

Rigotti M, Rubin DBD, Wang X-J, Fusi S (2010b) Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. Front Comput Neurosci 4:24.

Sakagami M, Niki H (1994) Encoding of behavioral significance of visual stimuli by primate prefrontal neurons: relation to relevant task conditions. Exp Brain Res 97:423–436.

Stokes MG, Kusunoki M, Sigala N, Nili H, Gaffan D, Duncan J (2013) Dynamic Coding for Cognitive Control in Prefrontal Cortex. Neuron 78:1–12.

Sussillo D, Abbott LF (2009) Generating coherent patterns of activity from chaotic neural networks. Neuron 63:544–557.

Tong MH, Bickett AD, Christiansen EM, Cottrell GW (2007) Learning grammatical structure with Echo State Networks. Neural Netw 20:424–432.

Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D (2007) An experimental unification of reservoir computing methods. Neural Netw 20:391–403.

Wang X-J (2001) Synaptic reverberation underlying mnemonic persistent activity. Trends Neurosci 24:455–463.

Wang X-J (2008) Decision making in recurrent neuronal circuits. Neuron 60:215–234.

Watanabe M (1986) Prefrontal unit activity during delayed conditional Go/No-Go discrimination in the monkey. I. Relation to the stimulus. Brain Res 382:1–14.

Watanabe M (1992) Frontal units of the monkey coding the associative significance of visual and auditory stimuli. Exp Brain Res 89:233–247.

White O, Lee D, Sompolinsky H (2004) Short-Term Memory in Orthogonal Neural Networks. Phys Rev Lett 92:148102.

Williams RJ, Zipser D (1989) A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Comput 1:270–280.

Tables

	dACC		Model
	# of significant neurons	% of significant neurons	average # of significant neurons
Phase	59	69.41	975.58 ± 5.31
Choice	57	67.06	998.47 ± 1.33
Epoch	84	98.82	1000.00 ± 0.00
Epoch * Phase	66	77.65	991.77 ± 3.95
Epoch * Choice	56	65.88	997.07 ± 2.63
Phase * Choice	33	38.82	999.23 ± 0.99
Epoch * Phase * Choice	16	18.82	993.87 ± 4.30

Table 1. Results of a three-way ANOVA (Epoch x Phase x Choice, p-value < 0.05) on reservoir (30 simulations, average ± std) and dACC neurons showing the number of neurons significantly modulated by the factors shown on the first column.

	dACC			Model		
	Phase	Choice	Phase X Choice	Phase	Choice	Phase X Choice
Early fixation	50	26	15	96.5 ± 0.9	98.4 ± 0.7	99.5 ± 0.3
Late fixation	32	31	16	96.5 ± 0.8	98.6 ± 0.6	99.6 ± 0.2
Before touch	36	43	13	96.3 ± 0.8	99.0 ± 0.6	99.6 ± 0.2
Before feedback	50	40	15	95.7 ± 0.9	99.5 ± 0.3	99.6 ± 0.3
After feedback	60	29	12	93.5 ± 1.6	99.9 ± 0.1	99.4 ± 0.5
At least 1 epoch	84	71	45	100 ± 0	100 ± 0	100 ± 0

Table 2. Results of a two-way ANOVA (Phase x Choice, p-value < 0.05) performed on successive epochs of a trial. Model results show percentages of significant neurons ± std (over 30 simulations). Last row counts the number of neuron that show significant effect for the factor for at least one epoch.

Part III

Discussion

Chapter 8

Discussion

In this discussion we will first review the results from the first article (chap. 6) that does not contain a discussion related to our hypothesis. Then we will review the results of the second article (chap. 7) and contrast them with our hypothesis. Finally, we will suggest future experiments directly related to our model followed by the proposition of a new modeling approach.

8.1 Task Variable Representation and Readout Mechanisms

The goal of the first experiment was to assess the capacity of six decoders in retrieving variables of a cognitive task. As a short reminder, macaque monkeys had to fixate a central point on a screen while two visual streams were presented on the screen. A cue appeared embedded in one of the streams and its color indicated the position of an upcoming target that the animals had to fixate. Two task variables were extracted from the activity of FEF neurons, namely, the position of the cue (exogenous information) and the interpreted position of the upcoming target (endogenous information). A striking result is that the endogenous information was as efficiently decoded as the exogenous information. Three findings of this study are of particular interest for our hypothesis. First, neurons not classified as specifically responsive to the upcoming position of the target (attention cell) contributed to increase the performance of most of the decoders. Second, a simple regularized linear decoder was as efficient as the state of the art in machine learning (namely, Support Vector Machine). Third, a "reservoir" with short time constants and without recurrent connections[1] can continuously decode the endogenous

[1] Note that the name "reservoir" is no longer appropriate for this network since there are no more recurrent connections within the hidden layer.

information. We will clarify each point, in the following sections.

8.1.1 "Non-Selective" Neurons Participate in Robust Representations

A subset of 21 over 131 recorded FEF neurons were categorized as attention cells if their activity between the presentation of the cue and the appearance of the target could significantly predict the position of the target. While the position of the upcoming target could be decoded efficiently from this small set of neurons, adding the remaining population of neurons increased the decoding performance. The whole population seemed to represent the attention position more reliably. According to the theory we support, the cortex should inherently represent relevant contingencies in activities that are highly complex and seemingly unrelated to these contingencies. So we expect that neurons which are not classified as attention cells nonetheless encode both task variables with distributed representations. Hence, decoding should be improved when one includes non-selective neurons, which is the result obtained here. Indeed, distributed activity underlies the representational power of the cortex, which is in part explained by the increased dimensionality of the population activity that eases the separation of task variables. Likewise, in chapter 1 we mentioned a study which showed that task variables can be relatively well decoded when the neurons most statistically correlated to the decoded task variable were removed (Meyers et al., 2012).

8.1.2 Extracting Task Variables with a Simple Linear Decoder

Globally, this first experiment found that the best decoders are a Support Vector Machine (SVM) and a regularized linear regression. This means that a readout as simple as a linear regression can extract task variables from a population of PFC neurons as efficiently as state of the art machine learning. SVMs expand their inputs into a higher dimensional space in order to better separate them with a linear hyperplane. In this experiment it seems that the expansion used by SVMs is not necessary to extract both task variables. A confirmation lies in the optimization of the reservoir decoder. In order to perform optimally, reservoir parameters must be explored to create the most interesting dynamics inside the recurrent layer. We observed that the optimal reservoir had no recurrent connections, and that its neurons did not use leaky integration. Interestingly, the resulting network was a feedforward layer with one hidden layer, learning taking place only between the hidden layer and the readout. In other words, it was an original Perceptron with randomly connected connections between the input and the hidden layer that do not process temporal information. Thus, this network was closer

to the linear decoder that elicit the best performance, strengthening the observation that a linear readout can be sufficient to extract task variables from cortical activity. However, expansion of neural activity may be necessary to reconstruct its original high dimensional space in the case when only a few neurons have been recorded (Balaguer-Ballester et al., 2011). But, in our case, since a further expansion with the SVM do not facilitate the extraction of task variables, we postulate that the activity was already high dimensional, a property that contributes to the representational power of the cortex.

8.1.3 Continuous Decoding of Task Variable

Finally, training a decoder on continuous activity resulted in sustained decoding performance. After optimization, the network decoder, which was originally a reservoir, had no recurrent connections and a short time constant for leaky integration, making each neuron in the network a running average of the combination of its recent inputs. This result shows that a readout mechanism can continuously extract a task variable without changing its weights even if the firing rate of neurons within the population source is dynamic. Reservoir computing uses the same mechanism; once trained, the readout layer is static and can extract and segregate inputs by separating trajectories within the activity of the recurrent layer (Buonomano and Maass, 2009). The second article of this thesis also demonstrate continuous extraction with a linear regression of a dynamically-represented task variable. Consequently, we assume that a simple mechanism like a linear regression can also continuously decode the variables of the present task. This means that a cortical neuron that receives afferent connections from the recorded population could similarly continuously extract the task variable (Buonomano and Maass, 2009). We will further explore this question in the next sections.

8.1.4 Conclusion

Activity within the FEF represents task variables in a few very specialized cells but also in activities that do not seem a priori to be selective for these task variables (Meyers et al., 2012). Since a linear decoder can extract the task variables as efficiently as methods that expand the activity into a higher dimensional space, we argue that the activity of the cortex may already be high dimensional, and that this property is present before learning takes place. Furthermore, a static network can continuously decode a task variable, meaning that a similar cortical network could also efficiently extract this information continuously. We will review in the next section an experiment of the second article (chap. 7) that demonstrates continuous decoding of a variable represented in dynamical activity with a simple linear regression.

8.2 Modelling Cortical Representations with a Reservoir

The goal of the second experiment was to demonstrate the mechanism of information representation in the cortex through the comparison of dorsal anterior cingulate cortex (dACC) single neuron activity with a reservoir model. We first demonstrated that a randomly recurrent network shows rich spatio-temporal representations that we denoted dynamic mixed selectivity in reference to the work of Rigotti et al. (2013). We showed that representing explicitly contextual information (i.e. searching the rewarded target or repeating the last target) in the network led to the creation of two attractors that produced different output behaviors depending on the context. A Principal Component Analysis suggest similar dynamics in the dACC. Finally, we found that the context can be continuously extracted with a simple static linear readout from the dynamic population activity of the model and the dACC. In the following paragraph we will confront specific points with our main hypothesis. A full discussion of the results can be found at the end of the corresponding article (chap. 7).

8.2.1 Dynamic Mixed Selectivity

In the selectionist approach that we defined in the state of the art section, adaptive behavior arises from the iterative selection of pre-existent combinations of available inputs (variation process). Our main hypothesis is that a striking property of the cortex, its highly recurrent local connectivity, endows it with a universal spatio-temporal representation with fading memory, and would underlie the variation process. In this context, our second experiment first demonstrated that every instance of a simple recurrent network with randomly-generated connections displays complex non-linear spatio-temporal combinations of the task variables in the activity of the recurrent neurons. The temporal aspect is revealed in the changing patterns of mixed selectivity with the epochs of the task. In addition, the nature of reservoir activity is necessarily spatio-temporal and spans almost a full trial as memory of the previous choice is required to choose correctly at any given trial.

Because the activity of dACC neurons recorded in the same task also displayed dynamic mixed selectivity of task variables, we argue that the common property between the model and the cortex, namely their recurrence, is at the origin of the rich combinations found in both systems. One corollary ensues from this reasoning. One should observe dynamic mixed selectivity in every cortical area of the brain before any learning takes place.

In accordance with this principle, Nikolić et al. (2009) found dynamic representations of visual stimuli in the primary visual (V1) cortex of cats. A population of V1 neurons displayed fading memory of characters presented in a sequence, i.e. characters were encoded in spatio-temporal representations. The authors demonstrated that the identity of characters can be extracted using a linear decoder that acts as a cortical readout neuron with static weights.

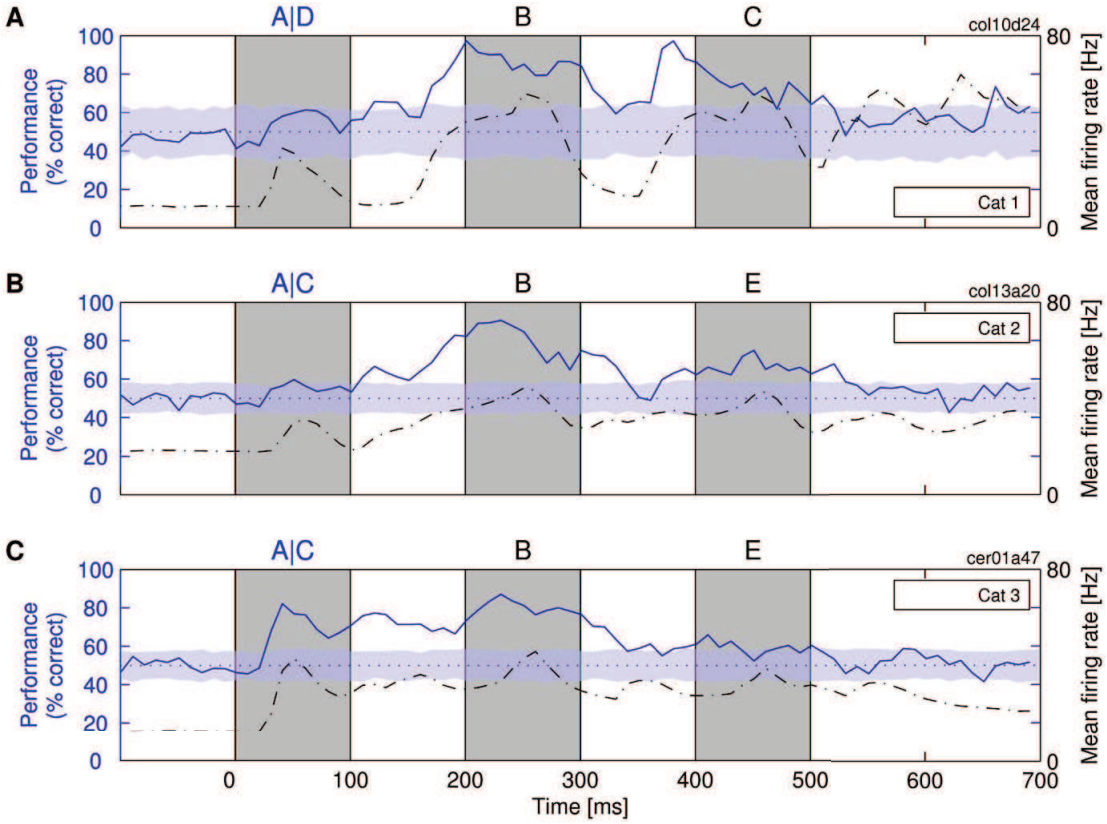

Figure 8.1: spatio-temporal encoding in V1 activity (Figure from Nikolić et al. (2009)). *The activity of V1 neurons of cats was recorded while sequences of characters were presented in the visual field of these neurons. Each of the three panels shows the decoding accuracy (blue line) of a linear decoder trained to extract the identity of the first character presented in each sequence (blue characters). The blue shaded area corresponds to the decoding accuracy that is not statistically above chance. The dash-dotted line represents the mean firing rate of the population of neurons (right scale). In each panel, the decoding of the first character (among two possible characters) is still possible even when the second and third character are presented. Consequently, representation of current visual inputs in a population of V1 neurons is influenced by previous inputs, a sign of spatio-temporal processing. In addition, a linear decoder representing a cortical neuron can continuously extract inputs represented in dynamic activity.*

Theses results first imply that a form of dynamic mixed-selectivity is present in the activity of single V1 neurons without any learning from the animal. In addition,

they show that a simple linear readout with static weights can continuously decode the identity of visual elements, meaning that a downstream neuron could similarly extract this information. Likewise, we found in our own experiment that a simple regularized linear regression with fixed weights is able to extract continuously the endogenous task variable related to the exploration behavior (search or repeat) of the model, and of the monkey. Thus, these results confirm in the PFC what as been found in the primary visual cortex.

The study of Meyers et al. (2012) presents results that apparently contradict our hypothesis concerning the presence of mixed selectivity in the PFC prior learning. The purpose of their experiments was to show the incorporation of new information in the activity of PFC after monkeys learned to perform match-to-sample tasks. One of the task depended the matching contingency between the identity of two stimuli, while the other involved the position of two identical stimuli. Prior training, single neural activity was first recorded while monkeys passively viewed two stimuli separated by a delay. The activity was also recorded after training when monkeys were performing correctly the match to sample tasks.

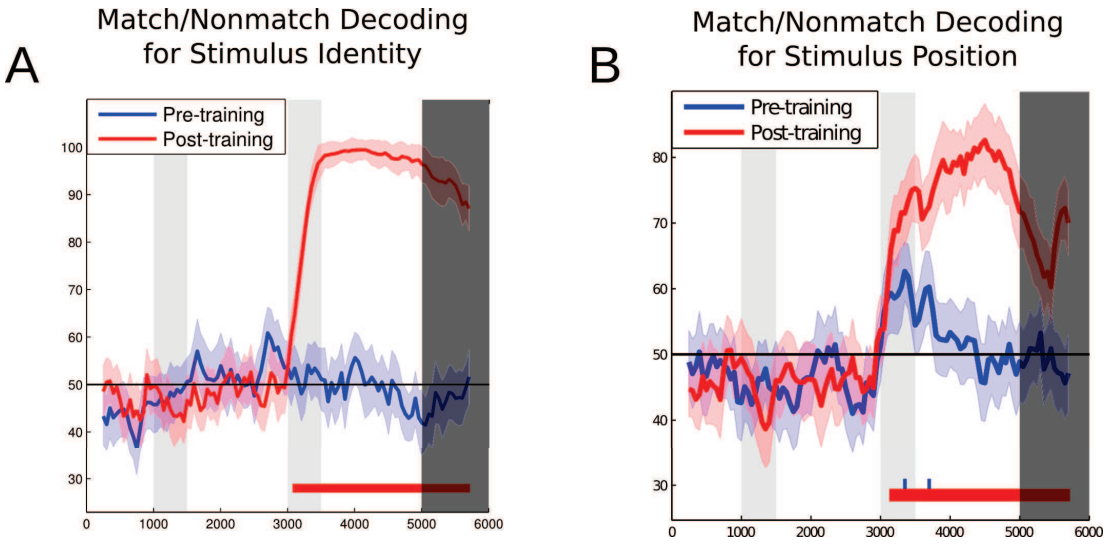

Figure 8.2: Match/non-match decoding before and after training in a match-to-sample task (Figures from Meyers et al. (2012)). *Prefrontal cortex neurons were recorded prior and after learning in monkeys that performed two match-to-sample tasks. Graphs illustrate the percentage of correct decoding of the match/nonmatch trial status pretraining (blue) and posttraining (red) for stimulus identity* ***A*** *and position* ***B*** *tasks. The gray shaded regions indicate the times when the first, second, and decision stimuli were shown, the black horizontal line indicates the level of decoding expected by chance, the color shaded regions indicate 1 SE in the decoding accuracy if different neurons were used, and the red and blue bars at the bottom of the figure indicate times when the decoding accuracy was above chance (permutation test,* $P < 0.005$*).*

To determine if task relevant information was incorporated after training, the authors decoded the match/nonmatch status of trial, i.e. whether the second stimulus was identical to, or, at the same position as the first one. In the position task, before training, results show that this contingency can be decoded better than chance just during and after the presentation of the second stimulus (Fig. 8.2B). However permutation tests show that this result is not significant. Learning the task seems to have strengthened the representation of this relevant information as shown by the high decoding accuracy with post-training activities. Conversely, in the stimuli identity task, the match/nonmatch status cannot be decoded before learning, but in this case decoding accuracy is not above chance (Fig. 8.2A). Yet, the authors demonstrate that stimuli position and identity can be decoded prior learning, hence all the necessary information to determine the matching contingency are present in the activity of PFC. According to our hypothesis, the match/nonmatch information should be represented prior learning in dynamic mixed selectivity which should allow decoding of this variable. The author show that this is not the case.

However, we believe that the maximum correlation coefficient method used to decode task variables in this study is poorly adapted to extract weakly represented variables. Our own experiments with cross-temporal pattern analysis (CPTA) and continuous linear decoding of dynamical activity revealed that weakly correlated population activities can nonetheless represent robustly a task variable. Indeed, exploring the dynamic nature of phase (the behavioral context) representation in the activity of dACC showed that this representation in the whole population activity was radically different between distinct periods of a trial, because the correlation is almost null. Yet, the linear decoder had a very high accuracy at separating the two behavioral contexts throughout full trials, meaning that context information is indeed present in the population activity. Consequently, we believe that the maximum correlation coefficient method used by Meyers et al., which relies on correlations between activities, is not suited to extract weakly represented information. The question of the contrast between CTPA and linear decoding is further explored in the section 8.2.3 Dynamics of Context Representation.

8.2.2 Explicit Context Representation

In the second article, explicit representation of the context (search or repeat) with a continuously active neuron was fed back into the recurrent network. We demonstrated that this allows the model to perform the task with less than half the number of neurons otherwise necessary. The main goal of the context neuron was to extend the influence of the task feedback (presence or absence of the reward) on the subsequent choice, and was therefore used as a temporal bridge between the feedback from one trial and the choice of

the subsequent trial. Indeed, the reward activation was a short impulse that propagated in the activity of the recurrent network. But, because of the time constant of the network dynamics, the brevity of this input could not sufficiently affect the dynamics of the network, as opposed to the choice readout that was activated for a longer period. In addition to create this temporal bridge, the context neuron robustly expressed a crucial information to perform this task, and thanks to its feedback connections, it contributed to refine the representation of the contingencies of the task. Let us confront this result with our theory.

According to our hypothesis, if a given contingency or context can be defined as spatio-temporal combinations of sensory, motor and internal variables, then a recurrent network with random connections fed with these variables will inherently represent it. In fact, in our case, the behavioral context (search vs repeat) is explained solely by the feedback variable. Instead of strengthening a combination of task variables in the spatial-temporal domain, we strengthened a contextual information in the temporal domain. Nevertheless, it is part of the retention process that allows for new combinations in the activity of the network. More precisely, in our case, it expressed explicitly a past input to strengthen the representation of contingencies, which are the combinations of previous choice and previous feedback inputs. With 300 neurons in the reservoir, the model could not perform the task perfectly without the context neuron, which means that these contingencies were not represented in the reservoir. However, with context feedback, the model made almost no errors. Hence, the temporal bridge created with explicit representation of the context allowed for the emergence of more robust contingency representations.

8.2.3 Dynamics of Context Representation

The mechanism used to maintain this contextual information through time is similar to the attracting dynamics of some of the models presented in the state of the art section (Wang, 2001; Rigotti et al., 2010a; Mante et al., 2013). However, our model is closer to the approach of Pascanu and Jaeger (2011) and Maass et al. (2007) that uses attracting dynamics to maintain information while processing is still carried out through transient dynamics. This dynamical regime is obtained with high dimensional attractors. In other words, some dimensions of the activity can segregate the state of a variable encoded in attractors, while the remaining dimensions are still available for spatio-temporal representations. However, these representations now rely on combinations that include the state of this variable. In this dynamical regime, activity of the population can still be highly dynamic while a few dimensions of the activity are engaged in attracting dynamics. In our case, the trajectories of search and repeat are very well separated, in the first

dimensions found with principal component analysis (PCA). This is due to the sharp input from the context neurons that strongly influenced the reservoir activity.

Conversely, it appears that the representation of the behavioral context in the dACC was similarly strengthened, as the activity of numerous single neurons clearly differentiate between search and repetition. PCA indicates that the trajectories of each behavioral context were separated in the state space, this suggests that dACC may also use attracting dynamics to robustly represent context. The attractors do not seem as well separated as in the model, because only a small part of the population activity variance separates well the trajectories.

However, continuous decoding of context indicates underlying attracting dynamics. Indeed, a single linear decoder with static weights could continuously extract the context from the population of dACC neurons throughout full trials. If the same linear decoder can correctly segregate all points of the trajectories, these trajectories must lie in different regions of the state space, hence suggesting an attracting dynamics (Durstewitz et al., 2010). However, the cross-temporal pattern analysis (CTPA) revealed that the representation of context in the whole population is rather dynamic. Indeed, the different contexts are best separated in the third component of the PCA which explains only 9% of the activity variance. This low percentage of variance implies that the attractors underlying the representation of context are only a small part of the whole dynamic of the population, which is consistent with the proposition of high dimensional attractors. In fact, we observed in the model that the strength of feedback had a strong impact on the separation of context in the trajectories (results not shown). In the results presented in the paper, trajectories of each context are separated in the first principal component. However, when decreasing the feedback weights of the context neuron, the separation of context was transferred to components of the PCA with lower explained variance.

Together, these results suggest that high dimensional attracting dynamics observed in reservoirs with feedback could explain the relatively dynamic but reliable representation of context in the dACC. However, the degree of dynamicity of this representation was not uniform throughout full trials. Next section proposes an interpretation of this phenomenon based on the literature.

8.2.4 Transient Dynamics with Successive Attractors

Both model and dACC populations showed cyclic trajectories, each representing one trial. This implies that the population activity goes through the same regions of the state space during each trial. The model's activity appears uniformly dynamic (with a constant multidimensional speed) compared to the dACC's dynamics that seem to fluctuate. The multidimensional speed of population trajectories (results not included

in the article), along with PCA and cross-temporal pattern analysis suggest that dACC dynamics may best be explained by the alternation between rather stable activities and dynamic ones.

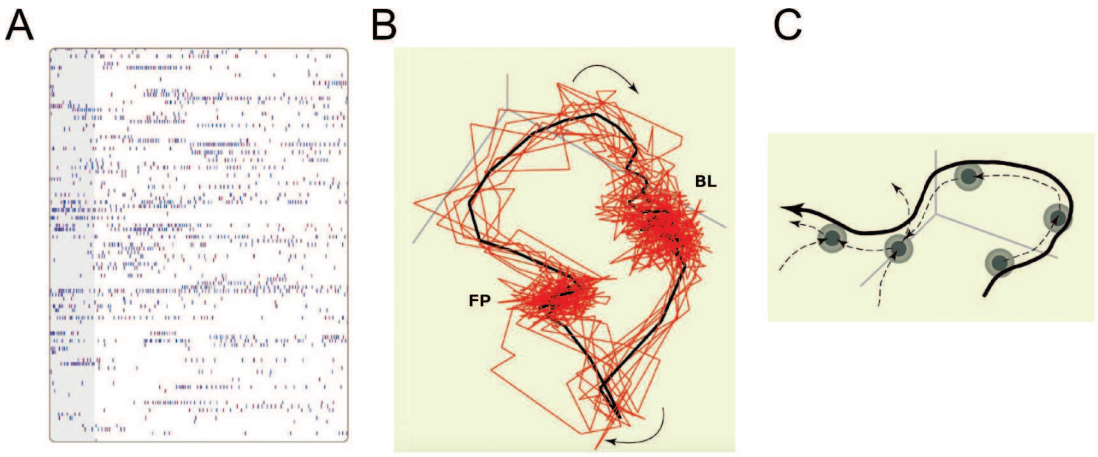

Figure 8.3: Transient dynamics for neural processing (Figure and caption from Rabinovich et al. (2008)). *A model of how neural networks in the locust antennal lobe process information.* ***A****. Single-trial responses of 110 locust antennal-lobe principal neurons to one odor can be recorded (gray bar, 1 s).* ***B****. Projections of principal-neuron trajectories, representing the succession of states visited by this neural network in response to one odor. Red lines, individual trials; black line, average of 10 trials. BL, baseline state; FP, fixed point (attractor), reached after 1.5 s.* ***C****. Putative dynamical model of transients: a set of dissipative saddles (semi-stable states, dark circles), sequentially connected by unstable separatrices (dashed lines). A single trajectory (continuous line) connects the neighborhoods of saddles in a heteroclinic channel.*

Recent studies strongly suggest that attractors represent the content of cognition (Colgin et al., 2010; Hyman et al., 2012; Durstewitz et al., 2010, 2000), and that cognitive processing is explained by the succession of attractors corresponding to mental states (O'Reilly, 2006; Rabinovich et al., 2008; Balaguer-Ballester et al., 2011). Based on the dynamics observed during odor processing, Rabinovich et al. (2008) develop the theory of *heteroclinic channels* referring to the dynamic activities of neural population that hop between semi-stable states interleaved by transient dynamics (Fig. 8.3). Likewise, in the population activity of ACC neurons, Balaguer-Ballester et al. (2011) found that population dynamics converge towards successive temporary stable states specific to each epoch of a task. These findings may participate in reconciling the cognitivist approach of mental states with the connectionist paradigm. Interestingly, another study involving PFC dynamics echoes these findings (Stokes et al., 2013). Neurons were recorded in the PFC of monkeys that were trained to perform a delayed paired-associate recognition task. Animals were first trained to associates 3 cues with 3 targets, then, in the delay

task, they were presented with a cue and a target interleaved with a delay. They had to activate a lever after a second delay if the target was associated with the cue during the training. A dynamic regime is found just after the presentation of the cue and of the target, while the first and second delay display stable regimes. Transient dynamics seem to be linked to the processing of the stimuli while stable regimes of activity generally associated with working memory may represent specific cognitive states.

While transient dynamics seem to explain the processing of incoming inputs, attractors may maintain cognitive states and participate in the integration of specific features of inputs (Mante et al., 2013). We will propose in the following sections a new model that would combine the transient and attracting approaches.

8.2.5 A Simple Mechanism to Learn Cognitive States and Context

In the two previous sections, we proposed that context is represented in high dimensional attractors, while successive semi-stable attractors represent the cognitive states of a task. Interestingly, the combination of these dynamics can be explained by the results of a model developed by Rigotti et al. (2010a) (which we reviewed in chap. 4). Indeed, they demonstrate that Hebbian learning allows for the formation of attractors that represent the successive cognitive states of the simulated task, which echoes the results presented in the previous section. These attractors gradually merge to form two attractors, each representing a distinct context. These results suggest that a very simple learning rule could potentially produce the attracting dynamics representing cognitive states and context that we observe in cortical activity.

8.3 Perspectives

Although the current version of our model has demonstrated the computational power of recurrent networks with fixed random connections in a cognitive task, further modelling is necessary to explore more biologically plausible architectures, dynamics and learning methods. The first section suggests to modify the learning of the current version of our model, while the second proposes to extend with more sophisticated attracting dynamics.

8.3.1 Towards a More Realistic Learning Method

Concerning our hypothesis, a major drawback of our model is that it does not model the selection of representations since it focuses on the representational power of recurrent

networks. The implemented learning method (FORCE learning) was primarily intended for motor learning and has low biological plausibility (see (Hoerzer et al., 2014)).

The cognitive task performed by our model was originally designed to understand the rapid alternation between search and repeat behaviors, and has been successfully modeled with the reinforcement learning (RL) framework (Khamassi et al., 2011, 2013). A logical follow up would be the training of our network model with RL to modify the weights between the reservoir and the readout. The output would implement a winner-take-all (WTA) mechanism to choose among the four possible targets, and feed back this information in the recurrent network. Reservoirs and RL have already met in the work of several teams.

Cognitive neural network modelling by Dominey et al. (1995) demonstrated how a cortico-striatal model with a form RL could process and produce sequences. Connections between the PFC (fixed recurrent network) and the striatum were modified with the sign of the prediction error. Interestingly, it was the first model which proposed that reward-related dopamine would underlie cortico-striatal plasticity.

In computational neuroscience, Hoerzer et al. (2014) developed a learning method similar to FORCE learning, but with Hebbian learning modulated by RL, which are more biologically plausible. They demonstrated the powerful computations that can be carried out with reservoirs endowed with this learning mechanism. Such learning rules could bring our model closer to biology and potentially lead to new insights in the representation of task variables in the PFC.

8.3.2 Bridging Attracting and Transient Approaches

While our model, and others in the transient dynamic community, may explain how powerful spatio-temporal processing can be carried out in simple recurrent network, the attracting dynamics have shown their importance for numerous cognitive functions, like working memory (Wang, 2001), decision making (**?**), and contextual processing (Mante et al., 2013). Attractors have the advantage of maintain robustly information (resistance to distractors), carry out specific computations (selective integration of information) and can span much longer periods than the fading memory of transient dynamics (WM and context representation).

Because of the need for attracting dynamics to explain cognition, and because of the inherent fading memory of reservoir networks, researchers in the transient dynamic community have integrated feedback mechanisms with attracting dynamics to their models in order to increase the memory capacity of the network and process inputs depending on context Pascanu and Jaeger (2011); Maass et al. (2007); Hoerzer et al. (2014). Although these models are able to reproduce the functional properties of attracting networks, they

do not seem to capture the complexity of the learning mechanisms that lead to complex adaptive behaviors. In this respect, we remind the reader of the work from Rigotti et al. (2010a) that proposes a very interesting mechanism to explain the emergence of context representation.

We speculate that the future of cortical modelling is in bridging the computational power of transient modelling approaches with the cognitive power of attracting dynamics. While the transient approach can be seen as a bottom-up or *analytical* approach that analyzes the capacity of generic cortical microcircuits, the attractor community has had a rather top-down or *constructive* approach in which circuits are designed to perform a specific function (Maass et al., 2007). We can see in the recent literature that both approaches have made steps toward hybrid dynamics. Indeed, the reservoir community has introduced attractors to augment the computational power of temporal recurrent networks while the cognitive attractor community has shown an interest in reservoir computing to explain the power of randomly generated networks.

We believe that the key to understanding the processing power of cortical circuits lies in specific learning methods that would allow generic networks to specialize to provide particular cognitive functions observed in specifically designed models. Indeed, it was pointed out that the same type of attractor network is able to carry out different cognitive functions (working memory and decision making) Wang (2013), in line with the description of other researchers that believe that the canonical cortical circuits implement winner-take-all through competitive dynamics (Douglas and Martin, 2004). The same cortical substrate may allow for the variety of cognitive processes observed in the different areas of the cortex. An explanation for the specific processing capabilities of each area may lie in the specialization of this cortical area for a particular dynamics through the right combination of transient and attracting dynamics.

One of the possible implementation would include a reservoir-style recurrent network connected to the inputs, and an attracting recurrent network endowed with learning that would act as a readout of the reservoir. This reservoir layer would expand the inputs in the spatio-temporal domain, while the attractor layer would develop the adapted cognitive function and express relevant information for the task at hand. The key to such a network would be the implementation of a biologically realistic learning that would allow the emergence of cognitive-related functions in the attractor network. The iterative approach of Rigotti et al. (2010a) (mentioned above) is a potential candidate. The following paragraph is dedicated to a possible biological substrate to such a system.

The majority of afferent connections to a cortical column are in layer IV, which may act as a form of reservoir, whereas the popular models of working memory and decision making cited above involve competitive dynamics with pyramidal neurons of layer II/III.

The transient dynamics of a reservoir (layer IV) must remain relatively stable, while the attracting dynamics in layers (layers II/III) could form through reward based learning with dopamine-mediated synaptic modifications. Interestingly, dopaminergic afferents from the midbrain project to all layers but layer IV (Berger et al., 1991). We let the reader appreciate the value of such speculations.

It has been more than 70 years since the first mathematical model from McCulloch and Pitts was introduced in science. From the early neural networks of this epoch to the current highly detailed models, the connectionist approach has uncovered fundamental mechanisms that slowly bridged the cognitive and neuroscience levels of explanation. This explanatory gap may be closed by merging the models of current constructive and analytical approaches.

8.4 Conclusion (Français)

Les dynamique transientes apparaissent comme un mécanisme dynamique puissant pour représenter l'information, et peuvent être implémentées par une architecture simple. Parce que les neurones du cortex cingulaire antérieur dorsal affichent des combinaisons non-linéaires complexes et dynamiques qui font écho à celles trouvées dans un réseaux de neurones avec des connections récurrentes aléatoires, nous soutenons que la puissance de traitement spatio-temporel du cortex est due en partie à la forte récurrence de sa connectivité locale. Cette capacité pour servir ce que nous dénommons le processus de variation qui permet au cortex de représenter des contingences arbitraires à travers la combinaisons d'informations sensorielles, motrices et internes disponibles. Les processus de sélection et de rétention pourrait en partie prendre la forme de la créations d'attracteurs pour maintenir et représenter des contingences de manière robuste. Des contingences représentées de manière explicite et retournées dans le réseau pourrait participer au développement de nouvelles combinaisons qui pourrait augmenter la puissance de représentation du cortex.

8.5 Conclusion (English)

Transient dynamics appear to be a powerful dynamical mechanism to represent information, and can be implemented with a simple architecture. Because dACC neurons possess complex dynamic non-linear combinations that echo those found in a neural network with random recurrent connections, we argue that the spatio-temporal processing power of the cortex is in part due to its highly recurrent local connectivity. This capacity may subserve what we refer to as the variation process that enables the cortex to rep-

resent arbitrary contingencies through the combination of available sensory, motor and internal information. The selection and retention processes may in part take the form of the creation of attractors to maintain and represent robustly contingencies. Explicitly represented contingencies fed back in the network would participate in developing new combinations which would enhance the representational power of the cortex.

Bibliography

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169.

Amit, D. J. (1989). *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press, New York, cambridge edition.

Amit, D. J. (1995). The Hebbian paradigm reintegrated: Local reverberations as internal representations. *Behavioral and brain siences*, 18(4):617–657.

Asaad, W. F., Rainer, G., and Miller, E. K. (1998). Neural activity in the primate prefrontal cortex during associative learning. *Neuron*, 21(6):1399–407.

Atiya, A. F. and Parlos, A. G. (2000). New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 11(3):697–709.

Baddeley, A. (2003). Working memory: looking back and looking forward. *Nature Reviews Neuroscience*, 4(10):829–39.

Balaguer-Ballester, E., Lapish, C. C., Seamans, J. K., and Durstewitz, D. (2011). Attracting dynamics of frontal cortex ensembles during memory-guided decision-making. *PLoS computational biology*, 7(5):e1002057.

Barbas, H. (1999). Prefrontal Cortex: Structure and Anatomy.

Barone, P. and Joseph, J. (1989). Prefrontal cortex and spatial sequencing in macaque monkey. *Experimental Brain Research*, 78(3):447–464.

Barto, A. (1998). *Reinforcement learning: An introduction*, volume 9. MIT Press.

Bathellier, B., Buhl, D. L., Accolla, R., and Carleton, A. (2008). Dynamic Ensemble Odor Coding in the Mammalian Olfactory Bulb: Sensory Information at Different Timescales. *Neuron*, 57(4):586–598.

Bechtel, W. and Abrahamsen, A. (1991). *Connectionism and the Mind: An Introduction to Parallel Processing in Networks*, volume 33. Blackwell, Cambridge MA.

Berger, B., Gaspar, P., and Verney, C. (1991). Dopaminergic innervation of the cerebral cortex: unexpected differences between rodents and primates. *Trends in neurosciences*, 14(1):21–7.

Bernacchia, A., Seo, H., Lee, D., and Wang, X.-J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nature neuroscience*, 14(3):366–72.

Blanc, J.-M. and Dominey, P. F. (2003). Identification of prosodic attitudes by a temporal recurrent network. *Brain research. Cognitive brain research*, 17(3):693–9.

Brunel, N. (1996). Hebbian Learning of Context in Recurrent Neural Networks. *Neural Computation*, 8(8):1677–1710.

Brunel, N. and Wang, X.-J. (2001). Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. *Journal of computational neuroscience*, 11(1):63–85.

Buonomano, D. V. and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–25.

Buonomano, D. V. and Merzenich, M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, 267(5200):1028–1030.

Carew, T., Walters, E., and Kandel, E. (1981). Classical conditioning in a simple withdrawal reflex in Aplysia californica. *The Journal of Neuroscience*, 1(12):1426–1437.

Clower, W. T. and Alexander, G. E. (1998). Movement sequence-related activity reflecting numerical order of components in supplementary and presupplementary motor areas. *Journal of neurophysiology*, 80(3):1562–6.

Cohen, J. D. and Servan-Schreiber, D. (1992). Context, cortex, and dopamine: a connectionist approach to behavior and biology in schizophrenia. *Psychological review*, 99(1):45–77.

Colgin, L. L., Leutgeb, S., Jezek, K., Leutgeb, J. K., Moser, E. I., McNaughton, B. L., and Moser, M.-B. (2010). Attractor-map versus autoassociation based attractor dynamics in the hippocampal network. *Journal of neurophysiology*, 104(1):35–50.

Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X.-J. (2000). Synaptic Mechanisms and Network Dynamics Underlying Spatial Working Memory in a Cortical Network Model. *Cerebral Cortex*, 10(9):910–923.

Dennett, D. C. (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life*, volume 48. Simon & Schuster, New York.

Dominey, P. F. (1995). Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, 73(3):265–274.

Dominey, P. F., Arbib, M., and Joseph, J. (1995). A model of corticostriatal plasticity for learning oculomotor associations and sequences. *Journal of Cognitive Neuroscience*, 7(3):311–336.

Dominey, P. F. and Ramus, F. (2000). Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant. *Language and Cognitive Processes*, 15(1):87–127.

Donahoe, J. W. (1999). Edward L. Thorndike: The selectionist connectionist. *Journal of the experimental analysis of behavior*, 3(3):451–454.

Douglas, R. J. and Martin, K. a. C. (2004). Neuronal circuits of the neocortex. *Annual review of neuroscience*, 27:419–51.

Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, volume 6, pages 2777–2780. IEEE.

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks*, 12(7-8):961–974.

Durstewitz, D. and Deco, G. (2008). Computational significance of transient dynamics in cortical networks. *The European journal of neuroscience*, 27(1):217–27.

Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nature neuroscience*, 3 Suppl(november):1184–91.

Durstewitz, D., Vittoz, N. M., Floresco, S. B., and Seamans, J. K. (2010). Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. *Neuron*, 66(3):438–48.

Edelman, G. M. (1987). *Neural Darwinism: The theory of neuronal group selection.* Basic Books.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):195–225.

Funahashi, K.-i. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806.

Funahashi, S. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neuroscience*, 6(2):331–349.

Funahashi, S. (2001). Neuronal mechanisms of executive control by the prefrontal cortex. *Neuroscience research*, 39(2):147–65.

Funahashi, S., Inoue, M., and Kubota, K. (1997). Delay-period activity in the primate prefrontal cortex encoding multiple spatial positions and their order of presentation. *Behavioural Brain Research*, 84(1-2):203–223.

Fuster, J. M. (1973). Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory. *Journal of Neurophysiology*, 36(1):61–78.

Fuster, J. M. and Alexander, G. E. (1971). Neuron Activity Related to Short-Term Memory. *Science*, 173(3997):652–654.

Genovesio, A., Brasted, P. J., Mitz, A. R., and Wise, S. P. (2005). Prefrontal cortex activity related to abstract response strategies. *Neuron*, 47(2):307–20.

Goldman-Rakic, P. S. (1995). Cellular basis of working memory. *Neuron*, 14(3):477–485.

Grant, D. A. and Berg, E. (1948). A behavioral analysis of degree of reinforcement and ease of shifting to new responses in a Weigl-type card-sorting problem. *Journal of Experimental Psychology*, 38(4):404–411.

Griniasty, M., Tsodyks, M. V., and Amit, D. J. (1993). Conversion of Temporal Correlations Between Stimuli to Spatial Correlations Between Attractors. *Neural Computation*, 5(1):1–17.

Harlow, J. M. (1999). Passage of an iron rod through the head. 1848. *Journal of neuropsychiatry and clinical neurosciences*, 11(2):281–283.

Hasegawa, R., Sawaguchi, T., and Kubota, K. (1998). Monkey prefrontal neuronal activity coding the forthcoming saccade in an oculomotor delayed matching-to-sample task. *Journal of neurophysiology*, 79(1):322–33.

Haykin, S. O. (1999). *Neural networks: A Comprehensive Foundation*, volume 8. Prentice-Hall Inc., Upper Saddle River, NJ.

Hebb, D. O. (1949). *The organization of behavior: A Neuropsychological Approach*, volume 63. John Wiley & Sons, New York.

Hermans, M. and Schrauwen, B. (2012). Recurrent kernel machines: computing with infinite echo state networks. *Neural Computation*, 24(1):104–33.

Hinaut, X. and Dominey, P. F. (2013). Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. *PloS one*, 8(2):e52946.

Hinton, G. E. and Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. E., McClelland, J. L., and the PDP research group, editors, *Parallel Distributed Processing*, chapter 7, pages 282–317. MIT Press, Cambridge, MA.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544.

Hoerzer, G. M., Legenstein, R., and Maass, W. (2014). Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral cortex*, 24(3):677–90.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(April):2554–2558.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Hull, C. L. (1943). *Principles of behavior: an introduction to behavior theory.* Appleton-Century-Crofts, New York.

Hyman, J. M., Ma, L., Balaguer-Ballester, E., Durstewitz, D., and Seamans, J. K. (2012). Contextual encoding by ensembles of medial prefrontal cortex neurons. *Proceedings of the National Academy of Sciences*, 109(13):5086–5091.

Jaeger, H. (2001). The" echo state" approach to analysing and training recurrent neural networks. Technical report, Technical Report GMD Report 148, German National Research Center for Information Technology.

Jaeger, H. (2002). Adaptive Nonlinear System Identification with Echo State Networks. In *Advances in neural information processing systems*, volume 4, pages 593–600.

Jaeger, H. (2005). A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. *Frauenhofer Institue for Autonomous Intelligent*, 2002:1–46.

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80.

Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007a). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–52.

Jaeger, H., Maass, W., and Principe, J. (2007b). Special issue on echo state networks and liquid state machines. *Neural Networks*, 20(3):287–289.

James, W. (1890). *The principles of psychology*, volume 1. Dover, New York.

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach (Tech. Rep. No. 8604).

Kable, J. W. and Glimcher, P. W. (2009). The neurobiology of decision: consensus and controversy. *Neuron*, 63(6):733–45.

Khamassi, M., Enel, P., Dominey, P. F., and Procyk, E. (2013). Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters. *Progress in brain research*, 202:441–64.

Khamassi, M., Lallée, S., Enel, P., Procyk, E., and Dominey, P. F. (2011). Robot cognitive control with a neurophysiologically inspired reinforcement learning model. *Frontiers in neurorobotics*, 5(July):1.

Kim, J. N. and Shadlen, M. N. (1999). Neural correlates of a decision in the dorsolateral prefrontal cortex of the macaque. *Nature neuroscience*, 2(2):176–85.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kuhl, P., Williams, K., Lacerda, F., Stevens, K., and Lindblom, B. (1992). Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, 255(5044):606–608.

Langton, C. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37.

Lashley, K. (1950). In search of the engram. *Symposia of the Society for Experimental Biology*, 4:454–482.

Le Cun, Y. (1986). Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*, pages 233–240. Springer, Berlin.

Legenstein, R. and Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural networks : the official journal of the International Neural Network Society*, 20(3):323–34.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Lukoševičius, M., Jaeger, H., and Schrauwen, B. (2012). Reservoir Computing Trends. *KI - Künstliche Intelligenz*, 26(4):365–371.

Luria, A. R. (1966). Higher cortical functions in man.

Ma, S. and Ji, C. (1998). Fast training of recurrent networks based on the EM algorithm. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 9(1):11–26.

Maass, W. (2011). Liquid State Machines: Motivation, Theory, and Applications. In Sorbi, A. and Cooper, B., editors, *Computability in Context: Computation and Logic in the Real World*, chapter 1, pages 275–296. Imperial College Press.

Maass, W., Joshi, P., and Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS computational biology*, 3(1):e165.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–60.

Maass, W., Natschläger, T., and Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal of physiology, Paris*, 98(4-6):315–30.

Mansouri, F. A., Matsumoto, K., and Tanaka, K. (2006). Prefrontal cell activities related to monkeys' success and failure in adapting to rule changes in a Wisconsin Card Sorting Test analog. *The Journal of Neuroscience*, 26(10):2745–56.

Mante, V., Sussillo, D. C., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84.

Martens, J. and Sutskever, I. (2011). Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040.

Mazor, O. and Laurent, G. (2005). Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron*, 48(4):661–73.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5:115–133.

Medler, D. (1998). A brief history of connectionism. *Neural Computing Surveys*, 1:61–101.

Meyers, E. M., Qi, X.-L., and Constantinidis, C. (2012). Incorporation of new information into prefrontal cortical activity after learning working memory tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 109(12):4651–6.

Miller, E. K. and Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24:167–202.

Miller, E. K., Erickson, C. a., and Desimone, R. (1996). Neural mechanisms of visual working memory in prefrontal cortex of the macaque. *The Journal of Neuroscience*, 16(16):5154–67.

Miller, G. A., Galanter, E., and Pribram, K. H. (1968). Plans and the Structure of Behavior. *Journal of the Operational Research Society*, 19(3):338–340.

Milner, B. (1963). Effects of Different Brain Lesions on Card Sorting: The Role of the Frontal Lobes. *Archives of Neurology*, 9(1):90–100.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*, volume 165.

Miyashita, Y. and Chang, H. S. (1988). Neuronal correlate of pictorial short-term memory in the primate temporal cortex. *Nature*, 331(6151):68–70.

Monsell, S. (2003). Task switching. *Trends in Cognitive Sciences*, 7(3):134–140.

Mozer, M. (1989). A focused back-propagation algorithm for temporal pattern recognition. *Complex systems*, 3(4):349–381.

Newell, A. and Simon, H. a. (1976). Computer science as empirical inquiry: symbols and search. *Communications of the ACM*, 19(3):113–126.

Newsome, W. and Pare, E. (1988). A selective impairment of motion perception following lesions of the middle temporal visual area (MT). *The Journal of Neuroscience*, 8(6):2201–2211.

Nikolić, D., Häusler, S., Singer, W., and Maass, W. (2009). Distributed fading memory for stimulus properties in the primary visual cortex. *PLoS biology*, 7(12):e1000260.

O'Reilly, R. C. (2006). Biologically based computational models of high-level cognition. *Science*, 314(5796):91–4.

Parker, D. B. (1982). Learning logic. Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University, Stanford, CA.

Pascanu, R. and Jaeger, H. (2011). A neurodynamical model for working memory. *Neural Networks*, 24(2):199–207.

Puskorius, G. V. and Feldkamp, L. A. (1994). Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 5(2):279–97.

Rabinovich, M., Huerta, R., and Laurent, G. (2008). Transient dynamics for neural processing. *Science*, 321(5885):48–50.

Rescorla, R. A. and Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., and Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–90.

Rigotti, M., Ben Dayan Rubin, D., Morrison, S. E., Salzman, C. D., and Fusi, S. (2010a). Attractor concretion as a mechanism for the formation of context representations. *NeuroImage*, 52(3):833–47.

Rigotti, M., Ben Dayan Rubin, D., Wang, X.-J., and Fusi, S. (2010b). Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in computational neuroscience*, 4(October):24.

Robinson, A. J. and Fallside, F. (1987). *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering.

Romo, R., Brody, C. D., Hernández, a., and Lemus, L. (1999). Neuronal correlates of parametric working memory in the prefrontal cortex. *Nature*, 399(6735):470–3.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408.

Rumelhart, D. E., Hinton, G. E., and McClelland, J. L. (1986a). A general framework for parallel distributed processing. In *Parallel distributed processing*, chapter 2, pages 45–76. MIT Press, Cambridge MA.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8, pages 318–362.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986c). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Sakagami, M. and Niki, H. (1994). Encoding of behavioral significance of visual stimuli by primate prefrontal neurons: relation to relevant task conditions. *Experimental Brain Research*, 97(3):423–436.

Schrauwen, B., Verstraeten, D., and Van Campenhout, J. (2007). An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, pages 471–482.

Schrauwen, B., Wardermann, M., Verstraeten, D., Steil, J. J., and Stroobandt, D. (2008). Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71(7-9):1159–1171.

Shima, K., Isoda, M., Mushiake, H., and Tanji, J. (2007). Categorization of behavioural sequences in the prefrontal cortex. *Nature*, 445(7125):315–8.

Skinner, B. F. (1981). Selection by consequences. *Science*, 213(4507):501–4.

Spencer, H. (1855). *The principles of psychology*, volume 1. D. Appleton and Company, New York.

Steil, J. J. (2007). Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. *Neural Networks*, 20(3):353–64.

Stokes, M. G., Kusunoki, M., Sigala, N., Nili, H., Gaffan, D., and Duncan, J. (2013). Dynamic Coding for Cognitive Control in Prefrontal Cortex. *Neuron*, 78(2):1–12.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18(6):643–662.

Sussillo, D. C. and Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–57.

Thorndike, E. L. (1898). Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i–109.

Thorndike, E. L. (1932). *The fundamentals of learning.* Teachers College Bureau of Publications, Columbia University, New York, NY.

Verstraeten, D., Schrauwen, B., D'Haene, M., and Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403.

Verstraeten, D., Schrauwen, B., and Stroobandt, D. (2006). Reservoir-based techniques for speech recognition. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1050–1053. IEEE.

Von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In *Automata Studies*, pages 3–42. Princeton University Press, Princeton, NJ.

Wallis, J. D., Anderson, K. C., and Miller, E. K. (2001). Single neurons in prefrontal cortex encode abstract rules. *Nature*, 411(6840):953–6.

Wang, X.-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in neurosciences*, 24(8):455–63.

Wang, X.-J. (2002). Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36(5):955–68.

Wang, X.-J. (2008). Decision making in recurrent neuronal circuits. *Neuron*, 60(2):215–34.

Wang, X.-J. (2013). The prefrontal cortex as a quintessential "cognitivetype" neural circuit: working memory and decisionmaking. In Stuss, D. and Knight, R., editors, *Principles of Frontal Lobe Function*, chapter 15, pages 226–248. Cambridge University Press.

Watanabe, M. (1986). Prefrontal unit activity during delayed conditional Go/No-Go discrimination in the monkey. I. Relation to the stimulus. *Brain research*, 382(1):1–14.

Watanabe, M. (1992). Frontal units of the monkey coding the associative significance of visual and auditory stimuli. *Experimental Brain Research*, 89(2):233–47.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Master thesis, Harvard University.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356.

White, I. M. and Wise, S. P. (1999). Rule-dependent neuronal activity in the prefrontal cortex. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, 126(3):315–35.

Widrow, B. and Hoff, M. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*, (4):96 – 104.

Williams, R. J. and Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280.

Winograd, S. and Cowan, J. D. (1963). *Reliable computation in the presence of noise*. Number 22. MIT Press Cambridge, Cambridge, MA.

Zipser, D. and Andersen, R. A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331:679–684.