



HAL
open science

Neural Networks and Sparse Information Acquisition

Behrooz Kamary Aliabadi

► **To cite this version:**

Behrooz Kamary Aliabadi. Neural Networks and Sparse Information Acquisition. Information Theory [math.IT]. Télécom Bretagne; Université de Bretagne-Sud, 2013. English. NNT: . tel-00962603

HAL Id: tel-00962603

<https://theses.hal.science/tel-00962603>

Submitted on 21 Mar 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.

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En habilitation conjointe avec l'Université de Bretagne-Sud

École Doctorale – SICMA

Neural Networks and Sparse Information Acquisition

Thèse de Doctorat

Mention : « STIC (Science et technologies de l'information et de la communication) »

Présentée par **Behrooz Kamary Aliabadi**

Département : Electronique

Laboratoire : Lab – STICC Pôle : CACS

Soutenue le 26 juin 2013

Jury :

M. Joseph Boutros, Professeur des Universités, Texas A&M University at Qatar (Rapporteur)

M. Pierre Siohan, HDR, Orange Labs (Rapporteur)

Mme. Bernadette Dorizzi, Professeur des Universités, Télécom SudParis (Examinatrice)

M. Franck Vermet, Maître de conférences, Université de Bretagne Occidentale (Examinateur)

M. Philippe Coussy, Maître de conférences, Université de Bretagne-Sud (Examinateur)

M. Claude Berrou, Professeur, Télécom Bretagne (Directeur de thèse)

Résumé

Le principe de parcimonie, si fondamental dans les systèmes biologiques du fait de ressources limitées, peut être exploité avec profit dans la conception de systèmes efficaces de stockage de l'information. La parcimonie est une contrainte; malgré cela, les systèmes biologiques - le cerveau en premier lieu - sont complexes, puissants et flexibles. Le principe de parcimonie appliqué au traitement des signaux et des données peut donc conduire à des solutions efficaces dans l'acquisition et le stockage d'informations. Cette thèse traite de la conception d'un système neuro-inspiré dont une des qualités recherchées est la parcimonie. Le système développé et validé est un réseau de neurones binaire et récurrent, analysé en détail dans le deuxième chapitre.

Le néocortex est la couche la plus récemment développée du cerveau humain et est le centre des pensées conscientes et du langage. Cette partie du cerveau présente donc un grand intérêt, en tant que référence biologique, dans la recherche de nouvelles solutions de mémorisation. Les premier et deuxième chapitres du mémoire de thèse proposent une courte analyse des caractéristiques et des modèles de la mémoire humaine et constituent une introduction à un réseau de neurones de stockage original exploitant le principe de parcimonie.

L'organisation et les propriétés du néocortex ont conduit Gripon et Berrou à proposer un nouveau type de réseau de neurones [GB11]. Ce réseau est constitué de nœuds regroupés en différents sous-ensembles disjoints appelés "grappes". Il lui correspond donc un graphe multiparti dont les connexions sont purement bi-

naires et non pondérées. Cependant, ce modèle ne répond pas complètement à la forte contrainte de parcimonie que doit satisfaire le “codage neural”. Afin d’aller plus loin dans le sens de la parcimonie, le réseau proposé dans [GB11] doit être repensé et réévalué en ce qui concerne l’utilisation des ressources disponibles : les messages à y stocker ne doivent pas utiliser toutes les grappes du réseau mais seulement une partie d’entre elles. Le type de réseau proposé dans cette thèse est donc une extension de celui présenté dans [GB11] pour lui conférer de plus fortes caractéristiques de parcimonie.

Ce nouveau réseau offre une capacité de stockage importante et un fort pouvoir de correction de messages erronés ou partiellement effacés. La diversité, c’est-à-dire le nombre de messages pouvant être mémorisés, est proportionnelle au carré du nombre total de nœuds, alors que dans le modèle classique, elle est proportionnelle au carré du nombre de nœuds dans chaque grappe. Par ailleurs, il a été constaté et démontré que, pour une même diversité, le taux d’erreur dans la récupération de messages tronqués est considérablement plus bas avec le réseau que nous proposons qu’avec le réseau classique. En outre, nous pensons que la structure et les principes de mémorisation de ce nouveau réseau sont plus proches de ceux du néocortex humain. Dans celui-ci, les micro-colonnes sont regroupées en colonnes à leur tour réunies en macro-colonnes. Dans notre modèle ainsi que celui décrit dans [GB11], les nœuds matérialisent les micro-colonnes qui sont regroupées dans des grappes représentant les colonnes et finalement toutes les grappes forment le réseau représentatif d’une macro-colonne. La différence réside dans l’utilisation des grappes, bien plus parcimonieuse dans notre cas, et donc plus plausible biologiquement.

En plus de sa grande capacité de stockage ainsi que de sa forte aptitude à la correction, les algorithmes d’écriture et de récupération de messages sont simples. Ce modèle est un bon candidat pour expliquer les mécanismes et les principes de la mémoire cérébrale à long terme et aussi pour servir de point de départ dans

la conception de machines capables de stocker un nombre considérable de messages, souvenirs, situations, voire même de séquences. Nos travaux ont fait l'objet d'une soumission d'article à IEEE trans. on Neural Networks and Learning Systems [AGJ12].

Dans le troisième chapitre, le réseau est analysé sous l'éclairage de la théorie de l'information et du codage. Le réseau s'appuie en effet sur des lois de codage redondant qui lui confèrent des propriétés de codage correcteur d'erreurs ou d'effacements. Ainsi il peut restituer un mot de code (un message) stocké à partir d'une version erronée présentée à son entrée. Les mots de codes linéaires, qui sont communément utilisés dans les systèmes de communication, dépendent d'une matrice génératrice. Le réseau de neurones, quant à lui, résout le codage de canal différemment: les mots de code ne sont pas nécessairement linéaires. Le réseau est en effet capable de stocker une série de mots de codes (par exemple, un ensemble de messages indépendants et uniformément distribués) qui peuvent être ou ne pas être linéairement dépendants les uns des autres, ce qui laisse au concepteur un important degré de liberté dans le choix du transcodage (le "mapping" des informations et des noeuds du réseau). Le concepteur peut donc fixer les paramètres en fonction des exigences du système. Il existe deux principales différences entre le réseau classique et le réseau proposé dans cette thèse quand ils sont utilisés pour des applications de codage de canal. Dans le modèle classique, le nombre de mots de code possibles est très large en comparaison du nombre de messages stockés. Dans le nouveau réseau, la diversité est beaucoup plus importante et est donc plus proche de la cardinalité des codes linéaires. Le deuxième avantage du nouveau réseau est la variabilité de la longueur des messages stockés. Cette dernière propriété rend le système capable d'exploiter des codes à longueur variable (quand les mots de code sont de différentes longueurs). Le système pourrait également réaliser une fonction de codage à taux de codage variable selon la qualité des canaux de communication.

Shannon [Sha48] suggérait que le codage de source et le codage de canal soient considérés comme des opérations de traitement de l'information distincts et disjoints. Des recherches ultérieures ont montré qu'un codage conjoint source canal pouvait améliorer les performances. Le réseau considéré dans cette thèse met en œuvre un type de codage conjoint source canal qui est élaboré de la façon suivante. Les messages sont stockés dans des cliques qui sont des sous-graphes entièrement connectés par des arêtes binaires. Considérons un message avec c sous-messages significatifs (non-zéro) stockés dans un réseau de χ grappes et ℓ nœuds dans chaque grappe ($n = \chi\ell$ est le nombre total de nœuds). Le message est associé à une clique avec $\binom{c}{2} = c(c-1)/2$ connexions binaires. Le message contient $c \log_2 \ell$ bits. Ainsi, pour obtenir un effet de compression, l'inégalité $\frac{c-1}{2} < \log_2 \ell$ doit être respectée. Par ailleurs, une clique a une structure redondante, c'est-à-dire que les c nœuds appartenant à un message sont liés par $\frac{c(c-1)}{2}$ connexions alors que $\lfloor \frac{c+1}{2} \rfloor$ connexions peuvent être suffisantes pour spécifier tous les nœuds de la clique. Du fait de cette redondance, une clique peut être considérée comme un mot de code d'un code redondant. On voit donc que, sous certaines conditions, codage de source et codage de canal peuvent être combinés dans une même opération de mémorisation dans le réseau.

La principale contribution de ce chapitre est le calcul des contraintes sur les paramètres du système, comme par exemple le nombre maximum de messages stockés et le nombre maximum acceptable de sous-messages manquants dans un message d'entrée. Ces contraintes sont calculées à l'aide de la théorie de l'information. Pour cela, la mémoire du réseau de neurones réalisant le stockage et le processus d'extraction est modélisé comme un canal de communication bruyant. L'ensemble de messages à stocker provient de la source de l'information et peut ensuite être récupéré à partir du message partiels. L'information est mesurée depuis trois points: (a) la quantité d'information contenue dans la source, (b) la quantité d'information fournie au réseau depuis le message partiel, (c) la quan-

tité d'information obtenue après la récupération. Il est attendu que la quantité d'information récupérée soit plus élevée que la quantité d'information donnée par le message partiel. Le différentiel du taux d'information I_{diff} est défini comme la quantité d'information récupérée déduite de la quantité d'information donnée par le message partiel. Le réseau est ainsi appelé *informatif* quand le différentiel du taux d'information est positif. La discussion ci-dessus implique que le différentiel du taux d'information soit dépendant du nombre de sous-messages manquants ainsi que du nombre de messages stockés. En effet, le taux d'information des messages partiels est fonction de ces deux facteurs. Ceci conduit à un jeu de contraintes sur le réseau afin qu'il demeure informatif, c'est-à-dire, $I_{\text{diff}} > 0$. De plus, l'inégalité de Fano est aussi utilisée pour dériver les contraintes appliquées au nombre de sous-messages manquants.

Dans le quatrième chapitre, une nouvelle technique d'acquisition de messages parcimonieux, à savoir *compressed sensing*, est étudiée du point de vue de la théorie de l'information. Différentes techniques sont utilisées pour compresser et stocker les signaux parcimonieux. Ces différentes techniques comprennent entre autres, la compression *Sample-then-Compress Framework* (SCF) et la compression *Compressed-Sensing* (CS). La première est une compression bien connue, employée notamment dans le standard de compression d'image Joint Photographic Expert Group (JPEG). Dans la compression SCF, un dispositif échantillonne un signal (incluant des informations redondantes) pour ensuite, dans une étape de post-traitement, extraire les informations importantes. CS est un domaine émergent de compression et d'acquisition de signaux dans lequel l'échantillonnage et la compression conjoints de signaux parcimonieux vise à réduire la complexité des encodeurs. Dans ce schéma, l'échantillonnage et la compression ne sont pas des procédures distinctes (c-à-d, opérées en série), mais sont réalisées simultanément. La compression CS permet d'éviter l'acquisition et le traitement d'informations redondantes : le taux d'échantillonnage est proportionnel à la teneur en informa-

tions sous-jacentes (autrement dit le niveau de parcimonie) plutôt qu'à la bande passante.

La structure CS est très similaire à la *Bidirectional Associative Memories* (BAM) proposée par Kosko *et al.*, où un ensemble de paires de vecteurs de dimensions différentes est stocké dans un réseau à mémoire. Les systèmes de CS ainsi que leurs connexions avec les réseaux de neurones à mémoire et les techniques d'apprentissage automatique sont discutés dans ce chapitre. La principale contribution en est le calcul de conditions de récupération plus strictes que l'état de l'art sur ce genre de systèmes. Une condition de récupération originale est aussi obtenue pour des signaux approximativement parcimonieux.

Preface

Parsimony in biological neural systems such as the brain inspires the design of systems that may be more efficient. Indeed, parsimony in signals can be exploited for efficient sampling and storage. In this thesis the design of a biologically inspired system that employs parsimony of signals is the main subject of study. This system is a neural network memory which is presented in chapters two and three. At first, the parsimonious features of the human neocortex and the human memory are briefly investigated from psychological and biological aspects. These features have inspired a neural network memory introduced by Gripon and Berrou. An extension to this network is proposed in this thesis. It has more embedded sparsity features of the human brain in comparison with the classical network. It is also capable of storing sparse messages unlike this classical network. The proposed network is analysed and its advantages are brought to the fore. In chapter three, the proposed network is analysed from information and coding theory aspects. The network can be considered for error correction because it is an auto-associative memory and can therefore store a set of codewords (as messages) and recover them from distorted versions. The network implements a kind of joint source-channel coding which is elaborated in this part. The main contribution of the information-theoretic analysis is the computation of constraints on the system parameters, such as the maximum number of stored messages and the maximum number of acceptable missing sub-messages in a partial message.

In the fourth chapter, a new sparse signal acquisition technique namely com-

pressed sensing is studied, exploiting information theory. Compressed sensing systems and their connection to neural network memories and machine learning techniques are discussed. The main contribution of this chapter is the computation of tighter information-theoretic recovery conditions on these systems. A novel recovery condition is also derived for approximately sparse signals.

Acknowledgment

Foremost, I would like to express my gratitudes to my thesis director Prof. Claude Berrou for all his continuous help and support.

I am grateful to my parents and my brother for all their lifelong support and inspiration.

I also thank *Futur et Ruptures* programme of Institut Mines-Télécom that funded my Ph.D. studies.

Contents

Contents	11
List of Figures	15
List of Tables	18
1 Introduction	19
1.1 Memory	20
1.2 Compression and Robustness	22
1.2.1 Neural Coding	23
1.3 Artificial Neural Networks	23
1.4 Thesis structure	26
2 Sparse Neural Network Memories	29
2.1 Sparse Coding	30
2.2 Messages	32
2.3 Storage	35
2.4 Retrieval	46
2.4.1 Winner-take-all	47
2.4.2 The Algorithm	49
2.5 Performance	53
2.5.1 Associative Memory	55
Guided Recovery	56

	Blind Recovery	56
	Variable Order Messages	60
2.5.2	Blurred Messages	62
2.5.3	Set Implementation	64
2.6	Optimal Network	66
2.6.1	Optimal Constant Message Order	67
2.6.2	Optimal Density	68
2.6.3	Optimal Variable Message Order	69
3	Information-Theoretic Analysis of the Neural Network Memory	71
3.1	Analysis of the Message Source	72
3.2	Channel Coding	74
3.3	Source-Channel Coding	76
3.4	Communication Channel Model	78
3.4.1	Puncturing Block Channel Model	81
3.4.2	Neural Network Memory Channel Model	84
3.4.3	Recovery Constraints: Differential Information Rate	84
3.4.4	Recovery Constraints: Fano's Inequality	87
3.4.5	Discussion	88
4	Information-Theoretic Constraints on Sparse Pattern Recovery	91
4.1	Recovery	94
4.1.1	Restricted Isometry Property (RIP)	95
4.1.2	Null-Space Property (NSP)	96
4.1.3	Verification of the Algebraic Conditions	96
4.2	Compressed Sensing and Neural Networks	97
4.3	Information-Theoretic Conditions	99
4.3.1	Bounds and Recovery Conditions	101
	Strictly Sparse Signals	102

<i>CONTENTS</i>	13
Approximately Sparse Signals	105
4.3.2 Proof	107
Theorem 4	107
Corollary 1	112
Theorem 6	113
Corollary 2	114
5 Discussion and Conclusion	115
Bibliography	119

List of Figures

1.1	Block diagram representation of nervous system.	24
1.2	Diagram of a single-layer feedforward neural network.	25
1.3	Diagram of a single-layer recurrent neural network with only one feedback.	25
2.1	The neural network inspired by the human neocortex. The dots represent the microcolumns and the regions surrounded with dashed lines show the columns (clusters) and a macrocolumn (the network) in the neocortex.	32
2.2	A clique (a complete subgraph) that represents a message containing eight sub-messages in the network. This clique has twenty-eight connections. Although a single loop graph with only seven connection can associate the fanals, a fully connected graph provides robustness against deficiencies.	38
2.3	A network composed of 9 clusters of 16 fanals each. A message with message order 4 forms a clique in the network.	39
2.4	The network density for four values of c as a function of the number of stored messages M where $\chi = 100$ and $\ell = 64$	43
2.5	The network diversity ($\eta = 1$) is depicted for various values of ℓ when $\chi = 100$ and when the stored messages have a constant order $4 \leq c \leq \chi$	44

- 2.6 A fanal is modeled similar to McCulloch-Pitts model of neuron. It computes the sum of the received signals which is known as activity level. Thereafter, if it is equal to the maximum of the maximum activity level in the network and the threshold, it fires. 48
- 2.7 A bipartite graph that represents an LDPC code is depicted. The nodes on top are variable nodes and the nodes in the bottom are check nodes. The check nodes are arithmetic nodes (that do not take values) whereas the variable nodes hold the result of iterative computation that is performed in each iteration. 49
- 2.8 A situation is demonstrated where three active fanals in a cluster send signals to a fanal in a different cluster. Since for a clique there is only one active fanal in a cluster the signal strength of each active fanal is normalized to get a total signal strength equal to unity. 52
- 2.9 An example of ambiguity in the network where a fanal (corresponding to a sub-message) is missing. The ambiguous fanals are connected to the known fanals with dashed lines. 54
- 2.10 A situation in the network that cause fanal activity oscillation. This is when the fanals indicated with star markers are activated in odd iteration steps and deactivated in even iteration steps. 58
- 2.11 Type III error rate is compared to total error rate for a network with $\chi = 100$ clusters and $\ell = 128$ fanals in each cluster. The partial messages have two unknown sub-messages $c_e = 2$. It is observed that over 30% are Type III errors. 60
- 2.12 The retrieval error rate of both blind and guided recovery of M i.i.d. messages. The network has $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The partial messages have $c_e = 3$ unknown sub-messages. . . . 61

2.13 The retrieval error rate of both blind and guided recovery of M i.i.d. messages. The network has $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The partial messages have $c_e = c/4$ unknown sub-messages, i.e., $\epsilon = 0.25$ 62

2.14 The retrieval error rate of messages that are distorted by permutation. The network has $\chi = 100$ clusters and $\ell = 64$ in each cluster. The stored i.i.d. messages have order $c = 12$ 64

2.15 The set implementation error rate as a function of the density of the network composed of $\chi = 100$ clusters of $\ell = 64$ fanals each, with cliques of order $c = 6$ or 9 . Densities corresponding to efficiency-1 diversities (M_{\max}) are also indicated. 66

2.16 The number of stored messages is depicted for the network with $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The curves are given for several blind retrieval error probabilities where $\epsilon = 0.25$ 68

2.17 The message order pairs are depicted for the network with $d = 0.5$, $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The curves are given for several blind retrieval error probabilities where $\epsilon = 0.25$. The least error probabilities are given in the legend and their corresponding pair has filled marker. 70

3.1 Communication Channel Model 79

3.2 Information-theoretic Block Diagram 80

3.3 Discrete Memoryless Erasure Channel (DMEC) diagram for the puncturing block. The information flow is from left to right. 83

3.4 Discrete Memoryless Erasure Channel (DMEC) diagram for the neural network memory. The information flow is from left to right. 83

3.5 Differential Information Rate I_{diff} versus Number of Stored Messages M is illustrated for the network with $\chi = 100$ and $\ell = 64$. $\epsilon = 0.25$ and $c = 12$ 85

3.6	Upper bounds on P_e exploiting the DMEC model for the network. The network has $\chi = 100$ clusters, $\ell = 64$ fanals in each cluster. The stored messages' order is $c = 12$	86
3.7	Upper bounds on P_e for the network with $\chi = 100$, $\ell = 64$ and $c = 12$	88
4.1	Compressed-sensing has been modeled as a single-layer neural network. The connections have random i.i.d. Gaussian weights.	98

List of Tables

4.1	Sufficient and Necessary Conditions on n for Exact Support Recovery in the Linear and the Sublinear Regimes. The sufficient condition in [Rad11] holds only for $\lambda^2 = \Omega(\frac{1}{k})$ or $\lambda^2 = O(1)$ whereas the other sufficient and necessary conditions hold for the entire signal-to-noise ratio range.	108
-----	--	-----

Chapter 1

Introduction

The brain is the most complex organ of the human body and the center of the nervous system. Although the brain has been studied for a long time from different aspects, such as philosophy, biology and psychology, it remains largely a mystery. Among the revealed properties of the brain, computational capabilities, power efficiencies and enormous memory capacity of the brain are the most significant features to the information technology community. Evolution has made the brain a powerful computing and reasoning machine. It has also evolved to a power efficient system that consumes only 10^{-16} joules per operation per second, whereas modern computers consume up to 10^{10} times more energy [Hay99].

The elements of the brain are distributed parallel processors, resulting in significant computational power. Neural computations take place in millisecond-scale, whereas semiconductor circuits perform nanosecond-scale operations. Today, semiconductor circuits have reached a critical point: to increase the processing speed, the clock frequency has to be increased. A higher clock frequency requires smaller components and circuits but the circuits can be shrunk only up to a certain limit. To speed up processing, another possibility is to increase the number of transistors, which increases the power consumption. This leads to heat production in a small area and the electronic circuit can work in a limited temperature range. This

is where the importance and power of the biological neural systems become more evident. Although they are slow at micro operations (in neurons, dendrites and axons), they have remarkable performance when it comes to complicated processing tasks, such as image processing [Hay99]. Modern computers may use biological parallelism features to overcome their circuitry limitations.

The capability of learning and generalizing is a property of the biological neural networks. The human brain learns an enormous amount of information through input stimuli and it is then able to respond to the stimuli that were learned before. The brain may also compute a response to a stimulus without knowing the direct answer by inferring from what has been learned. This is a desirable feature to be implemented in man-made machines [Hay99].

The brain has been divided anatomically and functionally into many regions. The one which is of our main interest throughout this report is the *neocortex*, which is also known as the *grey matter*. It is the 2 to 4 mm thick outermost layer of the cerebral hemispheres. This thin layer is the center for controlling physical movements, emotions, and reception and perception of vision, auditory, semantics, etc. Although there are many parts of the brain that are involved in what is called memory, we focus on this very recently biologically evolved region.

1.1 Memory

Memory in the human brain is the process of learning, storing and retrieving information [San03]. This process can be studied with respect to both biological and psychological aspects. The brain at the biological and circuitry levels is a connectionist network consisting of nodes (neurons) and connections (axons, dendrites and synapses) where information is stored in the connections [VC10]. The connections are mostly built during the infancy. They change throughout the lifetime when the brain learns by being exposed to input stimuli. A simplistic model is that

the brain learns by loosening and strengthening its connections, which is known as synaptic plasticity.

The brain at the mental level is considered by psychologists and philosophers as a system with a set of sub-systems in which information is processed and stored. From this point of view, learning, recovery, storage and decision-taking functions are treated abstractly without getting into cell-level operations. This phenomenon has been studied throughout history by philosophers such as Aristotle and psychologists [San03]. Ancient thinkers considered the memory as a storage space that keeps information without processing capability. Aristotle believed that the outside world leaves imprints on the brain in the mental matter. He modeled memory as a function that also connects the content of the mental matter rather than just a storage unit [San03]. This rather old perception of human memory has influenced our knowledge of memory even today. It is known now that memory and processing are not two distinct functions of the brain but rather are of the same process [VC10].

There are two modern psychological memory models to explain the mechanisms of the human memory. These are the Atkinson-Shiffrin [AS68] and Working Memory [BH74] models. The Working Memory model is proposed by Baddeley and Hitch and describes Short-Term Memory (STM) and its mechanisms. The model proposed by Atkinson and Shiffrin consists of Sensory Memory (SM), STM and Long-Term Memory (LTM) stages. Sensory memory describes the first few hundred milliseconds after informative items are perceived. Short-term memory refers to the first few seconds after the items are perceived. Long-term memory describes the memory's ability to remember what is learned after a long time from several minutes after informative items are perceived to much longer. Apart from the importance of each stage, it is worth noting that the storage in LTM requires perception of the items in early stages of SM and STM. Thus, storing an item in the long-term memory (which we refer to in everyday life as memory) is a process

rather than isolated operations. In this report, the memory is a system that captures a set of input information (through input stimuli) and is able to retain (for a long time) and reuse it.

Retrieval of stored information is a part of a memory system. In the brain, retrieval is carried out in different forms which are categorized into *recall* and *recognition* [San03]. Recall is the ability to retrieve what was learned by giving a hint. Recognition is the ability to acknowledge what was learned in the past [San03]. These are two fundamental functions of the memory that are also considered in the design of artificial memories. A memory that is exploited as an associative memory carries out a recall function. In set implementation applications, one needs to know if an element is already in a set to decide whether the element should be added to the set or not. This is what is called recognition.

The recall function has three types, namely *free recall*, *cued recall* and *serial recall*. Free recall is when the memorized items are retrieved without their order. Cued recall is the retrieval of an item through another item, where the items in question were learned in a pair or a group. Serial recall is the retrieval of items in the order that they were learned.

1.2 Compression and Robustness

The brain is a complex system including information processing and storage subsystems. It is known that the brain is capable of storing a large amount of information with fairly acceptable retrieval accuracy [WLW03]. To accomplish this, it requires that the information is stored efficiently with a mechanism to assure the robustness of recovery. An efficient storage strategy is one that requires the fewest memory to hold information content, which is known as compression. On the other hand, information retrieval is robust if there is redundancy to make error detection and correction possible. Reliability at the circuitry and biological level is obtained by

creating multiple paths between neurons. Therefore, if neurons lose part of their connections, the brain is still able to recover at least part of what has been learned.

1.2.1 Neural Coding

The brain receives an enormous amount of the sensory information from outside such as visual stimuli and from inside which are thoughts (as a product of thinking and imagination) [FÖ2]. This information is crude and needs to be prepared in a form that the brain is capable of perceiving and storing. It is transformed into electrical signals and chemical reactions which neurons, axons, dendrites and synapses are capable of handling. Moreover, information is also organised in the brain in a way to be efficiently processed and stored, rather than to be treated as a pile of random data. This requires a system that projects stimuli. The *neural code* is the way that the stimuli information is projected into the brain [BT99].

1.3 Artificial Neural Networks

Artificial neural networks are an effort to build systems that mimic biological neural systems, in order to exploit the capabilities and power of their architecture. These capabilities are parallel computation, adaptivity, and the ability to learn and generalize [Hay99]. In this work, an artificial neural system is proposed to model the human neural system, which has the brain as the center of the nervous system. The nervous system may be seen in three parts. Receptors convert stimuli into electrical signals to be sent to the brain. This process is represented as the neural network in Fig. 1.1. The effectors in the nervous system convert the electrical signals received from the brain into system responses [Hay99]. In Fig. 1.1, the arrows show that the communication between the brain and receptors and effectors is bidirectional.

There are two kinds of neural networks, namely feedforward networks and recur-

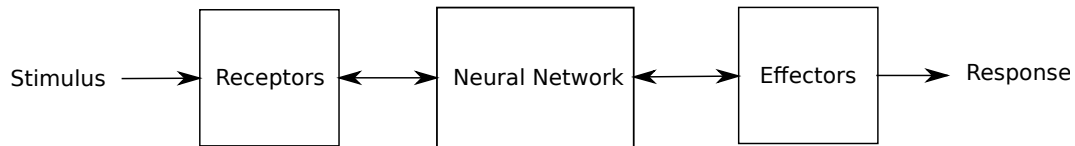


Figure 1.1: Block diagram representation of nervous system.

rent networks. The main difference between these two types, as their names imply, is that the former do not use feedback paths where the latter do exploit feedback paths. Feedforward networks process the received signals in input nodes and pass the computation results (which can be linear summation and thresholding) to the output or the input of the next layer. In this architecture, the processed signals in a layer are not fed back to that layer in any form. Therefore, more complex functions can only be implemented through adding layers to the network. Feedback has a profound effect on the overall performance of the network and brings the network richer dynamics [Hay99]. This way, the network is able to implement complex functions with fewer layers than feedforward networks. In Fig. 1.2, a simple form of feedforward networks is depicted. The signals are received by input source nodes and are sent to the output neurons. The connections are weighted and the arrows show the direction of the signal flow. Fig. 1.3 shows a simple recurrent network, namely the aforementioned feedforward network with a feedback path. Networks that comprise at least a single feedback path are considered recurrent, otherwise they are said feedforward [Hay99]. Artificial neural networks are exploited to compute complex functions, to perform memory functions and to carry out combinations of these two. The memory in neural networks appears in different forms such as the capability of the network to adjust and retain the synaptic connection weight in the learning process. In this report the memory function is considered, that is to say the function that makes a network learn and retrieve information. There are two types of memories, namely *content addressable* or *associative* memories, and *index addressable* memories. In the latter, the machine accesses the

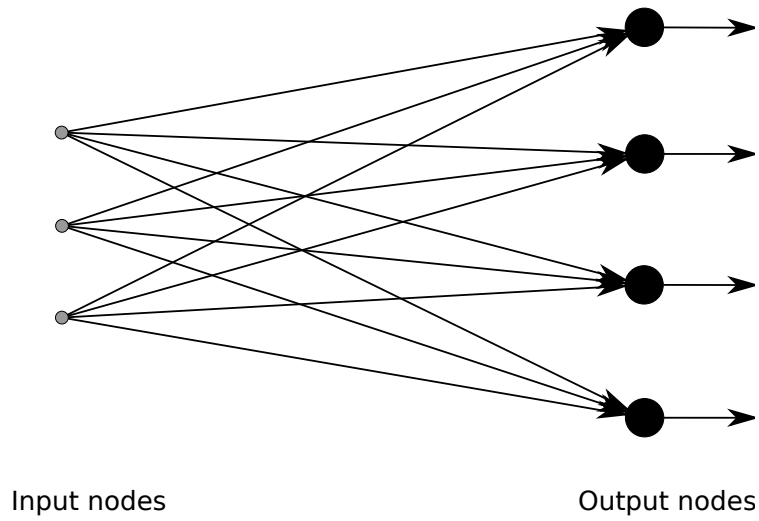


Figure 1.2: Diagram of a single-layer feedforward neural network.

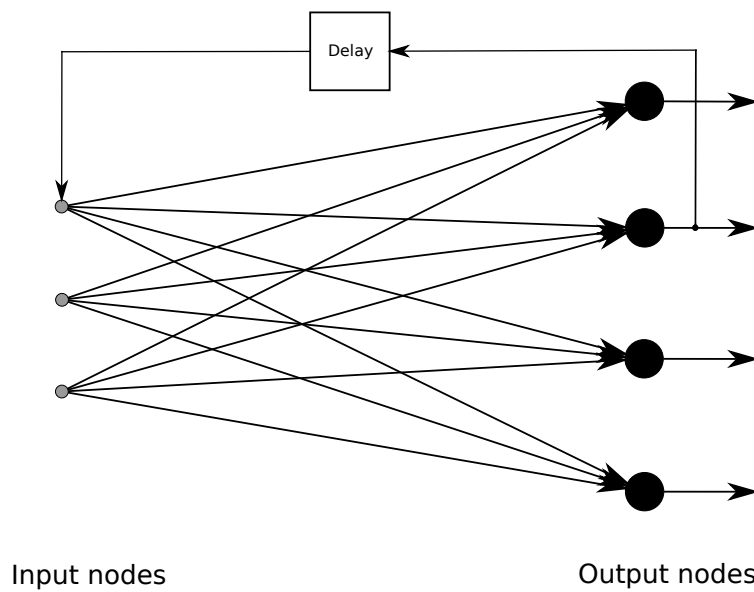


Figure 1.3: Diagram of a single-layer recurrent neural network with only one feedback.

memory content by knowing its storage address (i.e. index) whereas in the former the memory content is retrieved from a part of the learned item or a key that it is associated with. In other words, the memory is accessed by the contents that are associated to each other and not with their address. Content addressable memory is the type of the memory exploited by the human brain (see section 1.1). It is different from traditional computer memory in that if the address of memory contents are partially known, the stored information can not be recalled.

The associative memories are evaluated with two fundamental quantities, *memory capacity* and *content addressability*. Memory capacity indicates the maximum amount of information that a network is capable of storing. Content addressability is the ability of the network to retrieve the stored information correctly. Both of these characteristic are defined and elaborated in detail in the following chapter for the specific network that is considered in this report.

1.4 Thesis structure

Chapter 2 The proposed neural network memory that is capable of storing sparse messages is presented. The network is analysed and compared with the classical network from which it has been derived. It is shown that the proposed network improves significantly the maximum number of stored messages when the messages are strongly sparse.

Chapter 3 The proposed neural network is investigated from information-theoretic aspects. It can be exploited for channel coding applications and the network itself can be considered as a joint source-channel coding system. The network is then analysed using Fano's inequality from which a set of recovery constraints are derived.

Chapter 4 Recovery conditions for Compressed-Sensing (CS) are the subject of this chapter. It is mainly used for dimensionality reduction which is also a topic of machine learning and neural networks. CS is studied exploiting information theory, through which a few tight sufficient recovery conditions are computed.

Chapter 2

Sparse Neural Network

Memories

In this chapter, an artificial neural network based on the human neocortex structure is presented. This network embodies several features of the biological neural systems and more specifically of the neocortex in its design.

The brain comprises neurons, dendrites, axons and synapses as its main material and other cells such as glial cells for the metabolic support (although this may seem oversimplified). They construct a grid network with neurons as its nodes and, dendrites, axons and synapses as its connections [Hay99]. This structure is well recognised in mathematics and computer science as a graph. Hereafter, the artificial neural network is considered as a graph consisting vertices and edges. In [MP43, Hay99] vertices represent neurons and edges represent dendrites, axons and synapses. Although the graph representation of neural networks is adopted in this document, the notion of vertices and edges is different. More specifically, the vertices represent a group of neurons which is explained in the following section.

The principle difference between mathematical graphs and neural networks are the underlying dynamics of the neural networks: the connections (edges of the graph) and nodes (vertices of the graph) in a neural network may change due to

the effect of input stimuli from time to time [Hay99]. This resembles the living tissues and cells in the human brain. That turns the brain into an evolving and self-organising organ rather than a solid connected grid. The way that the vertices of a neural network are organised and connected has a profound effect on its performance and efficiency. To obtain a network with a desirable performance and efficiency, we refer to the brain that has evolved towards, efficiency throughout time.

In the human brain, the most recently developed part, the neocortex, is of our interest. The neocortex is a well structured layer of the brain and it is the center of conscious thoughts and language. The human neocortex has inspired the development and design of the network (in this document) that is sparse of different levels.

The basic structure of the proposed network is based on the network in [GB11]. This is adopted and extended in this document to store sparse data structures.

2.1 Sparse Coding

The notion of sparsity appears in different forms in the biological neural systems. The very first stage of sparsity in the brain is at the reception of numerous input stimuli. The neural systems are selective when they are exposed to the input stimuli. They do not capture every stimulus but a few at a time [Mil56]. This can be observed in the daily life where we always capture those information that we intend to. For example we are able to concentrate on a specific person's voice in a crowd. This can also be unconscious where much of the sensory information received by our skin is automatically neglected by the brain but those that are significant such as pain. The perceived information are passed to SM and STM afterwards (see section 1.1) to be noticed by the conscious mind. They are passed to a deeper memory level, LTM, depending on whether they need to be remembered

later. This structure acts as a biological sieve that filters out the unnecessary information.

The second sparsity feature is observed in the organization of neurons and connections. The neural system is organized in different sections and the neocortex is not an exception, in its both vertical and horizontal dimensions. It has six horizontal layers containing various types of the neurons and other vital cells. The one with greater importance in this report is columnar division of the neocortex [Mou97]. A column from top-down perspective is a cylinder containing a group of neurons with the same receptive field. In a column, there are microcolumns¹ that are groups of about 80 ~ 100 neurons [Mou97]. They are believed to be the basic units of the neocortex [Mou97]. The microcolumns are connected through horizontally organized neural layers which constructs a grid [Mou97]. A single neuron has a simple structure that may not be able to store and process a remarkable amount of information. Thus, a stand alone neuron is not sufficiently informative, which implies that the complicated operations in the brain require a developed organization of neurons. This has led the brain to evolve into a structure in which the stored information in the neocortex appears as activated microcolumns that are connected horizontally [F02]. These connections are locally dense and recurrent whereas they are sparse at the global level [San03].

The brain stores information that is received from the environment and its internal state (e.g. thoughts) in sparse patterns of simultaneously active microcolumns. This theory is called “neural coding” (see section 1.2.1) [ALP06, F02]. This sparse representation of information, i.e., neural coding is the feature that is exploited in the neural network in [GB11] and even more in this report.

The neural network is modeled as a graph in which vertices represent microcolumns in the neocortex. The vertices are further organized into clusters as groups of vertices. A vertex is called a *fanal* to indicate that in each cluster, there is only

¹A microcolumn is also called a *minicolumn* [Mou97]

one active node for a stored message. Each cluster represents a column in neocortex consisting of about hundred fanals. A collection of about hundred clusters is considered to be a network. In Fig. 2.1, the described model of the network

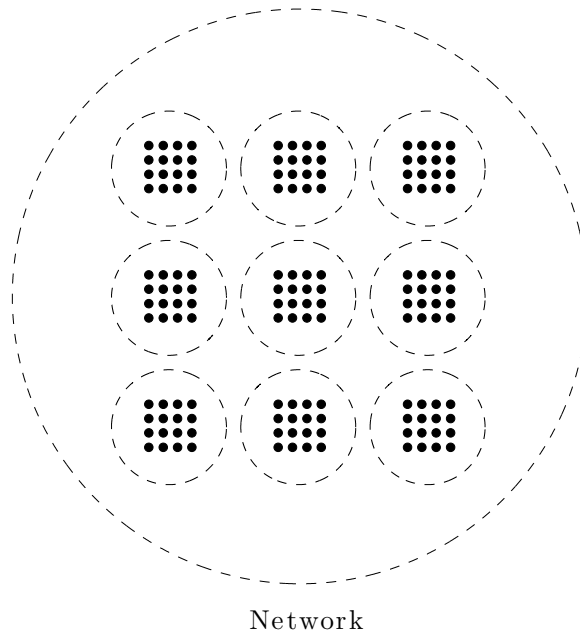


Figure 2.1: The neural network inspired by the human neocortex. The dots represent the microcolumns and the regions surrounded with dashed lines show the columns (clusters) and a macrocolumn (the network) in the neocortex.

is depicted in which the number of fanals in each cluster is equal to a constant. This model is further developed and analysed in the following sections as a neural network memory for which the network dynamics and its properties will be given.

2.2 Messages

Information in digital devices is quantized and packed into groups of bits and bytes, etc. In this report, information is represented with vectors in which the elements are drawn from an alphabet. The vectors can be the product of sampling real-world analog information or they may be obtained by the post processing of an informa-

tion source. It is assumed that these vectors are sparse, i.e., a few of their elements bear information and the rest is negligible or blank. Sparse signals and vectors are of great importance in information science and they have been studied extensively in mathematical frameworks such as Compressed Sensing [Don06, CT05, CW08], Information Theory [ASZ10, AT10, WWR10, Wai09], Machine Learning [WCF07], etc. The goal of the mentioned studies is to exploit sparsity to improve the storage, sampling, energy efficiency, etc. of information systems. Likewise, sparsity is a strong characteristic to be used in neural network memories.

An application of neural network memories and more specifically associative memories is image recognition. Assume a set of images such as peoples' faces. There is an image (in the image set) to be recognized when a part of it is covered or missing. This requires a post-processing framework that extracts unique image specifications which are stored (in the neural network memory) and used as the image signature. A modified Speeded Up Robust Features (SURF) [BETvG08] or Scale-Invariant Feature Transform (SIFT) [Low99] may be applied on images to get the images' specifications. These transforms return a set of pixel coordinates as a set of interest points, also known as keypoints. For each image with χ pixels, a vector (of size χ) is constructed in which the keypoints take a nonzero value and the rest are zero. It is assumed that these vectors uniquely represent each image in the image set and they are so-called specification vectors. Specification vectors are stored in the proposed network. When part of an image is missing, it appears as missing nonzero elements in its corresponding specification vector. The defected image is then recognized by feeding the network with its partially known specification vector and using the properties of the associative memory especially redundant coding.

Another application is wireless communications. A technique so-called Cognitive Radio [Hay05, IGM99] is the center of attention among the researchers of the field in recent years. Communication spectrum is a limited and therefore expensive

resource. This gets worse by the advent of new wireless hand-held and personal communication devices. In the past, organisations such as International Telecommunication Union (ITU) allocated communication bandwidths to many standards, especially mobile networks. Today, there is almost no more free-space left in the frequency spectrum. This has led the communications bodies to allocate shared bandwidths for various applications such as personal communications. There are various strategies to use shared radio spectrum bandwidths. One of the cognitive radio techniques is the *opportunistic* scheme through which radio devices access randomly the spectrum range. The issue that rises with this opportunistic scheme is that each radio device is responsible for surveying the frequency spectrum to find an empty slot (in time, frequency and space) for communication. Therefore, a radio device sees the frequency spectrum as a vector in which the occupied frequency slots appear as nonzero elements and the rest is zero. This vector is named here as spectrum state vector. In hand-held and battery-driven devices, energy is a valuable asset to be conserved for vital operations. Assume the spectrum changes among a limited number of patterns for a period of time. In this case the device may fully sense the spectrum and store the spectrum state vector in the proposed network (as an associative memory). Afterwards, the radio device may increase its energy efficiency and communication rate by sensing a limited part of the whole spectrum and recover the rest from the neural associative memory.

The basic item which the network perceives as an input stimulus is a *message*. Following the above discussion, a message with a few information bearing elements is so-called *sparse*. The sparse messages that are dealt with in this report can be either naturally sparse or result of a transform such as Fourier and Wavelet. In this report, the sparse messages are treated by the network regardless of their real-world source and they may belong to one of the mentioned or other possible applications.

A message is a vector with χ elements. An element of the message is so-

called *sub-message* that is drawn from an alphabet denoted by \mathcal{A} or it may be *blank*. The blank elements are the none information-bearing elements because they are null or negligible. The number of non-blank (significant) sub-messages is so-called *message order*. The sub-message alphabet has finite cardinality $|\mathcal{A}| = \ell$. Throughout this report, sub-messages are assumed to be uniformly distributed over the alphabet \mathcal{A} . Although the alphabet size can be arbitrary depending on the application, it is often a power of two. This choice makes the network compatible with the digital systems where information is quantized in bits. Therefore, a message may be considered as a sequence of bits to be stored in the network. This requires that the message is divided into c groups of $\kappa = \log_2 \ell$ bits (see section 2.3).

2.3 Storage

Neural networks are connectionist networks composed of nodes and their corresponding connections. This structure can be modelled as a graph in which these nodes are vertices and their interconnections are edges of the graph [Hay99, GB11]. Throughout this document, a network with binary connections is considered; in other words, either there is a connection between two nodes or there is no connection. In the following, the network and its components that store messages according to the sparse coding approach are defined.

Definition 1 (Neural Network). According to the previously given theory, a neural network is an undirected multipartite² graph $(\mathcal{F}_1, \dots, \mathcal{F}_\chi, \mathcal{W})$ where $\mathcal{F}_1, \dots, \mathcal{F}_\chi$ are sets of vertices and \mathcal{W} is the set of edges. All sets of vertices have equal cardinality, $\forall j \in \{1, \dots, \chi\}, |\mathcal{F}_j| = \ell$.

²A multipartite graph is a graph whose nodes are separated into disjoint sets. The edges of the graph may only be among nodes in disjoint sets of nodes, i.e., there is no edge between nodes belonging to the same set.

Definition 2 (Cluster). A set of vertices $\mathcal{F}_i \in \{\mathcal{F}_1, \dots, \mathcal{F}_\chi\}$ is a cluster. A cluster is indicated by an index and the set of cluster indices is $\mathcal{C} = \{1, \dots, \chi\}$.

Definition 3 (Fanal). A vertex of the graph which is an element of $\cup_{i=1}^{\chi} \mathcal{F}_i$ is a fanal. A fanal in a cluster is indicated by an index. The set of fanal indices is $\mathcal{F} = \{1, \dots, \ell\}$.

A neural network is uniquely identified with $(\mathcal{W}, \chi, \ell)$ where ℓ is the number of fanals in each cluster and χ is the number of clusters in the network that is so-called afterwards *cluster space size*. A pair $(i, j) \in \mathcal{C} \times \mathcal{F}$ uniquely specifies a vertex in the graph.

The defined graph along with the storage and retrieval mechanisms is a network that is capable of interacting (with another system or the outside world through an interface) as a system. The very first step toward building such a system is the storage process that the storage efficiency depends on it. It also significantly influences the retrieval process, which will be elaborated in the next sections.

The storage is the process of mapping a set of messages into the network through constructing binary connections among the fanals. To exploit the network efficiently, a well designed storage procedure is required in which various goals are pursued. The first purpose of the storage is to map a message in the network by constructing or removing edges of the graph (connections) in a manner that the message could be retrievable afterwards. The neural network memory retrieves the stored message by providing its associated content (e.g. partial message or blurred/noisy message) to the network. This implies that the storage is a one-to-one function. This mapping may be considered one-to-one where only a single message is subject to storage and recovery in the network.

The efficiency of the storage is also an important issue along with the correct storage and retrieval of messages. In other words, the storage should exploit the intrinsic property of the network and the messages (e.g. sparsity) to perform a kind

of compression (lossy or lossless). This is to avoid the storage of messages as a pile of information that would imply an inefficient use of resources. In this network, such a compression scheme is implemented by sharing fanals and connections among stored messages. This compression occurs where at least two distinct messages have a few corresponding equal sub-messages. For example consider two messages drawn from alphabet $\mathcal{A} = \{1, 2, \dots, 128\}$ and dashes as blank sub-messages,

$$m_1 = [-, 21, -, -, 2, -, 12, -, 5, -, -, -, 115, -]$$

and

$$m_2 = [-, 45, -, -, 2, -, 12, -, -, 65, -, -, 74, -]$$

where the fifth and seventh sub-messages of these two messages are equal. These messages may be associated with two disjoint cliques in the network that share two fanals and a single connection. This has side effects on content addressability of the network where in some cases the network may not be able to recover the stored messages. It is due to the fact that the compression in this network is obtained by sharing network connections, that is, by a lossy compression (see [CT06]) in which a part of stored information may not be recovered. The storage is also aimed to embed a well designed redundancy to combat deficiencies in the network connections and the input stimuli (e.g. noisy and/or partial messages) at the retrieval. It was also mentioned that the compression in use in this network may result in weakening content addressability of the network. A smart redundancy is the answer to such negative effect. Above discussion refers to source coding (compression) and channel coding (embedded smart redundancy) in information theory that was studied and quantified for the first time by Shannon in [Sha48]. This network will be studied from this perspective in the later sections and chapters of this report. The mentioned design criteria in the network are satisfied through construction of complete subgraphs. Assume there is a message to be stored. Each sub-message (from alphabet \mathcal{A}) is mapped to its corresponding fanal (one of the

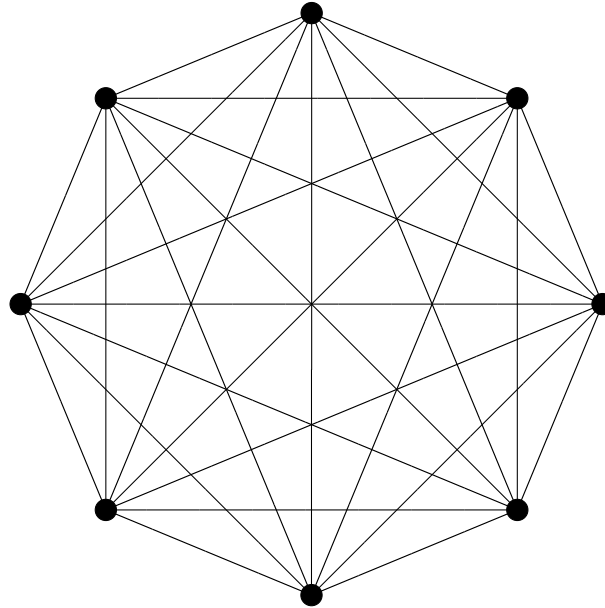


Figure 2.2: A clique (a complete subgraph) that represents a message containing eight sub-messages in the network. This clique has twenty-eight connections. Although a single loop graph with only seven connections can associate the fanals, a fully connected graph provides robustness against deficiencies.

ℓ fanals in the corresponding cluster). Then the message is stored by connecting all these fanals. This results in a complete subgraph for a message in the network³. In Fig. 2.2 a clique, consisting of eight fanals and twenty-eight connections is depicted. The total number of connections in a bidirectional clique (of this network) is

$$\frac{c(c-1)}{2} \quad (2.1)$$

where c is the message order, that is also called clique order.

In this neural clique (a clique in the neural network), the fanals have more than a single connection to their neighbour fanals. This makes the network capable of retrieving the message where a few but not all sub-messages (and correspondingly fanals) are unknown which is a robustness feature.

³A maximal complete subgraph is so-called *clique* [MM65, GB11]. The storage of a sole message in the network constructs a clique.

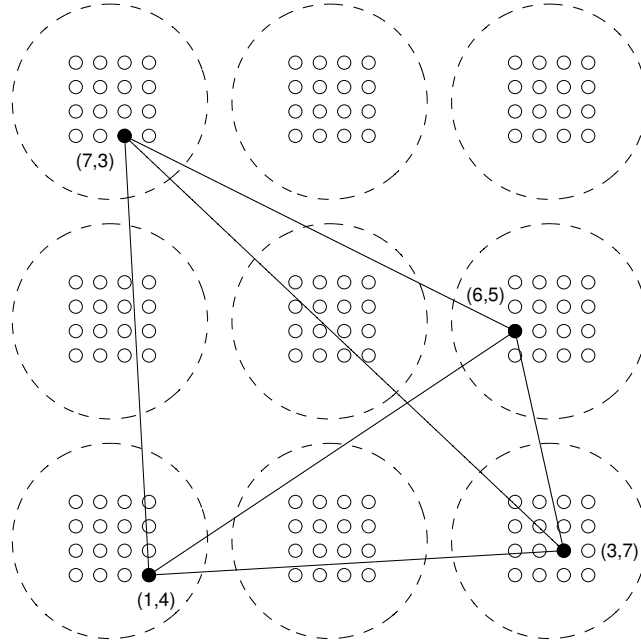


Figure 2.3: A network composed of 9 clusters of 16 fanals each. A message with message order 4 forms a clique in the network.

In Fig. 2.2, a clique is shown as a single entity. In Fig. 2.3, a message clique has been introduced to the network, in order to elaborate the storage procedure. In this figure, a neural network with cluster space size $\chi = 9$ and $\ell = 16$ fanals in each cluster is depicted. Assume message $m = [4, -, 7, -, -, 5, 3, -, -]$ and alphabet $\mathcal{A} = \{1, 2, \dots, 16\}$. The assigned set of fanals to this message is $\rho = \{(1, 4), (3, 7), (6, 5), (7, 3)\}$ where the first element in the pairs indicates the cluster index and the second element indicates the fanal index. The set of connections is then:

$$W(\rho) = \{ \{(1, 4), (3, 7)\}; \{(1, 4), (6, 5)\}; \{(1, 4), (7, 3)\}; \\ \{(3, 7), (6, 5)\}; \{(3, 7), (7, 3)\}; \{(6, 5), (7, 3)\} \}.$$

In this specific example, it is assumed that $\mathcal{A} = \mathcal{F}$, thus, the mapping is Identity

function ⁴. It is also assumed (in this example and in this report) that the support ⁵ of the messages are chosen uniformly random from $\binom{\chi}{c}$ subsets of size c (the message order). Likewise, the storage of a message set involves repeating the same operations.

In the following, the storage process is formalized for the network with arbitrary cluster space size χ and number of fanals ℓ in each cluster. Assume a network $(\mathcal{W}, \chi, \ell)$ that stores a message set \mathcal{M} . Then, the storage involves constructing connections' set \mathcal{W} of the neural network from the message set.

Assume message m is subject to storage, a one-to-one function b maps each non-blank message node $m_r \in m$ and $m_r \in \mathcal{A}$ to a cluster index $i \in \mathcal{C}$.

$$\begin{aligned} b : \mathcal{A} &\longrightarrow \mathcal{C} \\ m_r &\longrightarrow i = b(m_r). \end{aligned}$$

Function b is a one-to-one function, which guarantees that there is no more than only one activated fanal in the same cluster for a message. This is one of the conditions that enforces the sparsity of the network.

Next, a one-to-one mapping g maps each sub-message $m_r \in \mathcal{A}$ to a fanal index $f \in \mathcal{F}$.

$$\begin{aligned} g : \mathcal{A} &\longmapsto \mathcal{F} \\ m_r &\longmapsto f = g(m_r). \end{aligned}$$

After assigning the clusters and fanals through b and g , function D pairs corresponding clusters and fanals of message m in a set.

$$D : m \longrightarrow \{(b(m_r), g(m_r)) : m_r \in m\}.$$

To store the message in the neural network, one-to-one function W maps $\rho = D(m)$ to binary graph edges (network connections). It constructs a subgraph

⁴The cluster and fanal indices are assigned by counting them in the figure from left to right and down to top.

⁵The support of a message is the set of indices of sub-messages that are non-blank.

that involves graph vertices (fanals) in ρ ,

$$W(\rho) = \{ \{(i', j'); (i, j)\} \in \rho \times \rho : i \neq i'\}.$$

Definition 4 (Neural Training of set \mathcal{M}). Let $m \in \mathcal{M}$ be a message to be stored by the neural network. The training consists in setting the binary edges of graph $\mathcal{G} = (\mathcal{F}_1, \dots, \mathcal{F}_\chi, \mathcal{W})$ as

$$\mathcal{W} = \bigcup_{m \in \mathcal{M}} W(D(m)) = \bigcup_{m \in \mathcal{M}} W(\rho). \quad (2.2)$$

It is worth mentioning that as the given storage procedure suggests, the sequence order of messages is not stored. The messages can be stored in any order and at any time or all at the same time.

Since b and g are one-to-one functions, it can be directly deduced that W is also a one-to-one and reversible function. Assume any two messages m and m' that hold $m \neq m'$. This implies that we have $W(D(m')) \neq W(D(m))$.

The connections' set of \mathcal{W} can be alternatively represented as a four dimensional matrix

$$w_{(i'j')(ij)} = \begin{cases} 1 & \text{if } \{(i', j'); (i, j)\} \in \mathcal{W}, \\ 0 & \text{otherwise,} \end{cases} \quad (2.3)$$

where w has size $|\mathcal{C}| \times |\mathcal{F}| \times |\mathcal{C}| \times |\mathcal{F}|$. Such matrix representation of graph connections compresses the information (where the network is built and simulated). This way of representing network connections results in saving memory. Moreover, it simplifies the description of recovery algorithms that are explained in the following section.

Previously in this section, the storage procedure in the network was spelled out. The design and study of the network requires to formalize it with measurable quantities. These quantities are employed to optimize the network from different aspects. In the following, first the network is considered to store messages of length χ with constant number of significant (non-blank) sub-messages c and afterwards

it is extended to the storage of messages with variable number of significant sub-messages.

The connections among fanals are binary from which the total number of storage bits in the network can be deduced as

$$Q = \frac{\chi(\chi - 1)\ell^2}{2} \quad (2.4)$$

that is the number of potential bidirectional connections. This quantifies the maximum number of available bits and it may not be confused by the maximum number of stored bits (which is always less due to redundancy).

In graph theory, the edge density is used to study the average connectivity of the graph nodes. It is the ratio of the number of edges to the total number of possible edges in the graph [Die10]. Assume a single message of order c is stored, then the network density is $\frac{c(c-1)}{\chi(\chi-1)\ell^2}$. It is the ratio of the number of edges in a clique with c nodes in (2.1) to the total number of possible connections given by (2.4). The network connections after the storage of a large number of messages are considered asymptotically i.i.d. and uniformly distributed. This results in the probability of having a connection between two fanals of distinct clusters being approximately equal to the network density. Assume M i.i.d. and uniformly distributed messages are stored. The network density becomes:

$$d = 1 - \left(1 - \frac{c(c-1)}{\chi(\chi-1)\ell^2}\right)^M \quad (2.5)$$

where c is the message order and χ is the cluster space size. Reversely, the number of stored messages as a function of the network density is

$$M = \frac{\log(1-d)}{\log\left(1 - \frac{c(c-1)}{\chi(\chi-1)\ell^2}\right)}. \quad (2.6)$$

In Fig. 2.4 the evolution of the network density as a function of M is shown where $\chi = 100$ and $\ell = 64$. A network with a density close to unity may not recover the stored messages. As a result, a low density network is often desired. It can also be

concluded that since the network density grows slower for smaller message orders, the network is capable of storing more messages. In other words, the number of stored messages is proportional to the sparseness of messages.

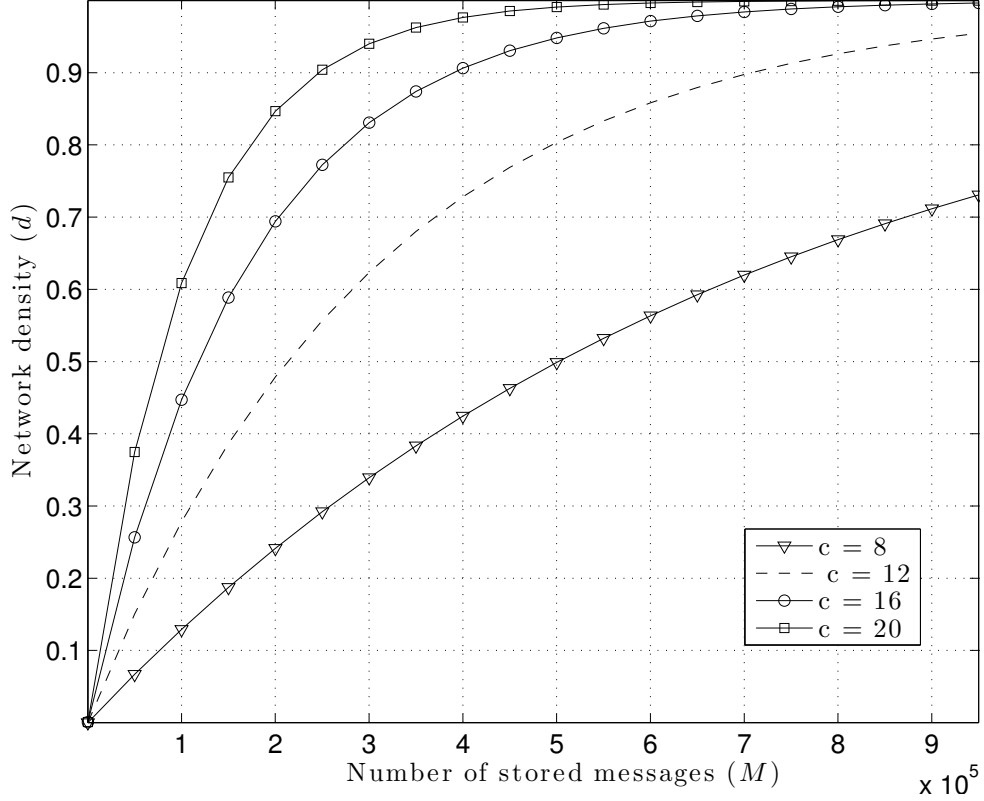


Figure 2.4: The network density for four values of c as a function of the number of stored messages M where $\chi = 100$ and $\ell = 64$.

A message $m \in \mathcal{M}$ with message order c (all messages have the message order c) stores $\log_2 \left(\binom{\chi}{c} \right) + c \log_2(\ell)$ bits in the network. For M messages, the network stores

$$B = M \left[\log_2 \left(\binom{\chi}{c} \right) + c \log_2(\ell) \right] \quad (2.7)$$

bits of information. The first logarithm is the number of bits required to store non-blank cluster indices and the second logarithm is the number of bits required to store the non-blank elements' values. The efficiency of the network is defined as the ratio of the number of stored bits to the number of total storage bits

$$\eta = \frac{B}{Q} = \frac{M \left[\log_2 \left(\binom{\chi}{c} \right) + c \log_2(\ell) \right]}{0.5\chi(\chi - 1)\ell^2}. \quad (2.8)$$

The efficiency is a metric without dimension that varies between 0 and 1: $0 \leq \eta \leq 1$. It may be used to measure the amount of redundancy for a stored set of messages.

Efficiency $\eta = 1$ occurs where all the network connections have been used to store a message set (i.e. there is no redundancy in the network). Diversity is defined as the maximum number of messages that the network is capable to store. This occurs when $\eta = 1$,

$$M_{\max} = \frac{\chi(\chi - 1)\ell^2}{2 [\log_2 \binom{\chi}{c} + c \log_2(\ell)]}. \quad (2.9)$$

Fig. 2.5 illustrates the efficiency-1 diversity in (2.9) for several values of ℓ and

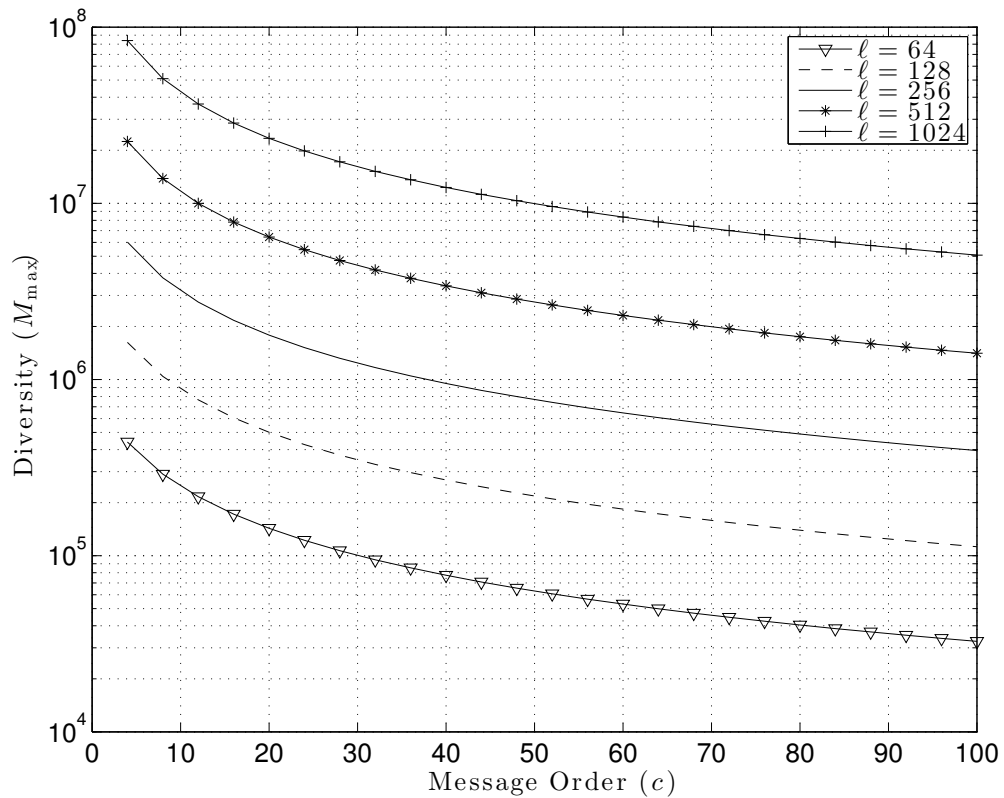


Figure 2.5: The network diversity ($\eta = 1$) is depicted for various values of ℓ when $\chi = 100$ and when the stored messages have a constant order $4 \leq c \leq \chi$.

for a network with cluster space size $\chi = 100$. It can be observed that a network with higher number of fanals in each cluster and higher sparsity level (i.e. lower message order) has larger diversity. The tail of all curves ends at $c = \chi$, that is

when the messages are no longer sparse. This point has the lowest diversity for specific ℓ and χ .

Diversity is an important measure when the network in [GB11] is compared with the proposed network. Indeed, diversity is improved with respect to the one in [GB11]

$$M_{\max} = 0.5(c - 1) \frac{\ell^2}{\log_2 \ell} \quad (2.10)$$

while the proposed network has a diversity⁶

$$M_{\max} = \Omega \left(\frac{\chi^2 \ell^2}{\log_2(\chi \ell)} \right).$$

The latter depends on the total number of fanals $n = \chi \ell$ whereas in (2.10), diversity only depends on the number of fanals ℓ (because the number of clusters is constant and it is equal to the message order). Therefore, the proposed network has significantly improved the diversity. On the other hand, since the network does not store the blank sub-messages, it exploits the network resources efficiently. This results in increasing the number of stored messages compared to the classic network in [GB11].

Now, the storage of messages with variable number of significant sub-messages is considered. Assume messages set \mathcal{M} in which the messages have $c_{\min} \leq c \leq c_{\max}$ significant number of sub-messages. The number of messages with order c is indicated by M_c . the density of the network that stores this variable order message set is

$$d = 1 - \prod_{c=c_{\min}}^{c_{\max}} \left(1 - \frac{c(c-1)}{\chi(\chi-1)\ell^2} \right)^{M_c}. \quad (2.11)$$

In this report, it is assumed that the message order $c_{\min} \leq c \leq c_{\max}$ is uniformly distributed between c_{\min} and c_{\max} . Therefore, the number of messages with order c is $M_c = M / (c_{\max} - c_{\min} + 1)$ when M is large [Pap91]. The binary information borne by a message $m \in \mathcal{M}$ is

$$\log_2 \left(\binom{\chi}{c_m} \right) + c_m \log_2(\ell),$$

⁶ $f(n) = \Omega(g(n))$ iff \exists positive c, n_0 such that $f(n) \geq c g(n) \geq 0 \quad \forall n \geq n_0$.

where c_m is the number of significant sub-messages of message m . Efficiency is equal to

$$\eta = \frac{2 \sum_{m \in \mathcal{M}} \left(\log_2 \left(\binom{\chi}{c_m} \right) + c_m \log_2(\ell) \right)}{\chi(\chi - 1)\ell^2} \quad (2.12)$$

when the message orders are uniformly distributed between c_{\min} and c_{\max} , this becomes

$$\eta = \frac{2M \sum_{c=c_{\min}}^{c_{\max}} \left(\log_2 \left(\binom{\chi}{c} \right) + c \log_2(\ell) \right)}{(c_{\max} - c_{\min} + 1)\chi(\chi - 1)\ell^2}. \quad (2.13)$$

Network diversity as it was computed before in (2.9) by setting $\eta = 1$, is

$$M_{\max} = \frac{(c_{\max} - c_{\min} + 1)\chi(\chi - 1)\ell^2}{2 \sum_{c=c_{\min}}^{c_{\max}} \left(\log_2 \left(\binom{\chi}{c} \right) + c \log_2(\ell) \right)}. \quad (2.14)$$

In this section, the essential parameters to evaluate and optimise the network were given. Indeed, the network can be characterized by its density, d , diversity, M_{\max} , and content addressability metrics such as successful retrieval probability. The retrieval probability and its computation will be discussed in the following sections of this chapter.

2.4 Retrieval

A biological neural system such as the human brain is exposed to the input stimuli during the learning process. These stimuli create or change the connections' strengths, which is the act of learning. Thereafter, the brain is able to retrieve these stimuli in various scenarios. An example of this is the capability of storing and reading the written words. For example, the brain stores a word (spell of the word) which afterwards it is able to remember completely even if it is given a part of it (a few but not all letters of the word) [PB04]. The proposed neural network memory, inspired from the human brain, is expected to perform similar functionality. It is though limited to the storage of a message set that can be considered as

words of a dictionary (similar to human experience of learning and remembering words). The messages (in the message set) consist of sub-messages taken from an alphabet (similar to a language alphabet like English alphabet).

This network is expected to be able to retrieve a stored message from its partially known sub-messages which is the fundamental property of associative memories. The network also addresses the problem where a hypothetical message is to be verified by the network whether it is stored. This is also known as Set Implementation. The brain is also able to recognize morphological or dictation errors. Such situations can take place for example when the letters of a written word are permuted [GW04]. The network is capable of retrieving a message where it is fed by its permuted sub-messages.

The classical retrieval algorithm was first introduced in [GB11]. The retrieval algorithm is based on a variation of *Winner-take-all* (see [GB11, Maa00]) and message-passing. The employed message-passing is similar to a kind of message-passing used in error control coding. In the following, the retrieval algorithm given in [GB11] is elaborated. Although the principles of the algorithm remain unchanged, it is modified in different ways to improve the performance in terms of retrieval error probability.

2.4.1 Winner-take-all

Winner-take-all (WTA) is a nonlinear operation in the human brain that is also adopted in artificial neural networks. It brings significant flexibility and capability to neural networks [Maa00]. The basic role of Winner-take-all is rather simple: among a group of neurons, the neuron with maximum activity level remains active and the rest is deactivated [Maa00]. This may be a good model for STM (Short-Term Memory) where in a group of cells that receives inputs, the network selects the neurons with maximum activity level and maintains that pattern [Erm92]. In this report, as it was explained earlier (see section 2.1), the fanals (that rep-

resent microcolumns in human neocortex) are the elements of the network to perform Winner-take-all. “Current data on the microcolumn indicate that the neurons within the microcolumn receive common inputs, have common outputs, are interconnected, and may well constitute a fundamental unit of the cerebral cortex” [CBP⁺05]. Considering above discussions, a fanal is modeled based on the McCulloch-Pitts model [MP43]. In Fig. 2.6, the fanal model is depicted. It is

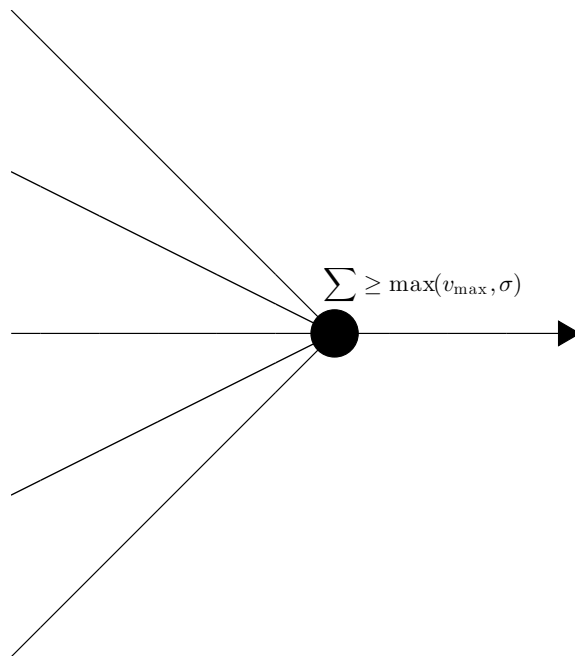


Figure 2.6: A fanal is modeled similar to McCulloch-Pitts model of neuron. It computes the sum of the received signals which is known as activity level. Thereafter, if it is equal to the maximum of the maximum activity level in the network and the threshold, it fires.

similar to the model proposed by McCulloch and Pitts for neurons in [MP43]. A fanal computes the sum of these received signals from other fanals in the network. The sum of received signals is so-called *activity level* of the fanal. A fanal fires if its activity level is equal to $\max(v_{\max}, \sigma)$. v_{\max} is the maximum activity level of a fanal in either a cluster (which the fanal belongs to) or all fanals in the network,

and σ is a threshold similar to the threshold in [MP43]. σ is set either for the fanals of a particular cluster or for all fanals in the network.

In addition to the McCulloch-Pitts model in which the neuron fires when its activity level exceeds a threshold, this model also includes the maximum activity level (in a cluster or in the entire network).

2.4.2 The Algorithm

The network is a multipartite recurrent graph that is comparable to the well known Low Density Parity Check (LDPC) code bipartite graph [Gal62, LC04]. In Fig. 2.7,

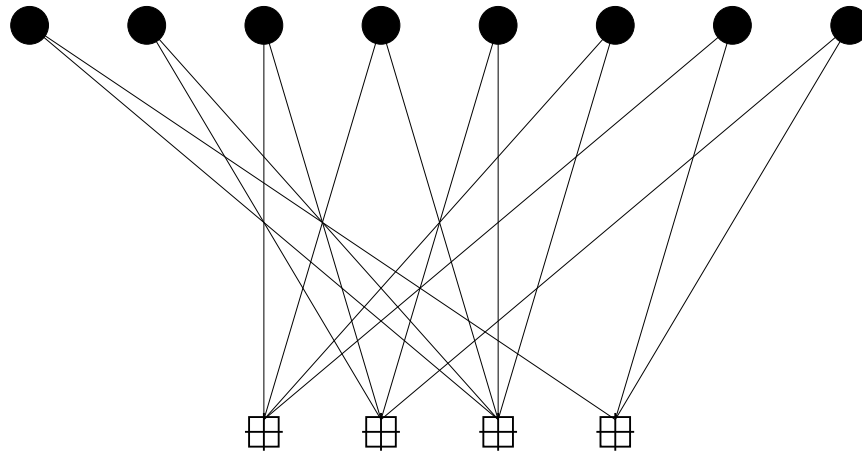


Figure 2.7: A bipartite graph that represents an LDPC code is depicted. The nodes on top are variable nodes and the nodes in the bottom are check nodes. The check nodes are arithmetic nodes (that do not take values) whereas the variable nodes hold the result of iterative computation that is performed in each iteration.

a simple LDPC bipartite graph is depicted where the edges are binary connections constructed according to the parity-check matrix. The advent of bipartite graphs representation of linear codes and of the message-passing decoding algorithms goes back to the seminal work of Gallager [Gal62]. Gallager proposed a simple iterative algorithm (hard information propagation) in which the values of variable nodes are calculated through parity-check constraints [Gal62, LC04]. This algorithm was

later developed to soft information propagation that is known today as Belief Propagation (BP) [Loe04].

The proposed network is a multipartite graph in which connections are constructed through the storage process (see section 2.3). Unlike LDPC codes, a node in the network is both a variable and a check node, i.e., it contributes to both computation and holding the resulting values (activated/deactivated). The principle that is shared by LDPC decoders and the network is the iterative process, that is, message-passing. Like the algorithm given in [Gal62], it is simply computing the values of nodes (they take binary values, 1 or 0, activated or deactivated) in the network through an iterative process. The iterative process is imposed by the fact that the state of each node depends on the state of other nodes in the multipartite graph that may take transitory values in iteration steps. The iteration may continue till the network converges to a steady state (there are situations where the network does not converge). A steady state is where all the activated fanals remain active and all the deactivated fanals remain deactivated when the algorithm passes from one iteration step to the next.

The graph cycles in linear code graphs may increase the number of iterations for which decoder converges whereas the cycles are an essential feature of the proposed network that provides robustness. Therefore, they are avoided in linear codes whereas in the proposed network having more cycles is desirable.

In the proposed network, the retrieval process is: (a) Accumulating received signals from the connected fanals to each one. (b) Applying Winner-take-all (see section 2.4.1) in order to determine the active fanals in the network. (c) repeating steps (a) and (b) iteratively until the network converges or for a limited iteration steps. (d) detect whether the remaining active fanals (after the last iteration step) construct a unique clique while there is only one active fanal in each cluster.

Assume the network has stored a message set \mathcal{M} and is stimulated with a particular input. Now consider the connection matrix w in (2.3) which is four

dimensional. In step (a) the network computes the accumulated received signals from clusters with index $1 \leq i \leq \chi$,

$$v(n_{ij}) \leftarrow \sum_{\substack{i'=1 \\ i' \neq i}}^{\chi} \max_{1 \leq j' \leq \ell} (w_{(i'j')(ij)} v(n_{i'j'})) + \gamma v(n_{ij}) \quad (2.15)$$

where γ is a memory effect that is taken $\gamma = 1$ throughout this report. In words it can be expressed that the activity level of the fanal j in cluster i , $v(n_{ij})$, increments by one unit if there is one or more active fanals in cluster with index i' . In the retrieval procedure given in [GB11], the sum $\sum_{j'=1}^{\ell}$ in

$$v(n_{ij}) \leftarrow \sum_{\substack{i'=1 \\ i' \neq i}}^{\chi} \sum_{j'=1}^{\ell} w_{(i'j')(ij)} v(n_{i'j'}) + \gamma v(n_{ij}) \quad (2.16)$$

takes all the active fanals in a cluster into account. This could be interpreted as there are more than one active fanal in each cluster. The network is expected to converge to a single clique (with only one active fanal in a cluster) after being fed with an input stimulus. This can be directly deduced from the storage procedure (see section 2.3). Therefore, the mentioned summation contradicts the principles of the network. To resolve this issue, that summation is replaced by the $\max(\cdot)$ function in (2.15). It returns 1 if there is one or more fanals in cluster i' that are connected to fanal n_{ij} [GB12].

The defect of having a fanal n_{ij} in (2.16) that receives more than one signal from fanals of cluster i' may have other solutions. Another proposed solution is the normalization of the emitted signals from cluster i' . Thus, instead of sending a single signal to fanal n_{ij} (in (2.15)), the active fanals send a weaker signal each equal to

$$\mu = \frac{1}{\sum_{j'=1}^{\ell} w_{(i'j')(ij)} v(n_{i'j'})}$$

from i' th cluster. The activity level of fanal n_{ij} is then computed as

$$v(n_{ij}) \leftarrow \sum_{i'=1}^{\chi} \sum_{j'=1}^{\ell} \frac{w_{(i'j')(ij)} v(n_{i'j'})}{\mu} + \gamma v(n_{ij}). \quad (2.17)$$

In Fig. 2.8, a situation is illustrated where there are three active fanals in a cluster that each sends a one-third signal. Normalizing the emitted signals in (2.17) turns

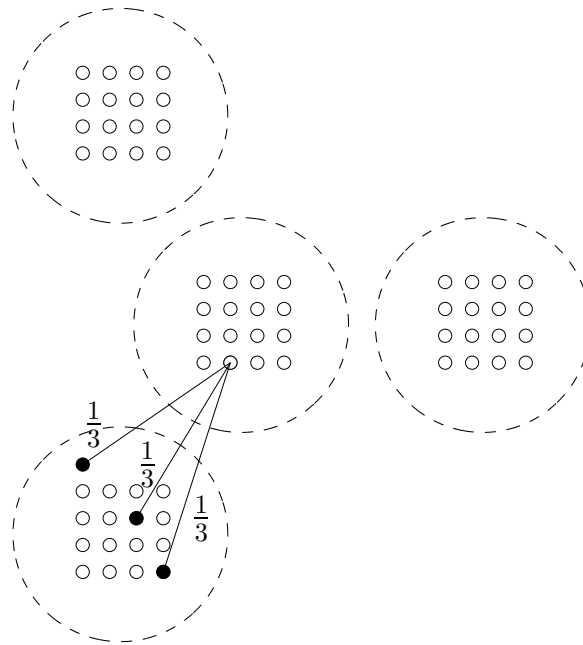


Figure 2.8: A situation is demonstrated where three active fanals in a cluster send signals to a fanal in a different cluster. Since for a clique there is only one active fanal in a cluster the signal strength of each active fanal is normalized to get a total signal strength equal to unity.

out to be an improper solution with an error rate performance that is worse than (2.15). Although (2.17) normalizes the receiving signals from several fanals in a cluster but it may not force the principle of the network (which is a clique may only have maximum one active fanal in a cluster). The activity level of fanal n_{ij} depends on other activated fanals in cluster i' that may have connections belonging to other stored messages.

After the fanals' activity levels are computed in step (a), next step (b) is Winner-take-all rule. The very first step before applying Winner-take-all is obtaining the maximum activity level in the network for the current iteration step. The

maximum activity level in cluster i , $v_{\max,i}$ is given by

$$\forall i, 1 \leq i \leq \chi : v_{\max,i} \leftarrow \max_{1 \leq j \leq \ell} (v(n_{ij}))$$

and finally the absolute maximum activity level in the network is given by

$$v_{\max} \leftarrow \max_{1 \leq i \leq \chi} (v_{\max,i}).$$

Next, Winner-take-all sets the fanals having the maximum activity level, v_{\max} , and above the threshold σ_i to 1. This is formulated as

$$\begin{aligned} &\forall i \text{ and } j, 1 \leq i \leq \chi, 1 \leq j \leq \ell : \\ v(n_{ij}) &= \begin{cases} 1 & \text{if } v(n_{ij}) = v_{\max} \text{ and } v_{\max} \geq \sigma_i, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (2.18)$$

where σ_i is the threshold of a cluster with index i in the network.

2.5 Performance

The storage and retrieval procedures of the proposed neural network memory were explained previously. The network performance in different scenarios is addressed in this section where the retrieval success rate (or correspondingly the retrieval error rate) is the main subject.

The network performs a successful recovery when it converges to a unique clique (corresponding to the stored message) after being fed by a stimulus. The recovery of stored messages in the network may be erroneous. The errors are categorized into two kinds: **(Type I)** The networks converges to a unique clique. This clique either corresponds to another stored message (with higher order) or it may be constructed from other stored messages' connections (it does not represent any of the stored messages). **(Type II)** The network dynamics finalizes towards a steady state where there are more than one active fanal in a cluster. Having more than a single active fanal in a cluster is so-called *ambiguity*. To clarify this point,

an ambiguous situation example where the network is not certain about the right clique is illustrated in Fig. 2.9. Therein, it is assumed that a message of order four (with four sub-messages) is stored. The network is fed with three known sub-messages that activate their corresponding fanals marked with black and they are connected with solid lines. The network dynamics activate two other fanals in a cluster (the cause of ambiguous state in the network) that are connected to the known fanals with dashed lines. In this case, the network can not answer which clique is the correct one.

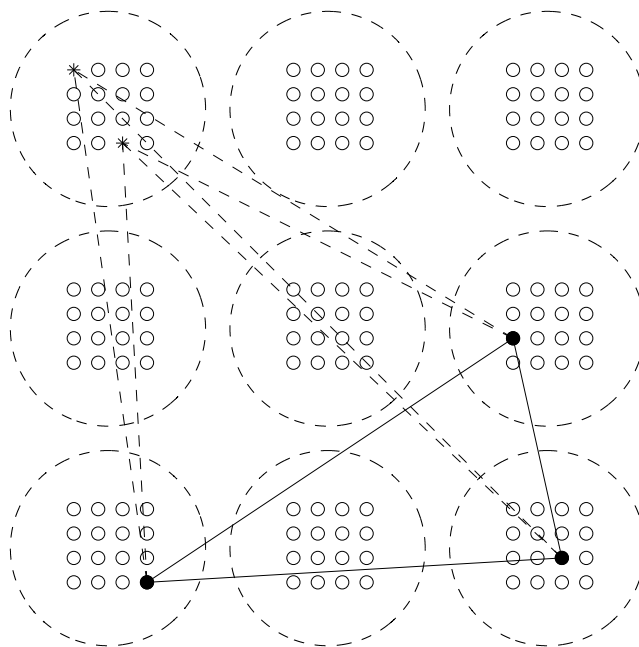


Figure 2.9: An example of ambiguity in the network where a fanal (corresponding to a sub-message) is missing. The ambiguous fanals are connected to the known fanals with dashed lines.

It is also possible that the network does not converge. A special case study is given later in this chapter where the network never converges (see section 2.5.1). In such situations, the network may stop after a pre-defined maximum number of iterations.

Assume the retrieval procedure recovers \hat{m} and the stored message is $m \in \mathcal{M}$. The error is defined as random variable [CT06]

$$E = \begin{cases} 1 & \text{if } m \neq \hat{m}, \\ 0 & \text{if } m = \hat{m}. \end{cases} \quad (2.19)$$

The probability $P_e = \Pr[E = 1] = \Pr[m \neq \hat{m}]$ is the retrieval error probability. Note that the condition, $m \neq \hat{m}$, addresses both Type I and Type II errors in the network. In case a Type II error takes place, the network returns a message/sign (e.g. an empty message) to indicate that it could not give a response. Indeed, where there is an ambiguity in a cluster the network is doubtful about the right clique representing the stored message.

2.5.1 Associative Memory

The most important function of the proposed network is Associative Memory. The network provides a neural memory whose contents can be accessed by known parts of the stored messages. Assume a set of sparse messages of size χ each having c significant sub-messages is stored in the network. A message in the stored set is retrieved by knowing a part of it. This is so-called *partial message* in which a few of the significant sub-messages are missing.

Recovering messages from their corresponding partial messages are considered for two distinct cases: (a) The partial messages consists only of a few known sub-messages and their corresponding positions in the message. This is so-called *blind recovery* since the positions of missing sub-messages in the support of the original message are not known. (b) The partial message consists of a few known sub-messages and the support of the original message is known. Therefore, the network is *a priori* aware of the clusters in which the sub-messages are missing. This is so-called *guided recovery*.

The recovery of the two mentioned types is explained and their corresponding retrieval probabilities are given in the following.

Guided Recovery

In guided recovery, the network is fed with a few known sub-messages (but not all) and the support of a stored message. This facilitates the recovery procedure due to the fact that the network dynamics are *a priori* aware of the cluster indices in which the fanals are missing. This simplifies down the retrieval situation to the network in [GB11] with the same retrieval error rate for the first iteration. Therefore, the retrieval error probability is given by

$$P_e = 1 - (1 - d^{c-c_e})^{(\ell-1)c_e} \quad (2.20)$$

where c_e is the number of missing sub-messages.

Blind Recovery

The network in [GB11] stores messages of order c that are not sparse therefore all clusters are exploited for the storage. In the proposed network, the messages are sparse and the network exploits this: it only stores the significant sub-messages of a message in a clique. Afterwards, when the network recovers a message from its partial message, it has to spot two different types of information for each missing sub-message. These are the cluster index and the fanal that represent the missing sub-message. This is a more complicated retrieval in comparison with the retrieval procedure of the classical network in [GB11].

The network that is stimulated with a partial message for which the support of the original message is unknown has to investigate the entire network. Assume the messages in the stored message set have order c and partial messages have c_e missing clusters each. The sufficient condition for message m to be certainly recovered from its partial message is that there are no fully connected irrelevant fanals to the known fanals. The network dynamics in the first iteration activates those fanals that are connected to all $c - c_e$ known fanals. Thus, if any other

fanals in the network has less than $c - c_e$ connections with the known fanals, the network converges to the right clique in the first iteration.

The probability of having a connection between two fanals in two distinct clusters is approximately equal to the network density $P \approx d$. Therefore, a single connection in the network can be considered as a random variable with Bernoulli distribution whose success probability is d .

Assume a fanal that does not participate in the construction of the sought clique. The probability that this fanal is not connected to all known fanals is now computed. This is the probability that the fanal is connected to any number of known fanals in $\{0, 1, 2, \dots, (c - c_e - 1)\}$ but not all $c - c_e$ fanals. It is computed through binomial distribution as

$$\begin{aligned} P_s &= 1 - \sum_{i=1}^{c-c_e-1} \binom{c-c_e}{i} d^i (1-d)^{(c-c_e-i)} \\ &= 1 - d^{(c-c_e)} \end{aligned}$$

where d is the success probability. Now this is extended to the entire network. The probability of not having an irrelevant fanal in the entire network (comprising the partially known message subject to recovery) is computed through binomial distribution

$$\begin{aligned} P_r &= \sum_{i=1}^{c_e(\ell-1)} \binom{c_e(\ell-1)}{i} P_s^i (1-P_s)^{(c_e(\ell-1)-i)} \\ &+ \sum_{i=1}^{\ell(\chi-c)} \binom{\ell(\chi-c)}{i} P_s^i (1-P_s)^{(\ell(\chi-c)-i)} \\ &= P_s^{[c_e(\ell-1)+\ell(\chi-c)]} = (1 - d^{(c-c_e)})^{[c_e(\ell-1)+\ell(\chi-c)]} \end{aligned}$$

where the first sum computes the probability of not having any irrelevant fanal from the clusters in which the missing sub-messages activate their corresponding fanals. There are $c_e(\ell - 1)$ irrelevant fanals in those clusters. The second sum is the probability of not having any irrelevant fanals from other clusters that do not participate in the message clique. There are $\ell(\chi - c)$ such fanals. The probability

of error for the first iteration is then:

$$P_e = 1 - P_r = 1 - (1 - d^{(c-c_e)})^{[c_e(\ell-1)+\ell(x-c)]}. \quad (2.21)$$

In this recovery procedure, the network retrieves the messages with only one iteration since more iterations can not improve the performance. To explain the

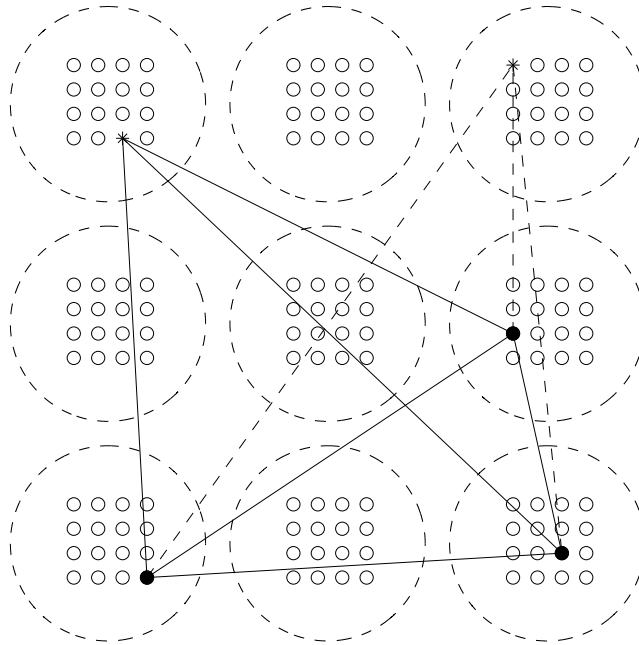


Figure 2.10: A situation in the network that cause fanal activity oscillation. This is when the fanals indicated with star markers are activated in odd iteration steps and deactivated in even iteration steps.

reason that results in a single iteration retrieval for blind recovery, an example is given in Fig. 2.10. The network is stimulated with a partial message that activates the fanals indicated with black markers. The network dynamics activate two other fanals in the first iteration that are indicated with star markers. The message clique corresponding to the stored message is depicted with solid lines and includes one of the activated fanals indicated with a star in the first iteration. The second star marker is an activated fanal that belongs to another stored message. Assume the

network passes to the second iteration step. At the end of this step, the fanals indicated with star markers are deactivated and only the fanals that were initially activated by the stimulus (the partial message) remain active. Thus, in case the recovery passes to the next iteration steps the network oscillates between two distinct states: (a) only the partial message fanals are activated. (b) the partial message fanals along with the two other fanals (that are connected to all partial message fanals) indicated with star marker are activated. Therefore, the network never converges and it does not recover the stored message. This is so-called **Type III** error that is the third kind of introduced errors along with Type I and Type II errors. Assume that in Fig. 2.10, the network has stored message

$$m = [4, -, 7, -, -, 5, 3, -, -].$$

Fanal (9, 13) is a false fanal that is connected to all three known fanals (1, 4), (3, 7) and (6, 5). At the end of the first iteration step, the activity levels of fanals are $v(n_{1,4}) = v(n_{3,7}) = v(n_{6,5}) = v(n_{7,3}) = v(n_{9,13}) = 3$. Therefore, all these fanals remain activated and passed to the second iteration step. At the end of the second iteration step the activity levels are $v(n_{1,4}) = v(n_{3,7}) = v(n_{6,5}) = 5$ and $v(n_{7,3}) = v(n_{9,13}) = 4$. Thus, fanals $v(n_{7,3})$ and $v(n_{9,13})$ are deactivated. This resets the network to the initial state before the first iteration. In Fig. 2.11, Type III error rate is depicted along with the total error rate for a network that has $\chi = 100$ clusters and $\ell = 128$ fanals in each cluster⁷. It can be observed that more than 30% of the errors after the second iteration are Type III. It is also observed through experiments that in blind recovery several iterations do not improve the performance.

In Fig. 2.12 the network performance is illustrated through simulation where it is used as an associative memory. The network has $\chi = 100$ and $\ell = 64$ fanals in each cluster. The messages have $c = 12$ significant sub-messages and they are

⁷This phenomenon is seen easier when ℓ is large and this is why $\ell = 128$ unlike other simulations that have $\ell = 64$.

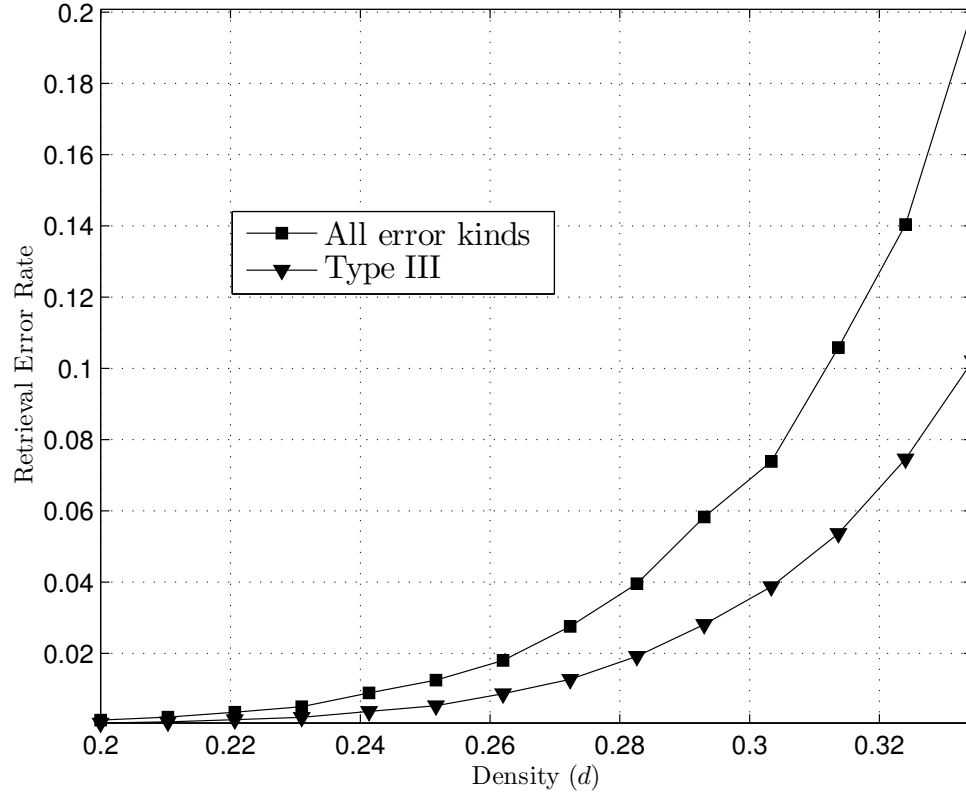


Figure 2.11: Type III error rate is compared to total error rate for a network with $\chi = 100$ clusters and $\ell = 128$ fanals in each cluster. The partial messages have two unknown sub-messages $c_e = 2$. It is observed that over 30% are Type III errors.

recovered from partial messages with $c_e = 3$ unknown sub-messages. The solid line is the theoretical blind retrieval error given by (2.21) and the dark square markers indicate the corresponding simulation results for a single iteration retrieval. The dashed line is the theoretical guided retrieval error rate in (2.20) and dark rhombus markers are the corresponding single iteration simulation results. The simulation results of a four iteration recovery is shown with dark triangle that demonstrates a significant improvement with respect to the single iteration retrieval.

Variable Order Messages

Assume the network stores a set of variable order i.i.d. messages. A message has an order c that is uniformly distributed between c_{\min} and c_{\max} , $c_{\min} \leq c \leq c_{\max}$.

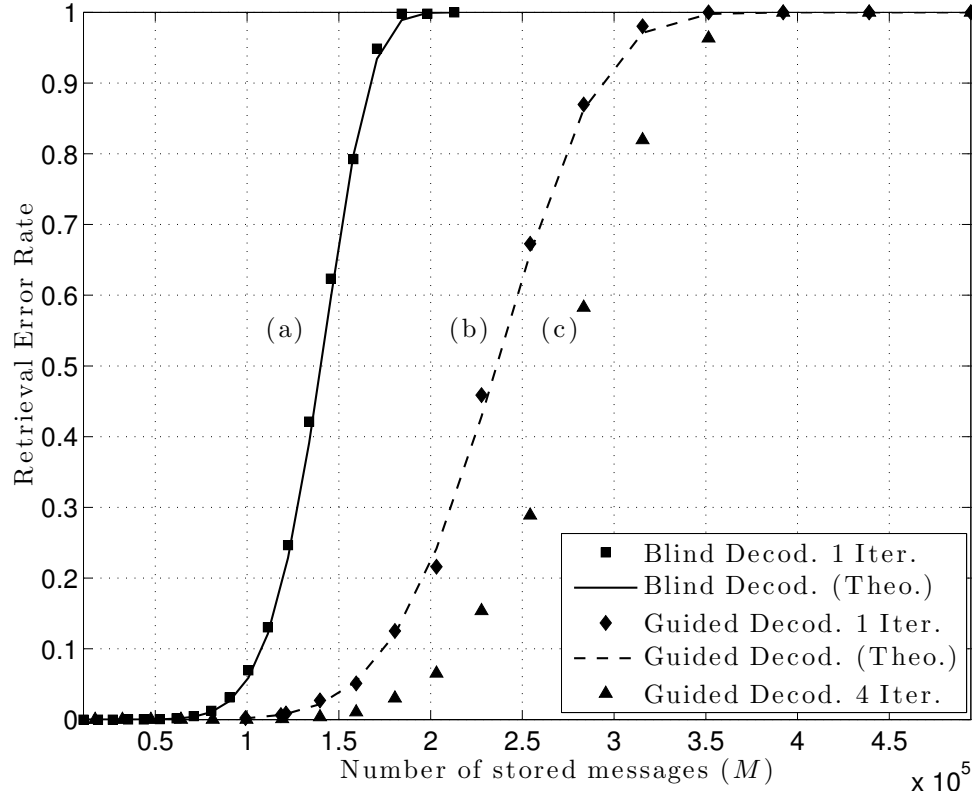


Figure 2.12: The retrieval error rate of both blind and guided recovery of M i.i.d. messages. The network has $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The partial messages have $c_e = 3$ unknown sub-messages.

The blind recovery retrieval error rate is then given by

$$P_e = 1 - \frac{1}{c_{\max} - c_{\min} + 1} \sum_{c=c_{\min}}^{c_{\max}} (1 - d^{(1-\epsilon)c})^{[(\ell-1)\epsilon c + (\chi-c)\ell]} \quad (2.22)$$

where $\epsilon = c_e/c$ is the ratio of the number of unknown sub-messages to the message order. In Fig. 2.13, the blind retrieval error rate for a network with $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster is shown. There are two distinct simulations for $6 \leq c \leq 18$ and $12 \leq c \leq 24$. It is observed that the simulation results depicted with dark markers lie on the theoretical curves given by (2.22) (the solid and dashed lines).

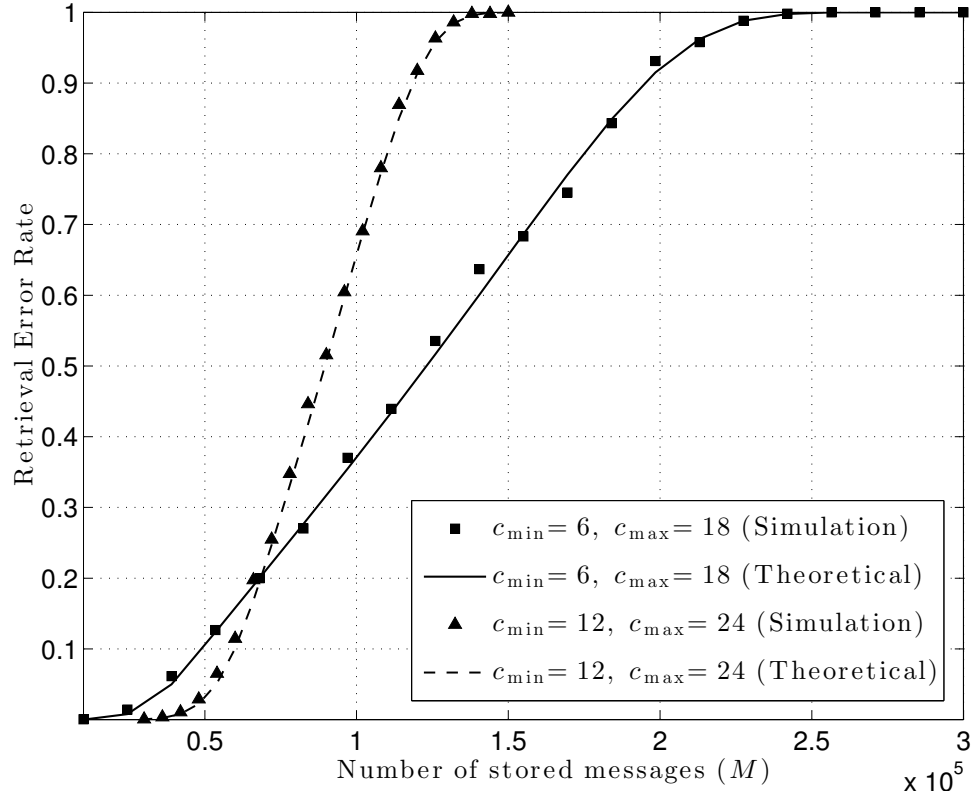


Figure 2.13: The retrieval error rate of both blind and guided recovery of M i.i.d. messages. The network has $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The partial messages have $c_e = c/4$ unknown sub-messages, i.e., $\epsilon = 0.25$.

2.5.2 Blurred Messages

The proposed network exploits cliques that are redundant structures to store messages, therefore, the error correction of distorted messages such as noisy or permuted messages is possible.

A message from a stored message set (in the network) in which the sub-messages are slightly changed due to a sort of disturbance is so-called a *blurred message*. The sub-messages of a blurred message may not be equal to those that are stored in the original message but they are close (e.g. with respect to Euclidean distance measure). Assume the network has *a priori* knowledge of the disturbance (e.g. the disturbance is a Gaussian noise with variance 1). To recover the original message, the network activates a few hypothetical fanals (depending on

the disturbance) that may be close to the correct ones. Thereafter, the network performs the retrieval procedure that may converge to a unique clique that is associated with the original message.

In order to demonstrate this capability, the problem of recovering permuted messages is considered here. For example “intelligence” is a twelve letter English word that may be received either as “nietllgineec” or “etnleilegcn”. If the network is able to recognize “intelligence” from these pairwise permuted versions, it is also able to correct less disturbed ones such as “intleligece” or “intelligene”. The human brain is reputed for having the capability of recognizing simple permutations in the words [GW04].

Having *a priori* knowledge that the disturbed sub-messages are close to the original message implies that the network needs to be stimulated by activating the corresponding close fanals. In the given example above, a limited permutation is considered where pairs of letters in a word are randomly permuted. To recover the original message it is required to activate the corresponding fanals of each sub-message and its neighbour (left and right) sub-messages. Therefore, in each cluster there are three fanals that are initially activated. The retrieval error probability is the probability that at least one of the permuted fanals (that corresponds to a sub-message) constructs a false clique in the network. This is given by

$$P_e = 1 - (1 - d^{c-1})^{2c}. \quad (2.23)$$

In Fig. 2.14 the network with $\chi = 100$ clusters and $\ell = 64$ fanals in each clusters recovers permuted messages after the first and sixth iteration steps. It can be observed from the illustrated simulations and theoretical (2.23) that the given retrieval error probability may be achieved after several iteration steps.

Assume an extreme situation where all fanals are activated to find the stored message. This occurs when the close fanals (to the stored fanals) may not be explicitly determined or all fanals may equi-probably participate in the stored clique.

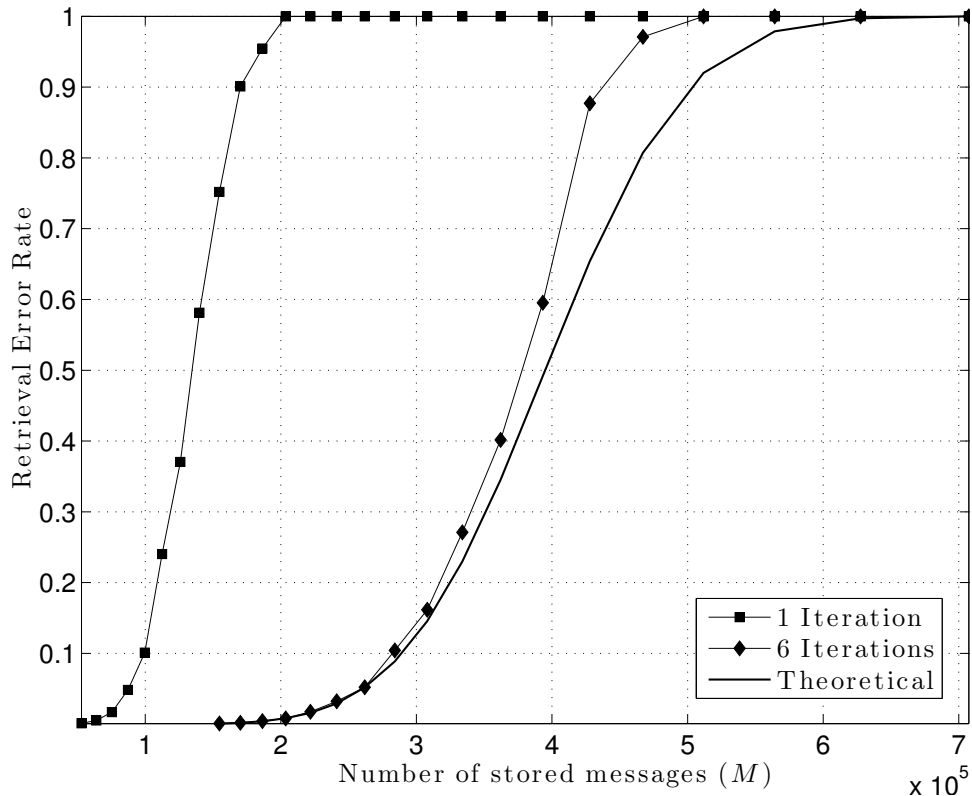


Figure 2.14: The retrieval error rate of messages that are distorted by permutation. The network has $\chi = 100$ clusters and $\ell = 64$ in each cluster. The stored i.i.d. messages have order $c = 12$.

The retrieval error probability is given by

$$P_e = 1 - (1 - d^{c-1})^{c(c-1)}. \quad (2.24)$$

2.5.3 Set Implementation

In this situation, the network stores a set of messages (one at a time or the whole set at once) and it can be asked afterwards if a hypothetical message (the message to be verified whether it is stored) is stored.

The query procedure (i.e. algorithm) is the one elaborated in section 2.4.2 exploiting (2.18) and (2.15). The threshold in (2.18) is equal to the number of significant sub-messages, $1 \leq i \leq \chi$, $\sigma = \sigma_i = c$. The under test message activates its corresponding fanals. If the message exists as a stored message, the

network converges to the fed message. If the network does not return the fed message, the message has not been stored yet.

As said before, the network always converges if the fed message is already stored in the network. This stems from the retrieval algorithm in which, if all of the fanals of a stored message are activated, there is no other fanals in the network with an activity level higher than the activated fanals. Assume a message of order c and the memory effect $\gamma = 1$. The fanals of this message in the network have activity level equal to c whereas any other competing fanal can have maximum score $c - 1$, since it can not receive any signal from its own cluster. The correct message fanals receive $c - 1$ signals from other activated fanals (associated with the message) in addition to the memory effect that results in an activity level equal to c . Therefore, a mistake is not possible if the network is asked to verify a stored message. It can be concluded that one iteration is sufficient and necessary but not more.

Although recovering back a stored message in the network is certain, there are other cases for which the network fails. This is when the network is fed with a message that is not stored but it converges and return the message. This may be mistakenly interpreted as the message was stored. This phenomenon is related to false cliques. These are cliques that are not associated with any of the stored messages but are created from the combination of two or more stored messages' connections. Therefore, by assuming again that the connections in the network are asymptotically i.i.d. and uniformly distributed, the probability of detecting a false clique (associate with a hypothetical message that was never stored) is

$$P_{\text{False-Clique}} = d^{\frac{c(c-1)}{2}} = \left(1 - \left(1 - \frac{c(c-1)}{\chi(\chi-1)\ell^2} \right)^M \right)^{\frac{c(c-1)}{2}} \quad (2.25)$$

where c is the message order of the hypothetical message. In Fig. 2.15, the simulation results are illustrated along with the theoretical error probability given by (2.25). In this simulation, the network has $\chi = 100$ clusters with $\ell = 64$ fanals

in each cluster. The densities from which the network stores more messages than limited by the efficiency-1 diversity (see section 2.3) is indicated with “ \times ” and “ $+$ ”. These points are computed through (2.9). It can be observed that even though the network is overloaded, the probability of false detection is very low. From this, it can be concluded that the network can be employed for set implementation reliably with efficiencies larger than one: $\eta > 1$.

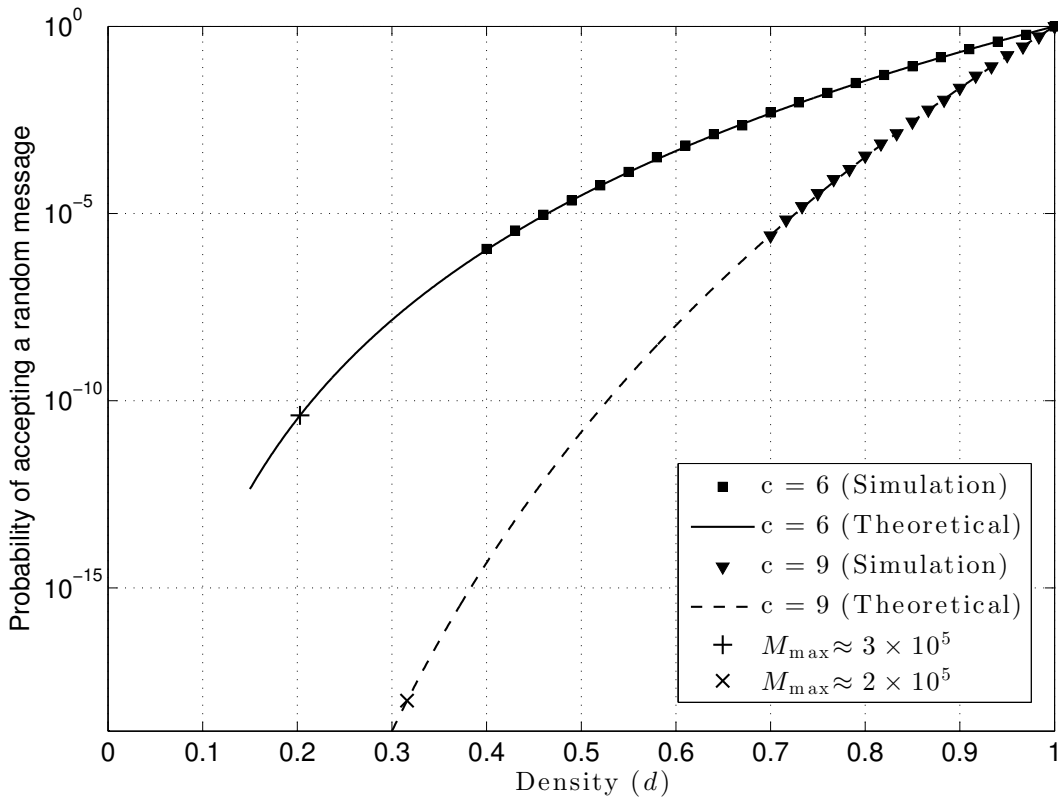


Figure 2.15: The set implementation error rate as a function of the density of the network composed of $\chi = 100$ clusters of $\ell = 64$ fanals each, with cliques of order $c = 6$ or 9 . Densities corresponding to efficiency-1 diversities (M_{\max}) are also indicated.

2.6 Optimal Network

The network and the message set need to be designed in order to establish an efficient and optimal neural. The optimality can be based on different cost functions.

The network is optimized with respect to two cost functions: the retrieval error rate and the network diversity. In the previous sections the given equations have established the relations among the network message set parameters and these cost functions. The network parameters are cluster space size χ and number of fanals in each cluster ℓ . The parameters of stored the message set are the minimum and maximum message orders (c_{\min} and c_{\max}) and number of stored messages M . These two sets of parameters are related by retrieval error probability, density and diversity equations.

2.6.1 Optimal Constant Message Order

The network with cluster space size χ and ℓ fanals in each cluster stores a set of messages with constant order c . As a design parameter, the optimal message order c is computed to maximise the number of stored messages.

The blind retrieval error probability in (2.21) is rewritten as

$$\begin{aligned} P_r &= (1 - d^{(c-c_e)})^{[\chi^\ell - (\epsilon + (1-\epsilon)\ell)c]} \\ &\approx (1 - d^{(1-\epsilon)c})^{\chi^\ell} \end{aligned} \quad (2.26)$$

where $\epsilon = c_e/c$. The network density is approximately computed as

$$d \approx (1 - P_r^{\frac{1}{\chi^\ell}})^{\frac{1}{(1-\epsilon)c}}.$$

For low density values $d \ll 1$ in (2.6), the number of stored messages is approximately given by

$$M \approx d^{\frac{\chi(\chi-1)\ell^2}{c(c-1)}}.$$

After taking the derivative of M with respect to c , the approximate optimal message order for a specific error rate P_0 is

$$c_{\text{opt}} \approx \frac{\log\left(\frac{\chi^\ell}{P_0}\right)}{2(1-\epsilon)}. \quad (2.27)$$

In Fig. 2.16, the number of messages versus the message order c is illustrated. It

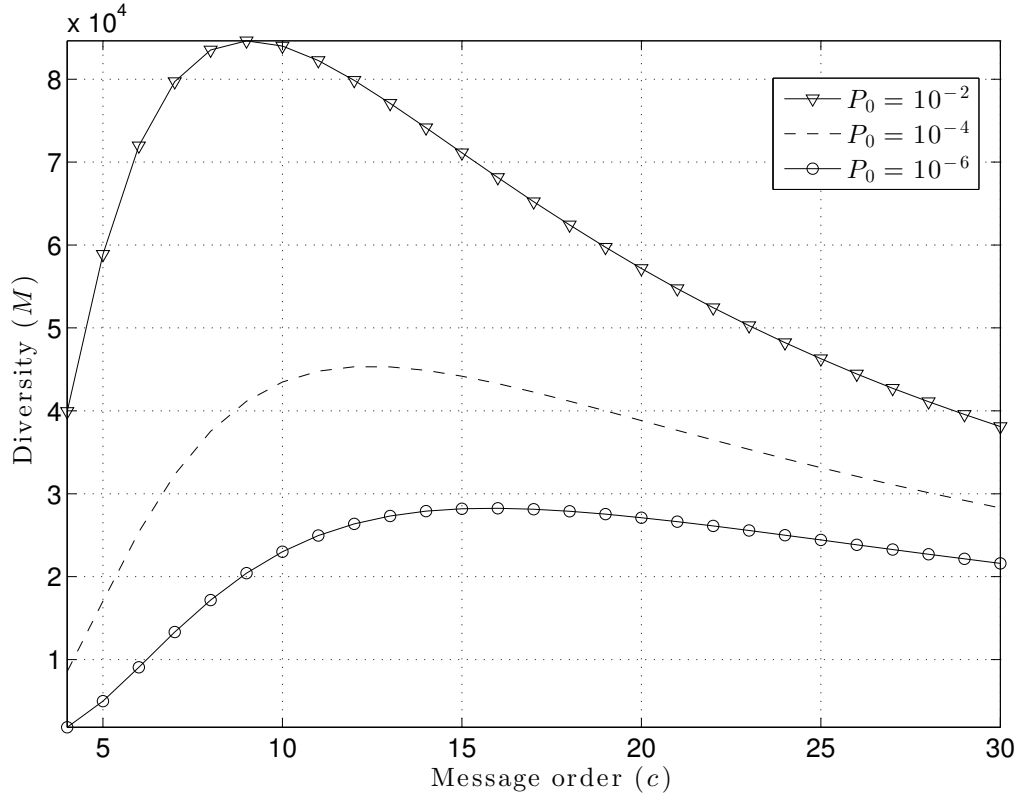


Figure 2.16: The number of stored messages is depicted for the network with $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The curves are given for several blind retrieval error probabilities where $\epsilon = 0.25$.

is observed that the number of stored messages reaches a maximum for a specific c that is given approximately by (2.27). The optimal message order can be used as a rule-of-thumb to design networks that store variable order messages by choosing the average order close to (2.27).

2.6.2 Optimal Density

The optimal density refers to a density for which the network exploits the connections efficiently. This is when the connections bear the maximum amount of information. The connections of the network take binary values and each can be considered as a Bernoulli random variable. Bernoulli random variables bear the maximum amount of information when their success probability is $\frac{1}{2}$. So, a network

that exploits efficiently its connections is the one with density $d = 0.5$. Therefore, the connection matrix in (2.3) does not need to be compressed when it is being transferred or stored on a device.

2.6.3 Optimal Variable Message Order

A network that stores variable order messages is more complicated to optimize although it has more capabilities (storage of variable order messages). To optimize the network two criteria are considered: the density $d = 0.5$ and the recovery error rate. It is aimed to find c_{\min} and c_{\max} for which the network has the least (or very low) retrieval error probability while $d = 0.5$. To achieve this goal, the network density in (2.11) where $\frac{1}{\chi(\chi-1)} \ll \ell^2$ is rewritten as

$$d \approx 1 - \left(1 - \frac{1}{\ell^2}\right)^{\sum_{k=1}^M \frac{c_k(c_k-1)}{\chi(\chi-1)}} \quad (2.28)$$

where M is large and c_k ($c_{\min} \leq c_k \leq c_{\max}$) is the k th message order. The message orders c_k are assumed to be uniformly distributed between c_{\min} and c_{\max} . The number of messages with order c is $M_c = \frac{M}{c_{\max} - c_{\min} + 1}$ (based on an axiom of probability theory described in [Pap91, p. 11]) that results in

$$\begin{aligned} d &\approx 1 - \left(1 - \frac{1}{\ell^2}\right)^{\sum_{k=1}^M \frac{c_k(c_k-1)}{\chi(\chi-1)}} \\ &= 1 - \left(1 - \frac{1}{\ell^2}\right)^{M_c \sum_{c=c_{\min}}^{c_{\max}} \frac{c(c-1)}{\chi(\chi-1)}} \\ &= 1 - \left(1 - \frac{1}{\ell^2}\right)^{M \frac{E\{(c-E\{c\})^2\} + E\{c\}^2 - E\{c\}}{\chi(\chi-1)}}} \end{aligned} \quad (2.29)$$

where \mathbf{c} (in bold) is the message order random variable (uniformly distributed).

Mean of random variable \mathbf{c} is

$$E\{\mathbf{c}\} = \frac{c_{\max} + c_{\min}}{2}$$

and its variance is

$$E\{(\mathbf{c} - E\{\mathbf{c}\})^2\} = \frac{(c_{\max} - c_{\min} + 1)^2 - 1}{12}.$$

In Fig. 2.17, the network is optimized for the maximum connection entropy, that is, for $d = 0.5$. It can be observed that there may be several combinations of c_{\min} and c_{\max} for which the network holds $d = 0.5$ when it stores a certain number of messages. In practice one should choose the best among these choices. A criterion may be the error rate for which each application requires a particular error rate. For example, image processing applications can tolerate high error rates while telecommunication systems generally need an error rate lower than 10^{-4} .

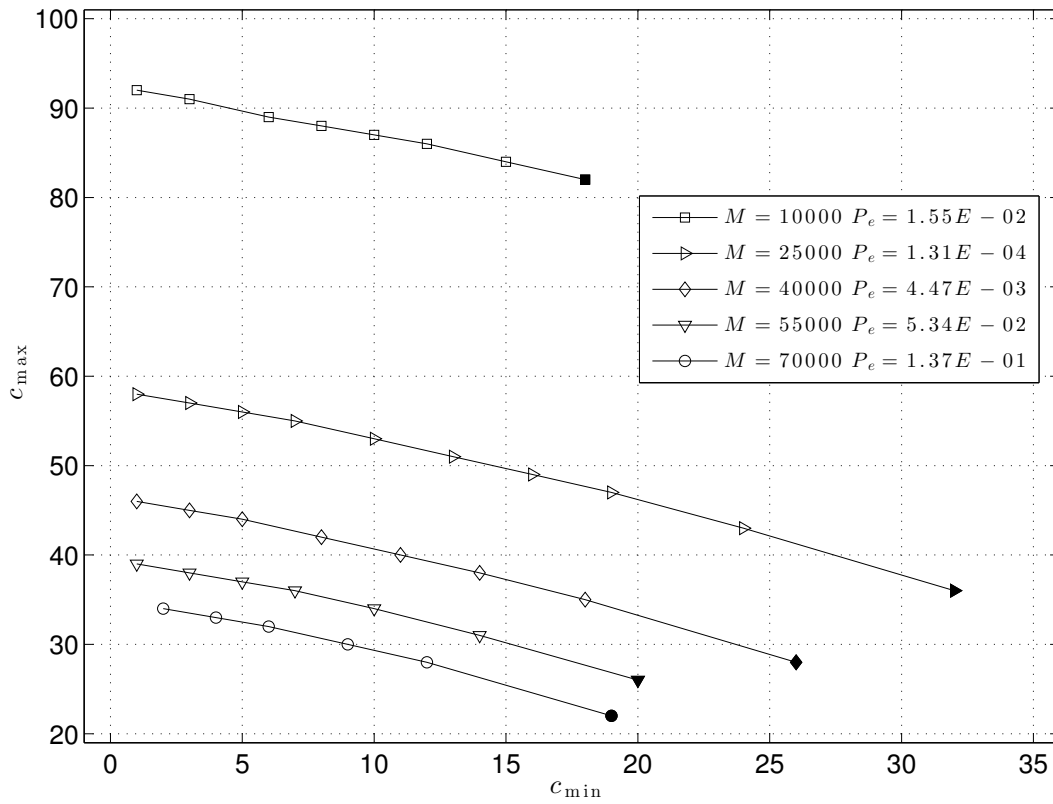


Figure 2.17: The message order pairs are depicted for the network with $d = 0.5$, $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster. The curves are given for several blind retrieval error probabilities where $\epsilon = 0.25$. The least error probabilities are given in the legend and their corresponding pair has filled marker.

Chapter 3

Information-Theoretic Analysis of the Neural Network Memory

Information theory is strongly related to probability theory. It was introduced first by Claude E. Shannon in 1940's to quantify information processed by a system, for which the input and output information can be modeled by probability density functions. Shannon introduced a metric, namely *entropy*, defined as a random variable \mathbf{x} : $\sum_{x \in \Omega} -P(x) \log P(x)$ where $P(x)$ is the probability of outcome x in the probability sample space Ω . Entropy is a real and positive value that measures the amount of randomness, i.e., that reaches its maximum value when \mathbf{x} is uniformly distributed (which represents maximum randomness) [Sha48, CT06]. Shannon established the separate source and channel coding strategies. He considered the source coding (compression) and the channel coding (error correction) as separate blocks: information is first processed by a source coding block to remove useless redundancy. The result is then processed by a channel coding block to embed smart redundancy. This redundancy helps the receiver at recovering the original information from the distorted received information.

The situation in the brain is comparable: external stimuli coming from the outside world are compressed through successive processing layers (such as in the

visual cortex) and then possibly stored with some added redundancy in order to ensure robustness and durability. This flow of neural signals that convey information (which results in learning, moving a muscle, etc.) can be measured [BT99], but the information-theoretic study of the biological neural systems such as the human brain is not really the subject of this thesis. This chapter is focused on artificial neural networks.

The neural memory proposed in chapter 2 will be discussed exploiting the information theory framework, that is, the storage and recovery of messages from their partially known or distorted messages will be studied from source and channel coding aspects. In particular, the last section is dedicated to information-theoretic constraints on the proposed network.

3.1 Analysis of the Message Source

The very first step toward the information theoretic analysis of the proposed neural network is the investigation of the message source. Throughout this report, it is assumed that the message source is i.i.d. and uniformly distributed. Such a source has the maximum entropy and therefore is intrinsically compressed. Moreover, this assumption simplifies the study of the network without loss of generality. Once the network is mathematically formalized for the i.i.d and uniformly distributed source of messages, it can be extended to other types of sources (by knowing their probability density function).

In chapter 2, the retrieval error probability of the network was formulated in (2.20) and (2.21) for the first iteration. These error probability equations are upper bounds on the performance of the network, that implies that the network may have a lower error rate with more than one retrieval iteration step. In this section, a lower bound on the retrieval error probability is computed to demonstrate that even if the network stores two messages, the error probability may not be zero.

The lower bound is computed based on the typical sequences fundamentals [CT06, Chapter 3].

Assume the network tries to retrieve the stored messages from their corresponding partial messages in either guided recovery or blind recovery (section 2.5.1) schemes. The network stores two messages of the same order c . The probability of these two messages having r distinct sub-messages is equal to

$$\binom{c}{r} (1 - 1/\ell)^r (1/\ell)^{(c-r)}$$

where $1 < r < c$. Although the two messages are distinct and can be retrieved, there is a situation in which none of them can be recovered. Assume the network retrieves one of the messages when being fed with a partial message. The partial message has c_e missing sub-messages that are chosen uniformly random among c sub-messages of either of the two messages. In case the r non-identical sub-messages are missing, an ambiguity situation occurs (see section 2.5) that even a genie decoder is not able to resolve. The probability of having the r non-identical sub-messages ($1 < r < c$) missing in a partial message is equal to

$$\epsilon^r (1 - \epsilon)^{(c-r)}$$

where $\epsilon = c_e/c$. The probability P_L that the network is not able to retrieve a stored message from its corresponding partial message is then lower bounded as

$$\binom{c}{r} (1 - 1/\ell)^r (1/\ell)^{(c-r)} \epsilon^r (1 - \epsilon)^{(c-r)} < P_L.$$

This trivial lower bound (regardless of the number of iterations) is imposed by the message source and the network architecture.

For example consider two messages that are drawn from $\mathcal{A} = \{1, 2, \dots, 128\}$ and stored in a network with eight clusters.

$$m_1 = [-, 12, -, 14, -, 78, 35, -]$$

and

$$m_2 = [-, 15, -, 14, -, 78, 35, -].$$

Consider that the first non-blank sub-message in m_1 is randomly missing ($c_e = 1$). This operation can be represented as an element-wise multiplication of two vectors m_1 and

$$O = [-, 0, -, 1, -, 1, 1, -]$$

where 0 represents the missing sub-message. The partial message is obtained as

$$m_{\text{par}} = m_1 \odot O = [-, 0, -, 14, -, 78, 35, -].$$

Message m_1 can not be retrieved from this partial message, because the same operation on m_2 gives the same result.

3.2 Channel Coding

Channel coding is the technique of adding smart redundancy to information in order to preserve the original information where it is exposed to distortions. A common application of channel coding is data protection in communication systems where a transmitter sends information to a receiver over a noisy channel. Transmission over a noisy communication channel may result in the erroneous reception of information at the receiver. Shannon [Sha48] and Hamming [Ham50] separately addressed this problem by proposing error correction codes to protect the transmitted information.

Definition 5. A code (M, c) of length c and cardinality M over an alphabet \mathcal{A} is a collection of M elements from \mathcal{A}^c . The elements of the code are called *codewords* [RU08, LC04].

Shannon showed that there is a code through which a system can reach a maximum communication rate known as channel capacity [Sha48]. Since then, researchers looked for such codes to achieve channel capacity. Turbo code [BG96] and LDPC code [Gal62] are two examples of linear codes getting close to Shannon limits. Although the formal definition of a code in [RU08, LC04] does not impose any constraint on the decoding and encoding procedures, these are of great

importance in practice. Encoding is the mapping of an input information to its corresponding codeword and decoding is retrieving the original input information from the distorted codeword. Since in practice an efficient code is enormously large (M is large), encoding and decoding algorithms are the main concern in coding theory (along with error correction capabilities of codes). There are not many successful coding schemes that comply both encoding/decoding and error correction capability requirements. Turbo and LDPC codes may be the master pieces of this never ending quest until now. These are linear codes that rely on parity check matrices. They are also known as generator matrices which are used to generate codewords. Properties of a code including error detection and correction capabilities, depend on codewords and computing the generator matrix from the codewords is formidable and non-trivial. Consequently finding such matrices that generate desirable codes¹ is difficult.

The classical network in [GB11] and the proposed network in this report attack the channel coding differently. The neural network provides a memory to store the codewords and a practical decoding algorithm. Unlike linear codes, the codewords are not extracted from a generator matrix. The network is capable of storing an arbitrary set of codewords that may or may not be linearly dependent with respect to each other, which is another degree of freedom. Therefore, the designer is able to choose each codeword (theoretically) carefully depending on system requirements.

The code rate of the system depends on the number of messages (i.e. codewords) which the network is able to store and recover. It is

$$R = \frac{\log_{\ell}(M)}{c} \quad (3.1)$$

where M is the cardinality of the code (it is also defined in chapter 2 as the number of stored messages), ℓ is the number of fanals in each network cluster and c is the

¹Desirable codes may be described by various metrics such as minimum Hamming distance [LC04, Ham50].

code block-length (i.e. message order).

There are two main differences between the network in [GB11] and the proposed network when they are used for channel coding applications. The number of codewords of a linear code is remarkably large in comparison with the diversity of the neural network in [GB11]. The proposed network (in this report) has a larger diversity (see section 2.3) than the classical one, which helps the network diversity get closer to the cardinality of linear codes. The second advantage of the proposed network is the storage of codewords with different lengths (i.e. variable message orders). This makes the system able to exploit non-block codes (where the codewords have different lengths). The system may also implement a multi-rate coding system with a variable coding rate depending on the communication channel quality.

3.3 Source-Channel Coding

In chapter 2, it was briefly mentioned that the network is capable of both compression and protection of the stored messages. This is known as source-channel coding where input information is compressed to remove its redundancy and then a smart redundancy is added to protect the compressed information against distortions. Consider the set-implementation application of the network in which the message support set and correspondingly the active clusters are *a priori* known.

It was observed in set-implementation (see section 2.5.3) that the false detection rate is very low even for network efficiency $\eta > 1$. A network efficiency larger than unity implies that the network stores more information bits than its maximum available bits Q (equation (2.4) in section 2.3). The compression and error protection features of the network allow a low false detection rate when $\eta > 1$. This is a kind of joint source-channel coding similar to the classical method in [LC04, CT06] that is elaborated in the following.

A message of order c is mapped to a clique with $\binom{c}{2} = c(c-1)/2$ binary connections. The message contains $c \log_2 \ell$ bits. Therefore, to compress a single message at the storage, inequality

$$\frac{c-1}{2} < \log_2 \ell$$

must be held. For example, previously in chapter 2 a simulation result was given for a network with $\ell = 64$ that stores messages of order $c = 12$. By substituting the network parameters in the inequality, it is found that the network satisfies the inequality, $(12-1)/2 < \log_2 64$. The network compresses further the messages through sharing connections. For example consider messages

$$m_1 = [-, 12, -, 14, -, 78, 35, -]$$

and

$$m_2 = [-, 15, -, 14, -, 78, 35, -]$$

that share $\binom{3}{2} = 3$ connections in their corresponding cliques. The total number of connections to store the two messages is equal to $2 \times \binom{4}{2} - \binom{3}{2} = 9$ instead of 12 (when they are stored separately as single messages). The above example simply shows that a set of stored messages with some equal sub-messages requires a fewer number of connections to be represented in the network. This feature, when the network stores a large number of i.i.d. and uniformly distributed messages, results in compression.

A clique has a redundant structure, i.e., the c active fanals belonging to a message are linked through $\frac{c(c-1)}{2}$ connections whereas $\lfloor \frac{c+1}{2} \rfloor$ may be enough to connect them together² (the minimal number of required connections to represent a message). Therefore, due to such redundancy feature, a clique in [GB11] is considered as a *clique code* (similar to a codeword) that protects the message. The classical codes introduced by Hamming are characterized by a well known

² $\lfloor q \rfloor$ is the largest integer not greater than q .

parameter, namely *minimum Hamming distance*. It is the minimum number of non-identical elements in two distinct codewords of a code. A larger Hamming distance implies a stronger code (in terms of error detection and correction). Hamming distance is the minimum Euclidean distance between two codewords and the network does not have an algebraic topology. Therefore, it does not characterize the clique codes. Assume two cliques that only have one different sub-message. The minimum distance is defined as the number of different edges between these two cliques [GB11], which is

$$d_{\min} = 2(c - 1) \quad (3.2)$$

when the cliques have order c . In [GB11], the authors defined the coding rate R as the ratio of minimum number of edges to associate a set of nodes and the number of edges in a clique which is $R = \frac{1}{c-1}$. The coding rate is an important parameter in coding theory. It quantifies the increased redundancy in order to provide error correction and detection capability. It is therefore desired that the code has a coding rate close to unity, i.e., having less redundancy as possible. This is further used to derive a parameter namely the merit factor $F = Rd_{\min}$. In the proposed network and in the one in [GB11], F is equal to 2, indicating that clique codes are good codes [GB11].

3.4 Communication Channel Model

A memory is expected to recover exactly what has been stored in it. This however does not happen in real-world applications. Memory contents change throughout time due to physical changes in their semiconductor substructure, electromagnetic interference, solar activities, etc. The proposed neural network memory like other types of memories is not error-free. Therefore, a part of stored information may not be recovered (see section 3.1). Erroneous recovery imposes limits on the neural network quality which is highly dependent on the number of stored messages M and

the number of unknown sub-messages at the input. From the error probabilities computed in chapter 2, the maximum number of stored messages can be estimated when the error rate is below a certain limit. Although this is a very useful rule-of-thumb, it may not be a precise criterion, because the error rate limit may change from one to another application. Information-theoretic analysis is an approach that could give rigorous limits on the proposed network rather than vague limits based on intuition. It has been previously considered in [SSP96, AM89] for neural network memories. In the following, information theory is exploited to compute a set of constraints on the proposed network when it is used as an auto-associative memory.

Unreliability in memories can be modeled as a communication channel for which there are known information-theoretic analysis. A communication channel is a probabilistic model to compute information loss in systems in which information is distorted randomly (that is, when the distortions can not be modeled as the result of a deterministic alteration of input and output information). In Fig. 3.1, the noisy communication system model in [PS07, Hay09] is illustrated. The set of messages (source) passes through the neural network memory (noisy communication channel) and the recovered messages are the distorted set of stored messages (sink) ³.

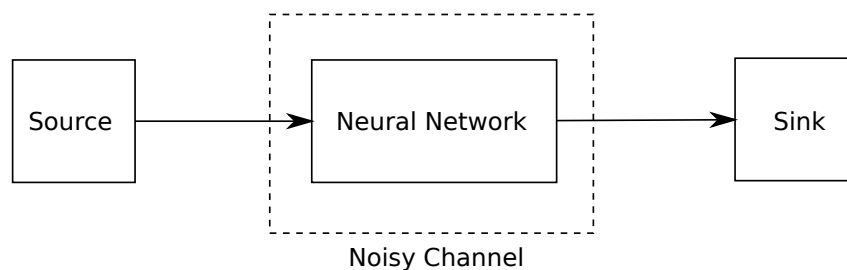


Figure 3.1: Communication Channel Model

In Fig. 3.2, the detailed flow of information in the system is depicted. The time-delay block indicates the separation in time of the recovery and storage in time

³The network including all storage and recovery procedures is considered as a black box.

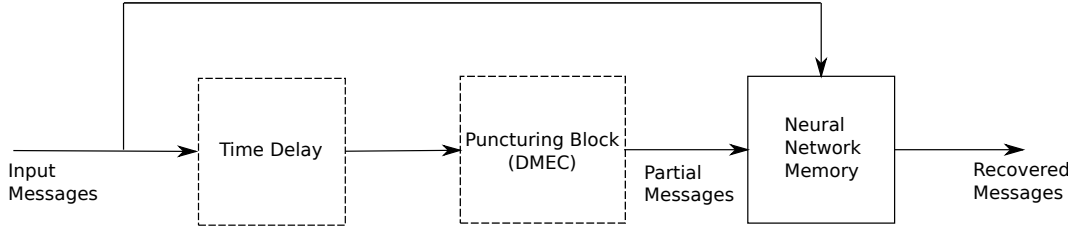


Figure 3.2: Information-theoretic Block Diagram

(in other words, the recovery is done after the storage). There are three points in the diagram at which the information is measured: (a) input messages (b) partial messages (c) recovered messages. The input messages are the realization of random vector \mathbf{m} , the partial messages are the realization of random vector $\tilde{\mathbf{m}}$ and the recovered messages are the realization of random vector $\hat{\mathbf{m}}$.

The mutual information rate between two random vectors is the amount of transmitted information from one random vector to another one, the mutual information rate between \mathbf{m} and $\hat{\mathbf{m}}$ being denoted by $I(\mathbf{m}; \hat{\mathbf{m}})$. The mutual information rate between \mathbf{m} and $\tilde{\mathbf{m}}$ is denoted by $I(\mathbf{m}; \tilde{\mathbf{m}})$. The network is fed with $\tilde{\mathbf{m}}$. It is then expected to extract additional information from the network connections, while exploiting $\tilde{\mathbf{m}}$. The given information to the network is equal to $I(\mathbf{m}; \tilde{\mathbf{m}})$. The extracted information after recovery is equal to $I(\mathbf{m}; \hat{\mathbf{m}})$. The network is called *informative* if and only if $I(\mathbf{m}; \hat{\mathbf{m}}) > I(\mathbf{m}; \tilde{\mathbf{m}})$. $I(\mathbf{m}; \hat{\mathbf{m}}) < I(\mathbf{m}; \tilde{\mathbf{m}})$ implies that the amount of information in hand by knowing $\tilde{\mathbf{m}}$ is more than what the network gives after the recovery. In this case, the network would lose information in the process of decoding. The differential information rate is defined as

$$\begin{aligned} I_{\text{diff}} &= I(\mathbf{m}; \hat{\mathbf{m}}) - I(\mathbf{m}; \tilde{\mathbf{m}}) \\ &= H(\mathbf{m}|\tilde{\mathbf{m}}) - H(\mathbf{m}|\hat{\mathbf{m}}) \end{aligned} \quad (3.3)$$

where $H(\cdot|\cdot)$ is the conditional entropy between two random vectors/variables and the network is informative if $I_{\text{diff}} > 0$. The computation of I_{diff} requires the computation of $I(\mathbf{m}; \tilde{\mathbf{m}})$ and $I(\mathbf{m}; \hat{\mathbf{m}})$. They are derived in the following sections.

3.4.1 Puncturing Block Channel Model

The proposed network is an auto-associative memory which recovers a message from its corresponding partial version. It was mentioned earlier that the missing sub-messages (in a partial message) are distributed uniformly random among significant sub-messages. To be precise, partial message \tilde{m} is message m with order c in which c_e sub-messages are missing. A sub-message in m may be unknown in \tilde{m} with probability $\epsilon = c_e/c$ (ϵ was introduced first in section 2.5.1). The generation of partial messages is assumed to be performed by a function, namely *puncturing block*. This block is modeled as a Discrete Memoryless Erasure Channel (DMEC) [CT06, PS07] in which a sub-message is either passed through the channel (unchanged) with probability $1 - \epsilon$ or it is mapped to null (here it is shown with 0) with probability ϵ . This channel model is depicted in Fig. 3.3. The amount of information at the DMEC input is equal to the sub-message's entropy $H(\mathbf{x})$ where \mathbf{x} is the sub-message random variable. Because \mathbf{x} is assumed i.i.d. and uniformly distributed over alphabet \mathcal{A} with cardinality ℓ , $H(\mathbf{x}) = \log_2 \ell$ [CT06]. \mathbf{z} is a random variable which takes values in $\mathcal{A} \cup \{0\}$. It represents the sub-messages that are passed through the puncturing block. The remained amount of information (in a sub-message) after the puncturing operation is computed through mutual information rate

$$\begin{aligned}
 I(\mathbf{x}; \mathbf{z}) &= H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}) \\
 &= H(\mathbf{x}) - \sum_z P(z)H(\mathbf{x}|\mathbf{z} = z) \\
 &= H(\mathbf{x}) - \epsilon H(\mathbf{x}|\mathbf{z} = 0) \\
 &= (1 - \epsilon)H(\mathbf{x}).
 \end{aligned} \tag{3.4}$$

The mutual information rate in (3.4) has a simple relation with source entropy $H(\mathbf{x})$: it reaches its maximum when there is no missing sub-messages, i.e., $\epsilon = 0$ and it reaches its minimum when all sub-messages are missing ($\epsilon = 1$).

The random vector \mathbf{m} can be decomposed into a c -tuple vector of c i.i.d. ran-

dom variable \mathbf{x} (sub-message random variable) and a support vector which holds the indices of the c significant values in \mathbf{m} . Therefore, the entropy of the message random vector is equal to

$$\begin{aligned} H(\mathbf{m}) &= \log_2 \binom{\chi}{c} + cH(\mathbf{x}) \\ &= \log_2 \binom{\chi}{c} + c \log_2 \ell \end{aligned}$$

bits, where the first term in the sum is the support vector entropy and the second is the significant sub-messages' entropy. The separation of the support vector and sub-messages entropies is of great importance, because the network may recover in two different modes: guided recovery and blind recovery. In guided recovery, the network is fed with the complete support vector information and in blind recovery, it is fed only with the support vector of the known sub-messages. The mutual information rate for guided recovery is computed as

$$\begin{aligned} I_G(\mathbf{m}; \tilde{\mathbf{m}}) &= \log_2 \binom{\chi}{c} + \sum_{i=1}^c I(\mathbf{x}; \mathbf{z}) \\ &= \log_2 \binom{\chi}{c} + c(1 - \epsilon) \log_2 \ell. \end{aligned} \tag{3.5}$$

In (3.5), the sum over $I(\mathbf{x}; \mathbf{z})$ is the implication of the previous assumption: the mutual information rate of c i.i.d. random variables in a vector is the sum of each random variable's mutual information rate [CT06].

In blind recovery, the support vector gives only the indices of $c_e = (1 - \epsilon)c$ known sub-messages. Therefore, the support vector requires $\log_2 \binom{\chi}{(1-\epsilon)c}$ bits to be represented. The mutual information rate for blind recovery is computed as

$$\begin{aligned} I_B(\mathbf{m}; \tilde{\mathbf{m}}) &= \log_2 \binom{\chi}{(1-\epsilon)c} + \sum_{i=1}^c I(\mathbf{x}; \mathbf{z}) \\ &= \log_2 \binom{\chi}{(1-\epsilon)c} + c(1 - \epsilon) \log_2 \ell. \end{aligned} \tag{3.6}$$

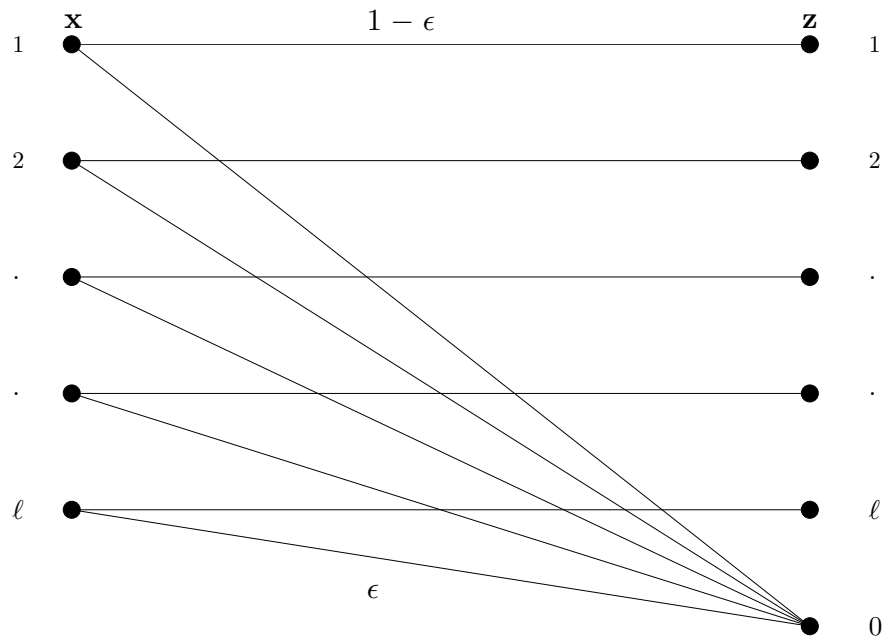


Figure 3.3: Discrete Memoryless Erasure Channel (DMEC) diagram for the puncturing block. The information flow is from left to right.

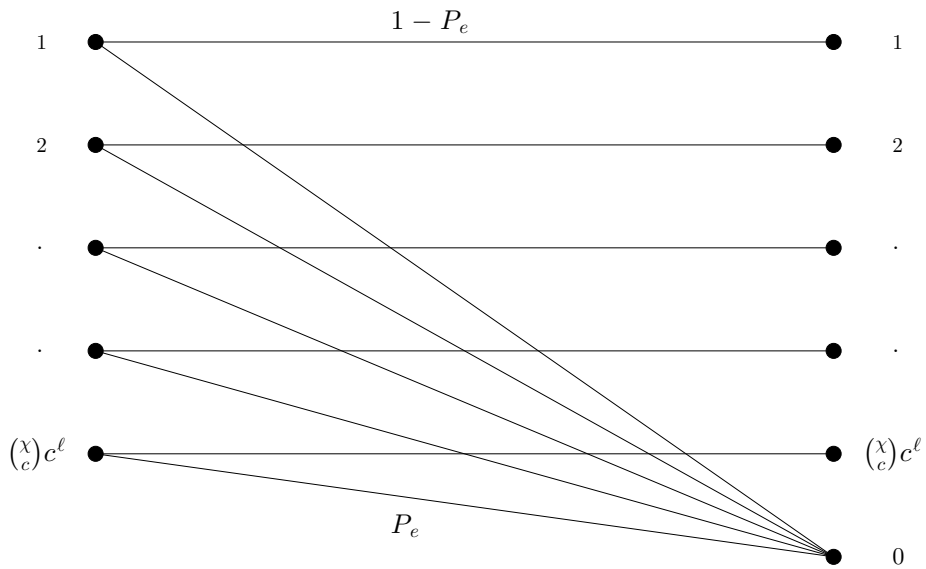


Figure 3.4: Discrete Memoryless Erasure Channel (DMEC) diagram for the neural network memory. The information flow is from left to right.

3.4.2 Neural Network Memory Channel Model

Set of stored messages \mathcal{M} passes through the network (as a black box) and is mapped to a set of retrieved messages. Although the internal storage and retrieval procedures of the messages are known, the underlying mapping between the set of stored messages and retrieved messages may not be easy to predict. This is worsen when considering more than one iteration step at the retrieval.

The recovery error was defined in section 2.5. If the network fails to recover even one sub-message in a message, an error occurs (although there may be correctly recovered sub-messages). By definition, an erroneous recovery bear no information about the stored message and its sub-messages, i.e., $I(\mathbf{m}; \hat{\mathbf{m}} | \mathbf{m} \neq \hat{\mathbf{m}}) = 0$. This is to guarantee the network functionality as a memory which is aimed to retrieve what has been stored. In Fig. 3.4, the network channel model that maps \mathbf{m} to $\hat{\mathbf{m}}$ is shown. The network mutual information rate is equal to

$$\begin{aligned}
 I(\mathbf{m}; \hat{\mathbf{m}}) &= \log_2 \binom{\chi}{c} + (1 - P_e)H(\mathbf{m}) \\
 &= \log_2 \binom{\chi}{c} + (1 - P_e)cH(\mathbf{x}) \\
 &= \log_2 \binom{\chi}{c} + (1 - P_e)c \log_2 \ell
 \end{aligned} \tag{3.7}$$

where P_e is the network error probability. P_e was given in section 2.5.1 for the single iteration blind and guided recoveries.

3.4.3 Recovery Constraints: Differential Information Rate

The first recovery constraint is guided recovery computed by simply substituting (3.7) and (3.5) into (3.3). Inequality

$$P_e < \epsilon \tag{3.8}$$

holds when the network is informative, i.e., $I_{\text{diff}} > 0$. From substituting (3.7) and (3.6) into (3.3), the blind recovery constraint is computed,

$$P_e < \frac{1}{c \log_2 \ell} \log_2 \frac{\binom{\chi}{c}}{\binom{\chi}{(1-\epsilon)c}} + \epsilon \quad (3.9)$$

when the network is informative. In Fig. 3.5, the differential information rate (as

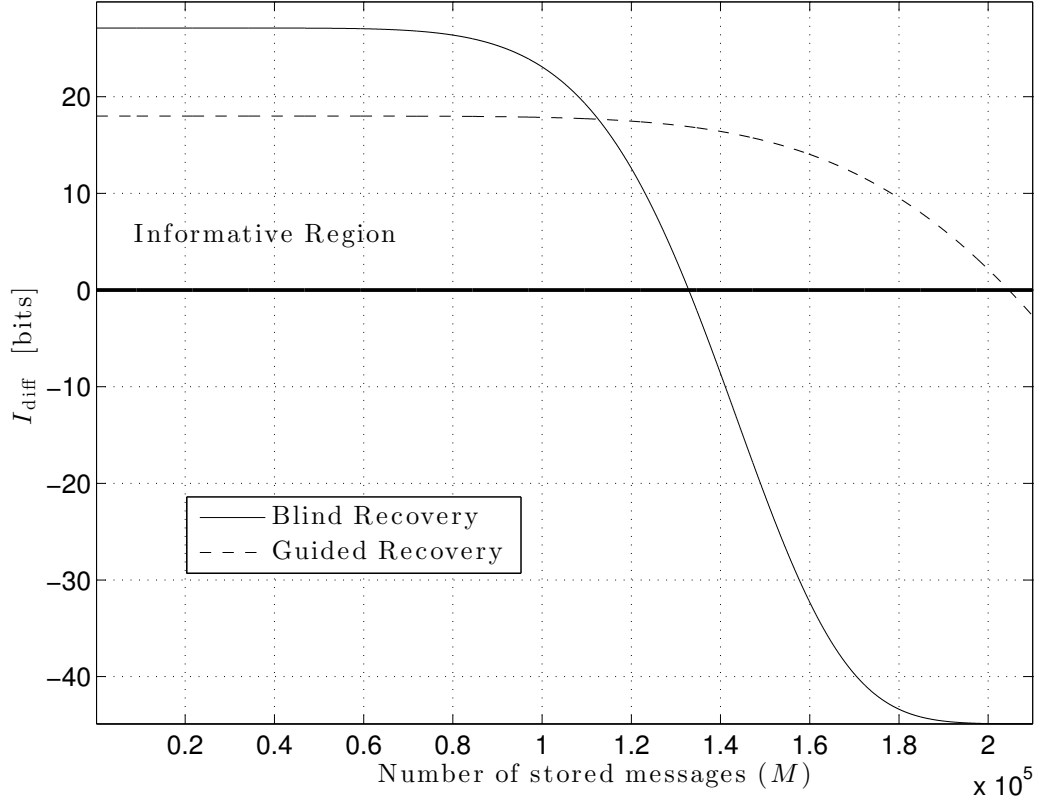


Figure 3.5: Differential Information Rate I_{diff} versus Number of Stored Messages M is illustrated for the network with $\chi = 100$ and $\ell = 64$. $\epsilon = 0.25$ and $c = 12$.

defined by (3.3)) for the network with $\chi = 100$ clusters and $\ell = 64$ fanals in each cluster is depicted. The network stores a set of messages with order $c = 12$ and it recovers the stored messages from their corresponding partial messages with $\epsilon = 0.25$. The error probabilities in (3.9) and (3.8) are calculated from the given error probabilities in chapter 2 for one iteration recovery. In this figure, the informative region where $I_{\text{diff}} > 0$ is separated with a thick solid line. As long as the network is in this region it extracts information from the network connections

and gives it to the additional information. The network in guided recovery mode can store more messages and remain in informative region in comparison with blind recovery. The differential information rates have a crossing point where they both are equal. It can be concluded that by feeding the network with the message support (in guided recovery) we may not gain in terms of the extracted information. The network also has a larger differential information rate for blind recovery than guided recovery when the number of stored messages is less than that of the crossing point. In Fig. 3.6, the given upper bounds in (3.9) and (3.8) on

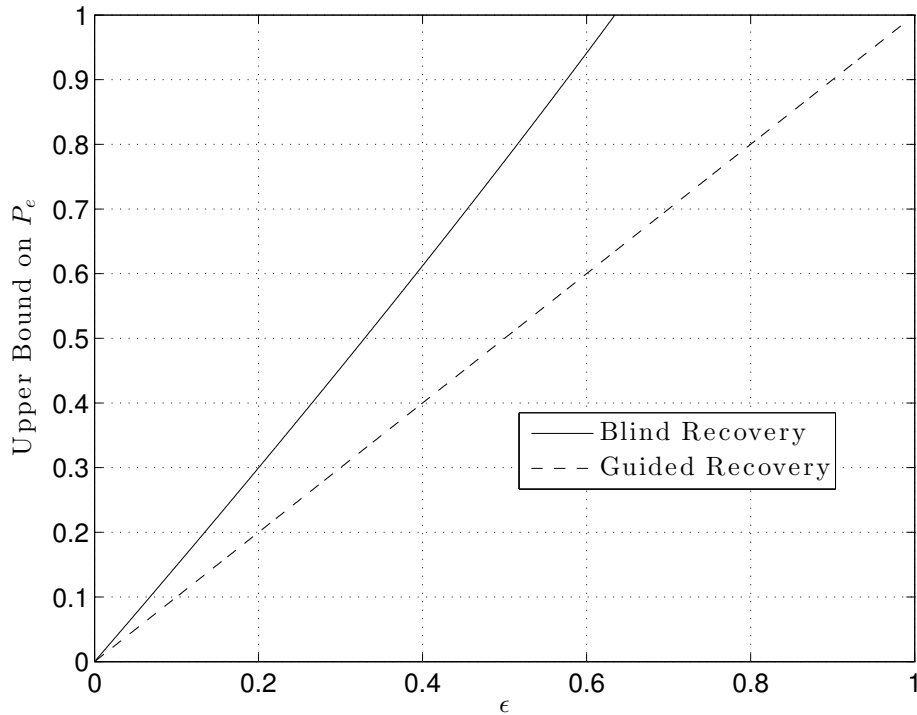


Figure 3.6: Upper bounds on P_e exploiting the DMEC model for the network. The network has $\chi = 100$ clusters, $\ell = 64$ fanals in each cluster. The stored messages' order is $c = 12$.

error probability P_e are depicted. The network remains in the informative region, i.e., $I_{\text{diff}} > 0$ if its recovery error rate is below the upper bounds. It is observed that the error probability upper bound for blind recovery grows faster and it reaches the unity where $\epsilon = 0.63$. This suggests that the network may not be in the

informative region for $\epsilon \geq 0.63$.

3.4.4 Recovery Constraints: Fano's Inequality

Fano's inequality is one of the information-theoretic utilities that are often used to compute bounds and constraints [CT06, Chapter 2]. Fano showed that for any two random variables \mathbf{y} and \mathbf{w} , inequality

$$H(p) + pH(\mathbf{w}) \geq H(\mathbf{w}|\mathbf{y}) \quad (3.10)$$

holds where $p = \Pr[\mathbf{y} \neq \mathbf{w}]$. This variation of Fano's inequality given in [CT06, p. 39] is used to compute the constraints in this section. The importance of Fano's inequality lies in the connection of the error probability with the information rate.

Theorem 1. Assuming $I(\mathbf{m}; \hat{\mathbf{m}}) < I(\mathbf{m}; \tilde{\mathbf{m}})$, the network holds

$$I(\mathbf{m}; \tilde{\mathbf{m}}) \geq (1 - P_e)H(\mathbf{m}) - H(P_e) \quad (3.11)$$

where $P_e = \Pr[\mathbf{m} \neq \hat{\mathbf{m}}]$.

Proof. Fano's inequality is written separately for $H(\mathbf{m}|\tilde{\mathbf{m}})$ and $H(\mathbf{m}|\hat{\mathbf{m}})$.

$$H(P_e) + P_e H(\mathbf{m}) \geq H(\mathbf{m}|\hat{\mathbf{m}}). \quad (3.12)$$

$$H(\epsilon) + \epsilon H(\mathbf{m}) \geq H(\mathbf{m}|\tilde{\mathbf{m}}). \quad (3.13)$$

The information rate of the puncturing block is

$$I(\mathbf{m}; \tilde{\mathbf{m}}) = H(\mathbf{m}) - H(\mathbf{m}|\tilde{\mathbf{m}}).$$

$I(\mathbf{m}; \hat{\mathbf{m}}) < I(\mathbf{m}; \tilde{\mathbf{m}})$ implies $H(\mathbf{m}|\hat{\mathbf{m}}) > H(\mathbf{m}|\tilde{\mathbf{m}})$. By substituting

$$H(\mathbf{m}|\tilde{\mathbf{m}}) = H(\mathbf{m}) - I(\mathbf{m}; \tilde{\mathbf{m}})$$

in

$$H(P_e) + P_e H(\mathbf{m}) \geq H(\mathbf{m}|\hat{\mathbf{m}}) > H(\mathbf{m}|\tilde{\mathbf{m}}),$$

equation (3.11) is obtained. □

The retrieval error probability for more than one iteration may not be straightforward to compute. Theorem 1 gives an upper bound on the network retrieval error probability. The network remains informative for a specific ϵ if and only if it has an error rate below P_e in (3.11). This inequality holds for any number of iterations and is independent of the storage and recovery procedures. In Fig. 3.7, the computed upper bound on P_e by (3.11) is depicted versus ϵ . The network has $\chi = 100$ clusters with $\ell = 64$ fanals in each cluster. The stored messages have $c = 12$ significant sub-messages.

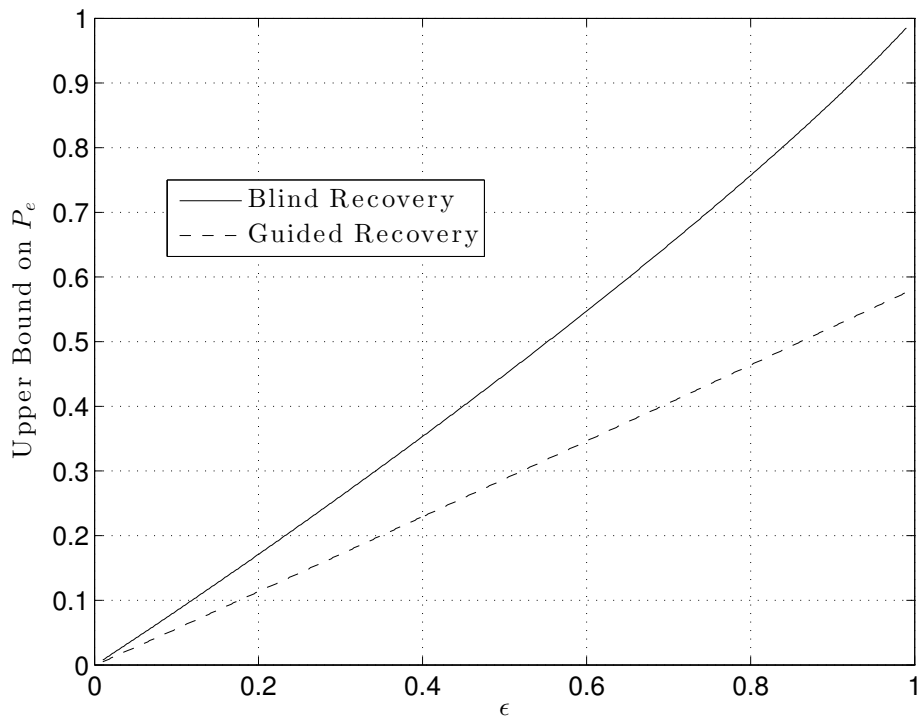


Figure 3.7: Upper bounds on P_e for the network with $\chi = 100$, $\ell = 64$ and $c = 12$.

3.4.5 Discussion

Two sets of upper bounds on P_e for blind and guided recoveries were computed in sections 3.4.3 and 3.4.4. The first upper bounds in section 3.4.3 were computed by condition $I_{\text{diff}} > 0$. In section 3.4.4, in addition to the condition on I_{diff} , Fano's

inequality was also used. Moreover, for the computation of the bound in Theorem 1 the network was not modeled as a DMEC. Indeed, the network may be considered to be any procedure which recovers \mathbf{m} from $\tilde{\mathbf{m}}$.

Fano's inequality is often exploited to obtain lower bounds on error probability. Theorem 1 computes the lowest value of P_e for which $I_{\text{diff}} < 0$. This implies that if the network error probability is equal to P'_e and $P'_e < P_e$, then $I_{\text{diff}} > 0$. Thus the given bound in Theorem 1 depends on not only I_{diff} but also Fano's inequality. Figs. 3.7 and 3.6 show that Theorem 1 gives a tighter bound than (3.9) and (3.8).

Chapter 4

Information-Theoretic

Constraints on Sparse Pattern

Recovery

Sparse signals have intrinsically a few nonzero elements or they can be represented in a basis that have a few nonzero coefficients. These signals appear in various applications such as image processing [BCNV08], Magnetic Resonance Imaging (MRI) [LDSP08], reflection seismology [SZ09, ZS10, HFYay] etc. As a very simple example, consider a signal that is constructed from superposition of a few sinusoids. This signal is not intrinsically sparse but its Fourier transform coefficients' vector is sparse. The nonzero coefficients are spikes on the frequency spots in which the sinusoids oscillate. Therefore, the coefficients form a sparse vector in which the informative parts are the nonzero elements.

Definition. A signal is k -sparse if its representation in a basis has only k nonzero coefficients [BCNV08]. A vector is k -sparse if it has only k nonzero elements¹.

¹In the existing literature in which sparse data structures are studied [BCNV08, CW08, Can08], sparse *signal* and sparse *vector* may be distinct. A sparse vector is a vector with a few nonzero elements whereas a sparse signal is not intrinsically sparse but it has sparse coefficients in a particular basis.

Although sparse signals and vectors are desirable, they are rare in practice. There may be a basis in which a signal is sparse along with very small nonzero residuals [Sto09]. This is the dominant situation in real-world applications. In order to have a purely sparse (without tiny residual coefficients), the signal must be noise-free (as a counterexample) and this is not usually the case. Besides, practical signals are naturally random and to find a basis that gives strictly sparse coefficients may not be feasible.

Definition. A vector is *strictly sparse* when there are only a few nonzero elements. A vector is *approximately sparse* when there are a few significantly large coefficients and the rest is small but not necessarily zero [BSB10, Sto09].

Sparseness or parsimony is an *a priori* knowledge about a signal. It may be exploited to improve the performance of information processing systems. In this report, the situation in which these signals have to be sampled and stored is considered. There are other applications such as blind deconvolution [HFYay, ZS10] that are not discussed here.

Various techniques are used to compress and store the sparse signals, including Sample-then-Compress Framework (SCF) [BCNV08] and Compressed-Sensing (CS) [BCNV08, CT05, Don06]. The former is a well known compression and storage scheme for which Joint Photographic Experts Group (JPEG) image compression standard is an example. In SCF, a device samples a real-world signal (including redundant information), then, in a post processing stage, it extracts useful information. Assume p -dimensional vector α is acquired through a sampling process that satisfies Nyquist criterion. Further assume α is sparse in a basis indicated by B and the complete set of transform coefficients $\{\beta_i\}$ is computed as $\beta = B\alpha$. The k largest coefficients in β are located and the $p - k$ smallest coefficients are discarded. The values and locations of those large coefficients are encoded and stored. This scheme has several drawbacks. It requires to sample at Nyquist rate

or higher that implies that the system acquires and stores enormous amount of redundant information. This imposes a data processing burden on the encoder whereas the decoding is less energy and computationally demanding. It may also lead to some practical limitations, especially in the hand-held devices and sensor networks where the devices are battery driven.

Compressed-sensing is an emerging field for compression and signal acquisition. It is the joint sampling and compression of sparse signals that is aimed to reduce the complexity of encoders. In this scheme, sampling and compression are not separate procedures (i.e. serially implemented) but they are performed simultaneously. The compressed-sensing systems avoid acquisition and processing of redundant information: the sampling rate is proportional to the underlying information content (sparsity level) rather than the bandwidth of the underlying signal [DMM09, TV11]. For example in [TV11], an under-Nyquist Analog-to-Information Conversion (AIC) scheme is proposed. CS can significantly reduce processing time and power consumption in the encoder (i.e. acquisition system). Moreover, there are situations in which acquiring sampled data is too expensive or practically infeasible. Such situations can be seen in physics experiments when the sensors are too costly or when a quantity is measured in a large spacial volume with few sensors. Although compressed-sensing is an efficient scheme at the sensory level, i.e., signal acquisition, it requires a sophisticated recovery procedure. A compressed-sensing setup is formulated as

$$Y = XB\alpha = X\beta \quad (4.1)$$

where $\alpha_{p \times 1}$ is the sparse signal, $\beta_{p \times 1}$ is the sparse vector, $B_{p \times p}$ is a basis and $X_{n \times p}$ is the sampling matrix. $Y_{n \times 1}$ is the sparse signal projection in a lower dimension hyper-space ($n < p$). It is assumed that β has at maximum k -nonzero elements (or significant elements if β is approximately sparse) and $k < p$. The system of linear equations in (4.1) is generally so-called *compressed-sensing* when $k/p \ll 1$. In this report, the system is named compressed-sensing regardless of k/p value.

The operations of a compressed-sensing system is packed into an underdetermined set of linear equations where the high dimensional sparse vector is mapped (compressed) into a lower dimensional vector. The compression is a simple mapping whereas the recovery (decompression) is a search for the corresponding vector (with higher dimension). In general such a system of equations has infinite number of solutions. It was first observed in reflection seismology that such a system may have a unique answer. In reflection seismology, a system transmits a series of sparse pulses after which it estimates the transfer function of the traversed path. It was known that if the pulses are sparse, there may be a unique solution (the transfer function) to the problem [SZ09, ZS10]. It was later shown (rigorously proved) in [Don06, Can06, CW08, Can08] that an underdetermined system of linear equations has a unique solution under certain conditions. These are algebraic conditions on the sparsity level and the sampling matrix (coefficients' matrix). There are two very well known conditions (among others) derived independently by Candès and Donoho. A sufficient condition on the sampling matrix that is so-called Restricted Isometric Property (RIP) given by Candès [Can06]. A sufficient and necessary condition on the sampling matrix that is so-called Null Space Property (NSP) given by Donoho [Don06].

4.1 Recovery

For sparse enough vectors in underdetermined linear system of equations there is an exact solution [Can06, Can08, Don06, BCNV08]. This is an optimization problem that finds the sparsest solution [Can06, Can08].

$$(P_0) \quad \begin{array}{ll} \text{minimize} & |\text{supp}(\tilde{\alpha})| \\ \text{subject to} & XB\tilde{\alpha} = Y, \end{array}$$

where $\text{supp}(\cdot)$ gives the support set of a vector. This optimization problem is non-polynomial (NP) time in general. In [CT05, Don06, CW08], Candès and Donoho

show that under Restricted Isometry Property (RIP) and Null-Space Property (NSP) conditions, the solution of

$$(P_1) \quad \begin{array}{ll} \text{minimize} & \|\tilde{\alpha}\|_{\ell_1} \\ \text{subject to} & XB\tilde{\alpha} = Y \end{array}$$

is equal to the solution of (P_0) . Unlike (P_0) , this optimization problem can be solved by a polynomial time algorithm that is so called ℓ_1 -norm minimization [CW08]. The importance of this is that Candès and Donoho rigorously showed that if the solution is sparse, there is a unique solution for the underdetermined system and this is obtained in polynomial time where NSP and/or RIP are/is satisfied. RIP is a sufficient condition whereas NSP is a necessary and sufficient condition which implies that a system satisfying RIP also satisfies NSP [Don06].

4.1.1 Restricted Isometry Property (RIP)

A sufficient condition for a stable recovery is that the matrix X preserves the length of any k -sparse vector being projected. This implies that the projected sparse vectors are not distorted by projection, therefore, the exact recovery is feasible. On the other hand, this condition is not necessary for the recovery of the signal and there might be other random projections (e.g. sampling matrices with smaller n) that lead to successful recovery.

Definition 6 (see [Can06, CW08]). A matrix $X \in \mathbb{R}^{n \times p}$ satisfies the RIP of order $k \in \mathbb{N}$ and isometry constant $\delta_k \in (0, 1)$ if

$$(1 - \delta_k)\|\beta\|_{\ell_2}^2 \leq \|X_\tau\beta\|_{\ell_2}^2 \leq (1 + \delta_k)\|\beta\|_{\ell_2}^2 \quad \forall \beta \in \mathbb{R}^{|\tau|} \quad (4.2)$$

where $\tau \subset \{i \in \mathbb{N} | i < n\}$ and $|\tau| \leq k$. X_τ is a submatrix of X with column indices from τ .

RIP expresses that every set of columns with cardinality less than k approximately behaves like an orthogonal vector space [CRTV05]. All matrices $X_\tau^t X_\tau$

have their eigenvalues in $[1 - \delta_k, 1 + \delta_k]$ and the k -sparse vectors can not be in the null-space of X_τ [YZ09, CW08].

This is bearing the pseudo-orthogonal vector space concept. RIP searches grouped vectors as vector spaces for an approximate orthogonality where X_τ is a $k \times |\tau|$ sub-matrix that holds (4.2) for any k -sparse vector in $\mathbb{R}^{|\tau|}$.

Theorem 2 (see [Can06, CT05]). Assume β is k -sparse and suppose that $\delta_{2k} + \delta_{3k} < 1$ or, better, $\delta_{2k} + \delta_{k,2k} < 1$. Then the solution $\hat{\beta}$ to (P_1) is exact, i.e., $\hat{\beta} = \beta$.

4.1.2 Null-Space Property (NSP)

The Null-Space Property (NSP) is given by Donoho in [Don06]. This is a necessary and sufficient condition which guarantees that none of the sparse vectors with sparsity level k lies in the null-space of the sampling matrix.

Theorem 3. (see [Don06]) A measurement matrix, $X_{n \times p}$ is assumed. Further, assume that $Y = X\beta$ and β has at most k nonzero elements and $\delta_{p \times 1}$ is a p -dimensional vector. Let $K \subset \{1, 2, \dots, n\}$ such that $|K| = k$ and let K_i denote the i -th element of K . Further, let $\bar{K} \subset \{1, 2, \dots, n\}/K$. Then (P_1) will produce a solution $\hat{\beta}$ that satisfies $\|\beta - \hat{\beta}\|_1 \leq \frac{2(C+1)}{C-1} \|\hat{\beta}_{\bar{K}}\|_1$ if and only if

$$(\forall \delta \in \mathbb{R}^p | X\delta = 0) \text{ and } \forall K, \sum_{i=1}^k |\delta_{K_i}| \leq \sum_{i=1}^{n-k} |\delta_{\bar{K}_i}|. \quad (4.3)$$

4.1.3 Verification of the Algebraic Conditions

The verification of both RIP and NSP for a specific sampling matrix is not a tractable task [dE10, dG11]. To verify any of those conditions, an algorithm should test the sampling matrix with a set of sparse vectors of infinite cardinality. This drawback has attracted the attention of researchers to randomly generated matrices (such as Gaussian matrices) and deterministic matrices [CW08, Can08, Don06].

The study of recovery conditions for random matrices is simplified since the probability density function of the generated elements is known. The deterministic matrices are generated (to satisfy the algebraic conditions) by carefully choosing the constructing vectors of the sampling matrix. For example in [LGGZ12], the deterministic construction of sampling matrices is studied. In [Don06], it is shown that, if the elements of the sampling matrix is drawn from the outcomes of an i.i.d. Gaussian random source, the matrix satisfies NSP.

4.2 Compressed Sensing and Neural Networks

Compressed-sensing provides under-Nyquist sampling [TV11] and dimensionality reduction. Neural network memory is concerned with learning mechanisms inspired by the human brain and information compression. The key to find the similarities between the two is the mathematical representation of compressed-sensing systems. It is a linear projection that maps a high dimensional sparse vector to a lower dimensional vector (that may not be sparse).

A single-layer feedforward neural network [Hay99] is similar to compressed-sensing formalization as expressed by (4.1). It is a network of input nodes that are connected to the output nodes through weighted connections (the sampling matrix elements). This is simply expressed by a linear projection that maps the inputs as a vector to an output vector [Hay99]. An example is Bidirectional Associative Memory (BAM) that was proposed by Kosko *et al.* in [Kos88]. It is a single-layer recurrent neural network memory that learns pairs of binary vectors in a matrix. The vector pairs may not have the same length. The recovery type is cued-recall (see section 1.1) where a vector in a pair is retrieved through another one. BAM decoding consisting the linear projection of the key vector (the known vector in a pair) followed by a thresholding function.

Dimensionality reduction is well studied under the formalism of machine learn-

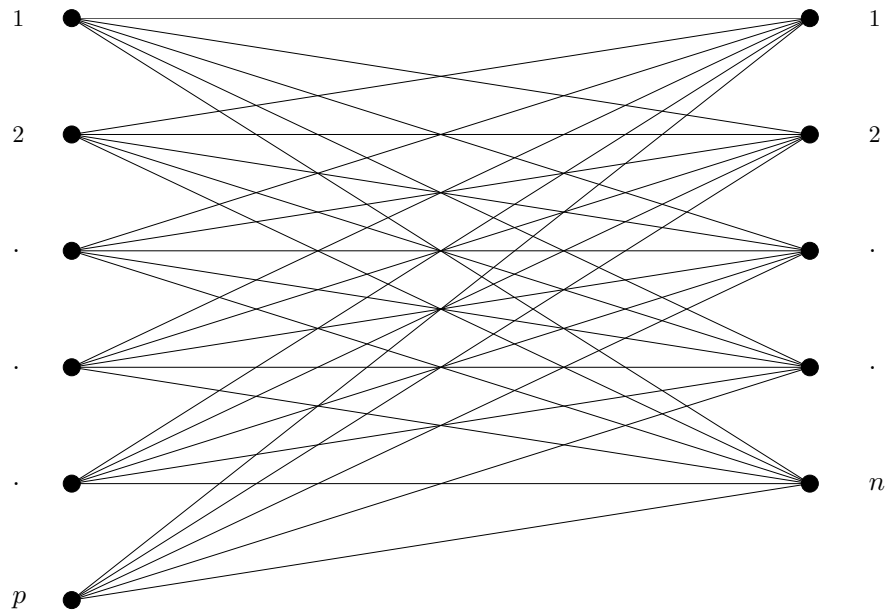


Figure 4.1: Compressed-sensing has been modeled as a single-layer neural network. The connections have random i.i.d. Gaussian weights.

ing. Principal Component Analysis (PCA) and Independent Component Analysis (ICA) [WCF07] are two machine learning techniques for dimensionality reduction. ICA and PCA methods find a basis for a known vector set in which the vectors may not be sparse. In case there are linear dependencies among the vectors, the computed basis maps the vectors to a lower dimension. This process is necessarily adaptive (because the basis is computed for a specific vector set) whereas compressed-sensing is not adaptive. On the other hand, the vectors subject to compression in ICA and PCA may not be necessarily sparse.

A compressed-sensing system may be considered as a single-layer neural network where the neural connections (weights) are i.i.d. Gaussian random values. These weights are kept constant throughout the learning process. In Fig. 4.1, this model is illustrated. The network responds to an input vector (stimulus) by giving a lower dimensional or a higher dimensional pair of that vector. In other words, the network has intrinsically learned (in the random matrix X) $\binom{p}{k}$ sparse vectors

and their corresponding lower dimensional pairs.

4.3 Information-Theoretic Conditions

The recovery conditions in section 4.1 were derived for an ideal setup when there is no distortion. In practice, sampling devices and digital computations introduce noise to information. Transistor junction noise, quantization noise and rounding noise are common sources of distortion. These examples are modeled in various systems as Additive White Gaussian Noise (AWGN) [CT06]. Computing bounds using algebraic tools on the number of measurements for such systems may be formidable. Thus, numerous researches [SBB06, AT10, Wai09, WWR10, ASZ10] have considered information theory as an alternative solution. Information theory stems from probability theory and the noise is intrinsically random and unpredictable. Therefore, such a probabilistic approach may be suitable to study the noisy systems in addition to the discussion given in section 4.1.3.

A noisy system of linear equations is considered for which it is *a priori* known that the solution is k -sparse (a vector with k nonzero elements).

$$Y = X\beta + W \quad (4.4)$$

where $X \in \mathbb{R}^{n \times p}$ is a random Gaussian measurement matrix with independently and identically distributed (i.i.d.) elements $X_{ij} \sim \mathcal{N}(0, 1)$. β is the k -sparse vector subject to measurement. W is a Gaussian noise vector $W \sim \mathcal{N}(0, I_{n \times n})$ where $I_{n \times n}$ is the n dimensional identity matrix. The support set of β is defined as

$$\text{supp}(\beta) \triangleq \{i : \beta_i \neq 0\},$$

that is a set of indices where elements of β are nonzero. The estimation of β as a function of X and Y is an inverse problem that consists of (a) detecting the support and (b) estimating the amplitudes of the nonzero elements [AT10, Rad11].

Once the support set of $\hat{\beta}$ is determined, the estimated sparse (optimal) solution is

$$\hat{\beta} = \arg \min_{\nu} \|Y - X_{\text{supp}(\hat{\beta})}\nu\|_2^2 \quad (4.5)$$

where $X_{\text{supp}(\hat{\beta})}$ is a $n \times k$ sub-matrix of the measurement matrix with column indices in $\text{supp}(\hat{\beta})$. Therefore, as it is discussed in [AT10, Rad11, WWR10], finding the optimal solution of such a noisy system is reduced to the exact support recovery.

The error metric is the 0–1 loss function defined in [Wai09, WWR10, AT10]

$$\rho(\beta, \hat{\beta}) = \mathbb{I} \left[\left\{ \hat{\beta}_i \neq 0, \forall i \in \text{supp}(\beta) \right\} \cap \left\{ \hat{\beta}_j = 0, \forall j \notin \text{supp}(\beta) \right\} \right] \quad (4.6)$$

where $\mathbb{I}(\cdot)$ is the indicator function. A decoder $\mathcal{D} : Y \rightarrow \theta$ maps the vector Y to a support set $\theta = \text{supp}(\hat{\beta})$. The probability of choosing a wrong support set is $\Pr[\theta \neq \text{supp}(\beta) | X, \beta]$ over the measurement noise and the sampling matrix. The average detection error is defined as in [Wai09, WWR10]

$$P_{\text{err}} = \frac{1}{\binom{p}{k}} \Pr[\mathcal{D}(Y) \neq \text{supp}(\beta) | X, \beta]$$

where the support $\text{supp}(\beta)$ is assumed to be chosen uniformly random from $\binom{p}{k}$ possible subsets of size k [Wai09, WWR10]. In an exact support recovery regime, the aim is to have asymptotically zero average detection error where $P_{\text{err}} \rightarrow 0$ as $n \rightarrow \infty$ [Wai09, WWR10].

In this report, asymptotic sufficient conditions (on exact support recovery) depend on the number of measurements n , the sparse vector dimension p , the sparsity level k and the signal-to-noise ratio

$$\text{SNR} = \frac{\mathbb{E}\{\|X\beta\|_2^2\}}{\mathbb{E}\{\|W\|_2^2\}} = \|\beta\|_2^2, \quad (4.7)$$

where the noise variance $\sigma^2 = 1$ and the measurement matrix elements are drawn from a Gaussian random source output with unit variance and zero mean. Though the signal-to-noise ratio is an important parameter, the exact support recovery

of a k -sparse vector is not solely guaranteed by its SNR [WWR10, Rad11]. The case in which the decoder has the highest failure probability for a given SNR is considered, by taking $|\beta_i| = \lambda$ where $i \in \text{supp}(\beta)$ and λ is the minimum absolute value of any nonzero element of the strictly sparse vector β . Therefore, by such assumption, the recovery of any k -sparse vector with $\text{SNR} \geq k\lambda^2$ is guaranteed [Wai09, WWR10, Rad11].

In [WWR10], the authors assume that β has a mean and variance stationary source. In [BSB10, SXC08], the authors model the support of a sparse signal as a vector of random elements, in which an element is an outcome of a random source with probability k/p to be nonzero that implies mean stationarity. In [BD11, JMB12] the authors model the high-dimensional sparse vector β as a realization of an ergodic stationary source. In signal processing, random sources are widely modeled as ergodic wide-sense stationary (EWSS) [Hay96, Hay01]. In this work, the sparse vector β is assumed to be random ergodic wide-sense stationary. Even though this assumption is common and most often inevitable in signal processing, it has not been considered in [Wai09, WWR10, Rad11, ASZ10, AT10, SBB06, JKR11, FRG09] to compute the information-theoretic constraints.

4.3.1 Bounds and Recovery Conditions

In [Wai09, WWR10, ASZ10] and this report, Fano's inequality [CT06] was exploited to obtain asymptotic constraints on the exact support recovery depending on (p, k, n, λ) . Fano's inequality in asymptotic form given in [WWR10, Rad11] is

$$\log \left(\binom{p-k+m}{m} - 1 \right) - \log 2 \leq \mathbb{E}_X I(\theta; Y), \quad (4.8)$$

for $m = 1, \dots, k$ where k is the sparsity level, p is the sparse vector dimension and $\mathbb{E}_X I(\theta; Y)$ is the expected information rate between measurement vector Y and detected support vector θ (see [ASZ10, WWR10]). m is the number of nonzero elements that is not known by the support detector, i.e., the detector *a priori*

knows indices of $k - m$ nonzero (or significantly large) elements. As it is discussed in [WWR10], having *a priori* knowledge about some nonzero elements may not facilitate the support detection. In fact, for some (p, k, m) where $m < k$ there are more support sets to choose than $m = k$. The cardinality of the support sets that may be chosen wrongly by the support detector is $\binom{p-k+m}{m} - 1$. This appears in the left-hand side of (4.8) that reaches a maximum value for a particular (p, k, m) due to its concavity (see the discussion in [WWR10]).

Strictly Sparse Signals

The recovery constraints in [Wai09, WWR10, Rad11, ASZ10, AT10, SBB06, JKR11, FRG09] are dedicated to strictly sparse signals. These signals have explicitly k nonzero elements and the rest are zeros.

The tightness of the recovery constraints through (4.8) depends on its right-hand side that is $\mathbb{E}_X I(\theta; Y)$ (see [WWR10, ASZ10]). To compute the information rate, the autocorrelation matrix of the sparse vector is required (see [WWR10, ASZ10, SBB06]). In general, the autocorrelation matrix is computed from the probability distribution function (PDF) of the process. The distribution function may not always be known and assuming a specific PDF restricts the given constraint to that specific random process. Therefore, in [SBB06, ASZ10] an upper bound on $I(\theta; Y)$ is computed and exploited to obtain their recovery constraints. In [WWR10], the authors compute a tighter upper bound (with respect to [SBB06, ASZ10]) on the information rate that results in tighter necessary conditions on the exact support recovery.

When the process is ergodic and wide-sense stationary (EWSS), the autocorrelation function and consequently the autocorrelation matrix can be computed from its time realizations [Pap91, Hay96]. The random sparse vector is chosen uniformly from $\binom{p}{k}$ possible k -sparse vectors of dimension p in which a vector element is nonzero with probability k/p [Pap91, BD11, SXC08]. This is employed

in [BD11, SXC08] to compute different statistics and sparse signals. EWSS assumption is exploited to compute the autocorrelation matrix for the worst case scenario (where $|\beta_i| = \lambda \forall i \in \text{supp}(\beta)$) without having knowledge of PDF. Having the autocorrelation function, the exact information rate can be computed for the worst case scenario EWSS signals. What remains as an obstacle is the computation of the expected information rate in (4.8) with respect to sampling matrix X (for which only its distribution is known). To overcome that, the combination of Jensen and Minkowski inequalities is exploited to obtain a lower bound on $\mathbb{E}_X I(\theta; Y)$. It is worth mentioning that the combination of Jensen and Minkowski inequalities for the information rate results in extremely tight bounds that is reported in [ONBP02, JG05]. This combination is of great importance. It facilitates the information rate computation through product of two polytopes' volumes. The first is a Wishart matrix (that defines a polytope) constructed from the Gaussian sampling matrix and the second is the autocorrelation matrix of the worst case ergodic wide-sense stationary signal. This tight lower bound on the expected information rate is used in (4.8) to get a sufficient condition depending on (p, n, k, λ) which is given in the following theorem.

Theorem 4. Assume a measurement matrix $X \in \mathbb{R}^{n \times p}$ whose elements are drawn from the outcome of an i.i.d. Gaussian random source with zero mean and unit variance, i.e., $X_{i,j} \sim \mathcal{N}(0, 1)$. A sufficient condition for asymptotically reliable recovery of a k -sparse ergodic wide-sense stationary signal β in which the nonzero elements $|\beta_i| \geq \lambda$ is

$$\log \left[\binom{p-k+m}{m} - 1 \right] - 1 \leq L \quad (4.9)$$

where

$$L = \frac{n}{2} \log \left[1 + \frac{m}{p-k+m} \lambda^2 \left(1 - \frac{m}{p-k+m} \right) \sqrt[n]{\Gamma(n) \binom{p-k+m-1}{n-1}} \right]$$

in which $\Gamma(\cdot)$ is Gamma function and $m = 1, \dots, k$.

By lower bounding further the information rate (right-hand side of (4.9)), the following sufficient condition is obtained.

Corollary 1. Assume a measurement matrix $X \in \mathbb{R}^{n \times p}$ whose elements are drawn from the outcome of an i.i.d. Gaussian random source with zero mean and unit variance, i.e., $X_{i,j} \sim \mathcal{N}(0, 1)$. A sufficient condition for asymptotically reliable recovery of a k -sparse ergodic wide-sense stationary signal β in which the nonzero elements $|\beta_i| \geq \lambda$ is

$$n > \max\{f_1(p, k, \lambda), \dots, f_k(p, k, \lambda), k\} \quad (4.10)$$

where

$$f_m(p, k, \lambda) = \frac{\log \left[\binom{p-k+m}{m} - 1 \right] - 1}{\frac{1}{2} \log \left(1 + \frac{m}{e} \lambda^2 \left(1 - \frac{m}{p-k+m} \right) \right)} \quad (4.11)$$

for $m = 1, \dots, k$.

For the sake of completeness and comparison, the necessary condition by Wang *et al.* is given.

Theorem 5. [WWR10] Assume the measurement matrix $X \in \mathbb{R}^{n \times p}$ whose elements are drawn from the outcome of an i.i.d. Gaussian source with zero mean and unit variance. A necessary condition for asymptotically reliable recovery of a k -sparse signal β in which nonzero elements $|\beta_i| \geq \lambda$ is

$$n > \max\{f_1(p, k, \lambda), \dots, f_k(p, k, \lambda), k\} \quad (4.12)$$

where

$$f_m(p, k, \lambda) = \frac{\log \left(\binom{p-k+m}{m} - 1 \right)}{\frac{1}{2} \log \left(1 + m \lambda^2 \left(1 - \frac{m}{p-k+m} \right) \right)} \quad (4.13)$$

for $m = 1, \dots, k$.

In Table 4.1 the given result for strictly sparse signals is compared with the sufficient condition in [Rad11] and the necessary condition in [WWR10]. In this table, the first three rows represent the constraints for linear sparsity regimes². The last three rows of the table show the constraints for sublinear regimes³. In the last two rows the signal-to-noise ratio tends to infinity, i.e., $k\lambda^2 \rightarrow \infty$.

Corollary 1 is obtained by loosening further the information rate lower bound with respect to right-hand side of (4.9) in Theorem 4. In spite of this, as demonstrated in Table 4.1, Corollary 1 is as tight as previous sufficient and necessary results. From this, it can be concluded that Theorem 4 may be asymptotically tighter than the results in [WWR10, Rad11].

The given tight sufficient condition in Corollary 1 is valid for the whole range of signal-to-noise ratio whereas the sufficient condition in [Rad11] is claimed to be restricted to $\lambda^2 = \Omega(\frac{1}{k})$ or $\lambda^2 = O(1)$ [Rad11].

The tightness of the sufficient conditions in Theorem 4 and Corollary 1 implies that the computed condition for approximately sparse signal (in the following section) may also be tight. This is concluded because the conditions (for strictly and approximately sparse signals) are computed in the same way.

Approximately Sparse Signals

The practical signals are not strictly sparse but they are approximately sparse with a few significantly large elements and the rest is small but nonzero. In this set-up, the existing constraints based on the sparsity level may not work.

A feature of Fano's inequality is connecting the detection error probability and the mutual information rate between source and estimation random vectors. The source random vector (here it is the approximately sparse vector) does not necessarily have a discrete alphabet but the estimated vector must be drawn from a

²In computation of the asymptotic constraints $k = \Theta(p)$ is taken as $k \approx p$, i.e., $k = p - 1$.

³In computation of the asymptotic constraints $k = o(p)$ implies that $k/p \rightarrow 0$, i.e., $k \ll p$.

discrete and countable alphabet [CT06]. This plays an important role in computation of recovery conditions for approximately sparse signals.

Assume that the wide-sense stationary and approximately sparse random vector β has k significantly large elements and the rest are small and the support detector recovers the k largest elements. To obtain recovery conditions for the approximately sparse signals assume decoder $\mathcal{D} : Y \rightarrow \theta$ where θ is a detected support set with cardinality k . The support set of an approximately sparse signal is defined to be the indices of the k largest elements

$$\text{supp}_k(\beta) \triangleq \{i : \beta_i \neq 0, \beta_i > \beta_j, k + 1 \leq j \leq p\}.$$

The error metric is defined as

$$\begin{aligned} \rho_k(\beta, \hat{\beta}) = \mathbb{I} \left[\left\{ \hat{\beta}_i \neq 0, \forall i \in \text{supp}_k(\beta) \right\} \right. \\ \left. \cap \left\{ \hat{\beta}_j < \beta_k, \forall j \notin \text{supp}_k(\beta) \right\} \right] \end{aligned} \quad (4.14)$$

where β_k is the k th largest element of β . The probability of choosing a wrong support set is $\Pr[\theta \neq \text{supp}_k(\beta) | X, \beta]$ and the average detection error is

$$P_e = \frac{1}{\binom{p}{k}} \Pr[\mathcal{D}(Y) \neq \text{supp}_k(\beta) | X, \beta]$$

where $\text{supp}_k(\beta)$ is the support set that is uniformly chosen from $\binom{p}{k}$ subsets of size k [Wai09, WWR10].

In this problem, a similar approach to strictly sparse signals is used. The combination of Jensen and Minkowski inequalities is used to obtain a lower bound on the expected information rate in (4.8). For the approximately sparse signals, it may not be possible to compute the autocorrelation matrix from (p, n, k) and λ can not generally be assumed constant. A lower bound on the volume of the polytope defined by the autocorrelation matrix can be alternatively computed from the signal power spectrum [Hay96]. Therefore, the power spectrum of the signal is assumed to be available through measurement and estimation. Having the power spectrum of the signal, the computation of the sufficient condition in the following theorem is straightforward.

Theorem 6. Assume a measurement matrix $X \in \mathbb{R}^{n \times p}$ whose elements are drawn from the outcome of an i.i.d. Gaussian random source with zero mean and unit variance, i.e., $X_{i,j} \sim \mathcal{N}(0, 1)$. A sufficient condition for asymptotically reliable recovery of a wide sense-stationary sparse signal β with k large nonzero elements is

$$\log \left[\binom{p-k+m}{m} - 1 \right] - 1 \leq L \quad (4.15)$$

where

$$L = \frac{n}{2} \log \left[1 + G \sqrt[n]{\Gamma(n) \binom{p-k+m-1}{n-1}} \right] \quad (4.16)$$

in which $\Gamma(\cdot)$ is Gamma function, $m = 1, \dots, k$ and G is the infimum of the approximately sparse signal power spectrum.

Corollary 2. Assume $k = \Theta(p)$. In Theorem 6 the sufficient condition for asymptotically reliable recovery of a wide-sense stationary and approximately sparse signal β is obtained by replacing

$$G = \frac{1}{2\pi} \int_0^{2\pi} \log S(\omega) d\omega \quad (4.17)$$

in (4.16) where $S(\omega)$ is the power spectrum of the sparse signal.

4.3.2 Proof

Theorem 4

To obtain the exact support recovery conditions on the number of measurements, Fano's inequality is exploited

$$P_e \geq 1 - \frac{I(\theta; Y) + \log 2}{\log \left(\binom{p}{k} - 1 \right)} \quad (4.18)$$

where $I(\theta; Y)$ is the mutual information between θ and Y [Wai09, WWR10, ASZ10, CT06]. θ is the detected support set that is given by the decoder, $\theta = \mathcal{D}(Y)$. P_e is the probability that the decoder fails to detect the correct support set, i.e.,

$$P_e = \Pr[\text{supp}(\beta) \neq \theta | X, \beta].$$

Scaling	Sufficient Condition Corollary 1 Necessary Condition Theorem 5 [WWR10] Sufficient Condition [Rad11]
$k = \Theta(p)$ $\lambda^2 = \Theta(\frac{1}{k})$	$n = \Theta(p \log p)$
$k = \Theta(p)$ $\lambda^2 = \Theta(\frac{\log k}{k})$	$n = \Theta(p)$
$k = \Theta(p)$ $\lambda^2 = \Theta(1)$	$n = \Theta(p)$
$k = o(p)$ $\lambda^2 = \Theta(\frac{1}{k})$	$n = \Theta(k \log(p - k))$
$k = o(p)$ $\lambda^2 = \Theta(\frac{\log k}{k})$	$n = \max \left\{ \Theta\left(\frac{k \log(p-k)}{\log k}\right), \Theta\left(\frac{k \log \frac{p}{k}}{\log \log k}\right) \right\}$
$k = o(p)$ $\lambda^2 = \Theta(1)$	$n = \max \left\{ \Theta\left(\frac{k \log \frac{p}{k}}{\log k}\right), \Theta(k) \right\}$

Table 4.1: Sufficient and Necessary Conditions on n for Exact Support Recovery in the Linear and the Sublinear Regimes. The sufficient condition in [Rad11] holds only for $\lambda^2 = \Omega(\frac{1}{k})$ or $\lambda^2 = O(1)$ whereas the other sufficient and necessary conditions hold for the entire signal-to-noise ratio range.

The error probability of the decoder, $P_{\text{err}}(\mathcal{D})$, is the average error probability with respect to Gaussian measurement matrix X . This is

$$P_{\text{err}}(\mathcal{D}) = \mathbb{E}_X P_e$$

where \mathbb{E}_X denotes the expected value operator with respect to X [AT10, WWR10]. Further, assume that the decoder has *a priori* knowledge of all nonzero locations of β but m locations with smallest values ($1 \leq m \leq k$). The decoder has to choose from $\binom{p-k+m}{m}$ support sets [WWR10]. Let \mathcal{U} be the set of unknown location indices where $|\mathcal{U}| = m$. The n -dimensional observation vector is

$$\tilde{Y} = \tilde{X} \tilde{\beta} + W$$

where \tilde{X} is the measurement matrix with column indices in \mathcal{U} . $\tilde{\beta}$ is the vector subject to measurement with element indices in \mathcal{U} . Therefore, the error probability of the decoder is bounded as

$$P_{\text{err}}(\mathcal{D}) \geq 1 - \frac{\mathbb{E}_{\tilde{X}} I(\theta; \tilde{Y}) + \log 2}{\log \left(\binom{p-k+m}{m} - 1 \right)}. \quad (4.19)$$

The mutual information $I(\theta; \tilde{Y})$ is given by

$$\begin{aligned} I(\theta; \tilde{Y}) &= H(\tilde{Y}|\tilde{X}) - H(\tilde{Y}|\theta, \tilde{X}) \\ &= H(\tilde{Y}|\tilde{X}) - H(W) \\ &= \frac{1}{2} \log \left| I_n + \tilde{X} R_{\tilde{\beta}} \tilde{X}^\dagger \right| \end{aligned} \quad (4.20)$$

where $R_{\tilde{\beta}}$ is the autocorrelation matrix of random vector $\tilde{\beta}$ and \dagger indicates Hermitian transpose. The equality $|I_p + AB| = |I_n + BA|$ holds for any pair of matrices $A_{p \times n}$ and $B_{n \times p}$ [Mey01]. This is an algebraic equality that implies equality of the volumes. This is required to separate the autocorrelation matrix from the sampling matrix and its conjugate transpose that facilitates the computation of the information rate. Therefore, (4.20) can be rewritten as

$$I(\theta; \tilde{Y}) = \frac{1}{2} \log \left| I_{p-k+m} + \tilde{X}^\dagger \tilde{X} R_{\tilde{\beta}} \right|. \quad (4.21)$$

$\text{rank}(\tilde{X}^\dagger \tilde{X} R_{\tilde{\beta}}) \leq \min\{\text{rank}(\tilde{X}^\dagger \tilde{X}), \text{rank}(R_{\tilde{\beta}})\}$. In [FZ07] it is shown that $\text{rank}(X^\dagger X) = n$, which implies $\min\{\text{rank}(X^\dagger X), \text{rank}(R_{\tilde{\beta}})\} = n$.

A lower bound on the mutual information is computed by applying Brunn-Minkowski inequality [CT06],

$$\begin{aligned} I(\theta; \tilde{Y}) &\geq \frac{n}{2} \log \left(1 + |\tilde{X}^\dagger \tilde{X} R_{\tilde{\beta}}|^{1/n} \right) \\ &= \frac{n}{2} \log \left(1 + \exp \left(\frac{1}{n} \log |\tilde{X}^\dagger \tilde{X} R_{\tilde{\beta}}| \right) \right). \end{aligned} \quad (4.22)$$

Now, by taking the expectation in (4.22) and by using Jensen's inequality

$$\mathbb{E}_{\tilde{X}} I(\theta; \tilde{Y}) \geq \frac{n}{2} \log_2 \left[1 + \exp \left(\frac{1}{n} (a + b) \right) \right] \quad (4.23)$$

where

$$a = \mathbb{E}_{\tilde{X}} \sum_{i=1}^n \log \sigma_i \left(\tilde{X}^\dagger \tilde{X} \right) \quad (4.24)$$

and

$$b = \sum_{i=1}^n \log \sigma_i \left(R_{\tilde{\beta}} \right). \quad (4.25)$$

$\sigma_i(\cdot)$ is the i th eigenvalue of a matrix. In (4.25), the eigenvalues are sorted $\sigma_1(R_{\tilde{\beta}}) < \sigma_2(R_{\tilde{\beta}}) < \dots < \sigma_{p-k+m}(R_{\tilde{\beta}})$.

(4.24) is the expected value of the logarithm of random Wishart matrix determinant $|\tilde{X}^\dagger \tilde{X}|$. The product of the eigenvalues of this random Wishart matrix is distributed as product of n chi-square distributed random variables [YL03, ONBP02], and therefore

$$\prod_{i=1}^n \sigma_i(\tilde{X}^\dagger \tilde{X}) \sim \prod_{j=1}^n \chi_{p-k+m-j+1}^2. \quad (4.26)$$

By taking the logarithm and then the expectation of the right-hand side of (4.26)

$$a = \mathbb{E} \sum_{j=1}^n \log \chi_{p-k+m-j+1}^2 = -n\gamma + \sum_{j=1}^n \sum_{\ell=1}^{p-k+m-j} \frac{1}{\ell} \quad (4.27)$$

where γ is Euler's constant [YL03, ONBP02]. In the asymptotic regime where $P_e(\mathcal{D}) \rightarrow 0$ as $n \rightarrow \infty$ that implies $p \rightarrow \infty$

$$\begin{aligned} \gamma &= \lim_{p \rightarrow \infty} \left[\sum_{\ell=1}^{p-k+m-j} \frac{1}{\ell} - \log(p-k+m-j) \right]. \\ a &= \sum_{j=1}^n \log(p-k+m-j) \\ &= \log(p-k+m-j-1)! - \log(p-k+m-n-1)! \\ &= \log \frac{\Gamma(p-k+m)}{\Gamma(p-k+m-n)} \\ &= \log \frac{\Gamma(p-k+m)}{\Gamma(n)\Gamma(p-k+m-n)} \Gamma(n) \\ &= \log \binom{p-k+m-1}{n-1} + \log \Gamma(n) \end{aligned} \quad (4.28)$$

where $\Gamma(\cdot)$ is Gamma function. The sparse vector elements $\tilde{\beta}_i$ is modeled as output of an ergodic wide-sense stationary random vector source. Therefore, the elements of such random vector is also ergodic wide-sense stationary. The nonzero elements appear with probability $\Pr[\tilde{\beta}_i \neq 0] = \frac{m}{p-k+m}$ [Pap91,BSB10,SXC08]. The autocorrelation matrix of $\tilde{\beta}$ is a Hermitian Toeplitz matrix that is obtained from its autocorrelation function [Hay96]. Assume that the nonzero elements $\tilde{\beta}_i$ where

$i \in \mathcal{U}$, are negative with probability ξ , i.e., $\xi = \Pr[\tilde{\beta}_i \in \mathbb{R}^-]$. The autocorrelation function at lag $\tau = 0$ is

$$r_{\tilde{\beta}}(0) = \frac{1}{p-k+m} \sum_i \tilde{\beta}_i \tilde{\beta}_i = \frac{m}{p-k+m} \lambda^2.$$

For lag $\tau \neq 0$ and $|\tau| \leq p$, the autocorrelation function is

$$\begin{aligned} r_{\tilde{\beta}}(\tau) &= \frac{1}{p-k+m} \sum_i \tilde{\beta}_i \tilde{\beta}_{i+\tau} \\ &= \frac{p-k+m-|\tau|}{p-k+m} \left(\frac{m}{p-k+m} \right)^2 \lambda^2 \\ &\quad \times [\xi^2 + (1-\xi)^2 - 2\xi(1-\xi)]. \end{aligned}$$

For $|\tau| > p$ the autocorrelation function $r_{\tilde{\beta}}(\tau) = 0$. Therefore, the autocorrelation function is

$$r_{\tilde{\beta}}(\tau) = \begin{cases} \frac{(p-k+m-|\tau|)^2 m^2}{(p-k+m)^3} (4\xi^2 - 4\xi + 1) \lambda^2 & \text{if } |\tau| \leq p, \tau \neq 0 \\ \frac{m}{p-k+m} \lambda^2 & \text{if } \tau = 0, \\ 0 & \text{Otherwise.} \end{cases}$$

The autocorrelation matrix of $\tilde{\beta}$ is a Hermitian Toeplitz matrix with $r_{\tilde{\beta}}(\tau)$ in its first row, i.e.,

$$R_{\tilde{\beta}} = \text{Toeplitz}\{r_{\tilde{\beta}}\}.$$

$R_{\tilde{\beta}}$ and $r_{\tilde{\beta}}$ is used to find a lower bound on (4.25),

$$n \log \sigma_{\min} \leq \sum_{i=1}^n \log \sigma_i (R_{\tilde{\beta}})$$

where $\sigma_{\min} \leq \sigma_i(R_{\tilde{\beta}})$ is the lower bound on all eigenvalues of $R_{\tilde{\beta}}$. The minimum eigenvalue of $R_{\tilde{\beta}}$ is lower bounded by the infimum of the power spectrum [Hay96],

$$\sigma_{\min} = \min_{\omega} S_{\tilde{\beta}}(\omega) \quad (4.29)$$

where

$$\begin{aligned} S_{\tilde{\beta}}(\omega) &= \left[\frac{\sin(\omega(p+1/2))}{\sin(\omega/2)} \right]^2 \left(\frac{m}{p-k+m} \lambda \right)^2 (4\xi^2 - 4\xi + 1) \\ &\quad + \frac{m}{p-k+m} \lambda^2 - \left(\frac{m}{p-k+m} \lambda \right)^2 (4\xi^2 - 4\xi + 1) \end{aligned} \quad (4.30)$$

112
is the power spectrum of $\tilde{\beta}$. It is computed by taking Fourier transform of $r_{\tilde{\beta}}$ [Hay96]. Therefore,

$$\sigma_{\min}(\xi) = \left[\frac{m}{p-k+m} - \left(\frac{m}{p-k+m} \right)^2 (4\xi^2 - 4\xi + 1) \right] \lambda^2 \quad (4.31)$$

from (4.30) and (4.29). σ_{\min} is a function of ξ . The term $0 \leq 4\xi^2 - 4\xi + 1 \leq 1$ reaches its maximum when $\xi \in \{0, 1\}$, i.e., the sparse vector subject to measurement is unipolar. To lower bound the minimum eigenvalue ξ is chosen either 0 or 1, $\xi \in \{0, 1\}$, that results in

$$b \geq n \log \left[\left(1 - \frac{m}{p-k+m} \right) \frac{m}{p-k+m} \lambda^2 \right]. \quad (4.32)$$

Substituting left-hand side of (4.32) and (4.28) in (4.23) gives

$$\mathbb{E}_X I(\theta; \tilde{Y}) \geq \frac{n}{2} \log \left[1 + \sqrt[n]{\Gamma(n) \binom{p-k+m-1}{n-1}} \right. \\ \left. \left(1 - \frac{m}{p-k+m} \right) \frac{m}{p-k+m} \lambda^2 \right]. \quad (4.33)$$

Finally, the sufficient condition is computed as

$$\log \left[\binom{p-k+m}{m} - 1 \right] - \log(2) \leq L \quad (4.34)$$

where

$$L = \frac{n}{2} \log \left[1 + \sqrt[n]{\Gamma(n) \binom{p-k+m-1}{n-1}} \right. \\ \left. \left(1 - \frac{m}{p-k+m} \right) \frac{m}{p-k+m} \lambda^2 \right] \quad (4.35)$$

for $m = 1, \dots, k$.

Corollary 1

In proof of Theorem 4

$$a = \log \left(\binom{p-k+m-1}{n-1} \right) + \log \Gamma(n) \quad (4.36)$$

that can be lower bounded to obtain another lower bound on $\mathbb{E}_X I(\theta, \tilde{Y})$. On the other hand

$$\sqrt{2\pi(n-1)}(n-1)^{(n-1)}e^{-(n-1)} \leq \Gamma(n) \quad (4.37)$$

and

$$(n-1) \log \left(\frac{p-k+m-1}{n-1} \right) \leq \log \binom{p-k+m-1}{n-1}. \quad (4.38)$$

Considering $n \rightarrow \infty$ (see the asymptotic Fano's inequality in [WWR10]) and consequently $p \rightarrow \infty$

$$\begin{aligned} a/n &\geq \frac{(n-1)}{n} \log \frac{p-k+m-1}{n-1} + \frac{1}{n} \log \left[\sqrt{2\pi(n-1)}(n-1)^{(n-1)}e^{-(n-1)} \right] \\ &\geq \frac{n-1}{n} \log(p-k+m-1) - \frac{n-1}{n}. \end{aligned} \quad (4.39)$$

By replacing the right-hand sides of (4.39) and (4.32) in (4.23)

$$L = \frac{n}{2} \log \left[1 + \frac{m}{e} \left(1 - \frac{m}{p-k+m} \right) \lambda^2 \right] \quad (4.40)$$

in (4.34).

Theorem 6

In proof of Theorem 4 it is assumed that β is strictly sparse. Where the signal is not strictly sparse it may not be tractable to compute analytically a lower bound on (4.25) as it is computed in Theorem 4. In the proof of Theorem 4 a lower bound on b is obtained through computation of a lower bound on the minimum eigenvalue of the signal autocorrelation matrix. The minimum eigenvalue of the autocorrelation matrix of the signal is lower bounded by the infimum of the power spectrum (see [Hay96] for details). Assume that the power spectrum of the approximately sparse signal is known (from experiments or from computations). By replacing $\sigma_i(R_{\tilde{\beta}})$ with its lower bound (see (4.29) and (4.25)) that is obtained from the signal power spectrum the sufficient condition in Theorem 6 is computed.

Corollary 2

Consider $k = \Theta(p)$. In (4.25) the eigenvalues of the autocorrelation matrix are sorted. They are n smallest eigenvalues out of p . To tighten the condition in Theorem 4 a lower bound in [MV97] is used on the product of n smallest eigenvalues of the matrix instead of lower bounding (4.25) by replacing the eigenvalues with the minimum eigenvalue. Therefore, a lower bound on (4.25) is computed as

$$\begin{aligned} b &\geq (p - n) \log \left(\frac{p - n}{\text{SNR}} \right) + \log |R_{\tilde{\beta}}| \\ &= (p - n) \log \left(\frac{p - n}{\text{SNR}} \right) + \left[\frac{n}{2\pi} \int_0^{2\pi} \log S(\omega) \, d\omega \right] \end{aligned} \tag{4.41}$$

where the second term in the right-hand side is the integral of the logarithm of the signal power spectrum [MV97, Gra06]. Noting that inequalities $k \leq p$ and $n \geq k$ hold, $k = \Theta(p)$ implies that

$$G = \frac{1}{2\pi} \int_0^{2\pi} \log S(\omega) \, d\omega.$$

Chapter 5

Discussion and Conclusion

The first part of this report was dedicated to the presentation and analysis of a new neural network able to store a large amount of sparse messages and to retrieve them from partial and/or distorted versions. The simulation results showed that we can store a significant fraction of M_{\max} in the network (we can reach efficiencies as high as 30% with low recovery error but this depends on the different parameters). It can be concluded that the network does not offer strict optimality but something close to it. In addition to the mentioned properties, a main feature of the proposed network was plausibility. It was intended to design a network that mimics the biological and psychological features of the human neocortex.

In the second part, we were interested in compressed sensing (CS) systems, mainly through the evaluation of their feasible compression rates in presence of noise. A new tight sufficient condition for strictly sparse signals was established. This holds for the entire signal-to-noise ratio unlike another existing condition. The same computation technique was used to determine a sufficient condition for approximately sparse signals. To the best of our knowledge there was no such conditions for these types of signals before. It may be concluded, that this condition is also tight, because the computation procedure had been led to a tight condition before (for strictly sparse signals).

These two parts are strongly related: the purpose of the neural network is to store sparse messages with the largest possible efficiency (i.e. with the best use of available material resource) whereas the CS system is intended to transmit sparse messages spending the less possible energy (i.e. with the shortest possible length). It was then natural to try to study the properties of the proposed network in the light of CS theory.

The CS principle can be described by a bipartite graph which associates vectors of length p with vectors of length $n < p$ through weighted connections (cf. Fig. 1.2). This is the simplest graphical way to describe a matrix-vector product, which can then be directly transcribed into a perceptron-like neural network. The network proposed in this report is quite different: (a) it is not an operator but a memory. (b) it has a binary structure (connections have weight 0 or 1). (c) source and channel coding are both performed by the network.

Regarding the last point, source coding comes from two features. First, any message of length χ is stored as a clique with only c vertices ($c \ll \chi$) for sparse messages); second, when M messages are stored, the corresponding cliques share vertices and edges, which adds to the compression rate. As for channel coding, it is assured by the strongly redundant structure of cliques.

Although the CS operation and the proposed neural network are both intended to process sparse messages, the comparison in terms of compression rate and performance is not straightforward.

To compare compressed sensing and the proposed network, consider a scenario. The same set of sparse vectors is compressed by both systems, thereafter, the compressed information is transmitted. The main interest is finding out which one is possibly more efficient. (1) Assume a noise-free compressed sensing system along with a genie support detector. It transforms a k -sparse analog vector of length p into an analog vector of length $n < p$. Then, the latter is quantized on ℓ levels, thus producing a message of $n \log_2 \ell$ bits. In a given slot of time, M such

messages are compressed and sent by some communication system. The total amount of transmitted information is $Mn \log_2 \ell$ bits. (2) M c -sparse vectors of length χ , each character (or sub-message) being quantized on ℓ levels, are stored in the proposed neural network. Then, the communication system sends the state of the network, that is, $Q = \frac{\chi(\chi-1)\ell^2}{2}$ bits. According to efficiency-1 diversity given by (2.5), an upper bound on M is

$$M_{\max} = \frac{0.5\chi(\chi-1)\ell^2}{\log_2 \binom{\chi}{c} + c \log_2 \ell}.$$

If we want the classical CS system to send as many messages as M_{\max} , with $p = \chi$, $k = c$, the same level ℓ of quantization and with the same total amount of bits Q , we need $Q = M_{\max} n \log_2 \ell$, which leads to:

$$n = \frac{\log_2 \binom{\chi}{c} + c \log_2 \ell}{\log_2 \ell}$$

For some values of the parameters, the first term in numerator may be neglected by comparison with the second one, which then leads to a rather interesting result: $n \approx c$, meaning that the classical system must achieve optimality (n of the order of $c = k$). Reversely, this means that the neural network may be considered as optimum as a compressed-sensing and storing unit.

Certainly, all this is approximation. Moreover, in the second situation, the messages are not ordered in the network and we have to use additional clusters to store addresses (for instance, with $\chi = 100$, we can solve this problem, with 5 additional clusters, that is about 5% extra material).

The proposed network and compressed-sensing were both modeled as noisy communication channels. Then, they were analysed using Fano's inequality to compute the recovery conditions. Discrete Memoryless Erasure Channel (DMEC) model was used for the analysis of the network and Multiple Input Multiple Output (MIMO) noisy channel was used for compressed-sensing. Regardless of the type of channel model, Fano's inequality connects the detection error probability to the system information rate. It was observed that the computed bounds are tight: the

network bound is tighter than the given bound by l_{diff} introduced in [SSP96] and compressed-sensing conditions are tighter than previous works.

Bibliography

- [AGJ12] B. K. Aliabadi, C. Berrou V. Gripon, and X. Jiang. Learning sparse messages in networks of neural cliques. *IEEE Trans. Neural Networks*, 2012. Submitted to.
- [ALP06] B. B. Averbeck, P. E. Latham, and A. Pouget. Neural correlations, population coding and computation. *Nature Reviews Neuroscience*, 7(5):358–366, 2006.
- [AM89] Y. S. Abu-Mostafa. Information theory, complexity and neural networks. *IEEE Communications Magazine*, 27(11):25–28, November 1989.
- [AS68] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet Taylor Spence, editors, *Psychology of Learning and Motivation*, volume 2, pages 89–195. Academic Press, 1968.
- [ASZ10] S. Aeron, V. Saligrama, and M. Zhao. Information theoretic bounds for compressed sensing. *IEEE Trans. Inf. Theory*, 56(10):5111–5130, October 2010.
- [AT10] M. Akcakaya and V. Tarokh. Shannon-theoretic limits on noisy compressive sampling. *IEEE Trans. Inf. Theory*, 56(1):492–504, January 2010.

- [BCNV08] R. G. Baraniuk, E. J. Candès, R. Nowak, and M. Vetterli. Compressive sampling. *IEEE Signal Processing Magazine*, 25(2):12–13, March 2008.
- [BD11] D. Baron and M. Duarte. Universal map estimation in compressed sensing. In *Proc. 49th Annual Allerton Conf. on Commun., Control, and Computing*, September 2011.
- [BETvG08] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, June 2008.
- [BG96] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Commun.*, 44(10):1261–1271, October 1996.
- [BH74] A. D. Baddeley and G. J. Hitch. Working memory. In G. Bower, editor, *The psychology of learning and motivation*, volume VIII, pages 47–89. Academic Press, 1974.
- [BSB10] D. Baron, S. Sarvotham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Trans. Signal Processing*, 58(1):269–280, January 2010.
- [BT99] A. Borst and F. E. Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.
- [Can06] E. J. Candès. Compressive sampling. In *Proc. of the International Congress of Mathematicians, Madrid, Spain, 2006*.
- [Can08] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l'Academie des Sciences, Serie I(346)*:589–592, February 2008.

- [CBP⁺05] L. Cruz, S. V. Buldyrev, S. Peng, D. L. Roe, B. Urbanc, H. E. Stanley, and D. L. Rosene. A statistically based density map method for identification and quantification of regional differences in microcolumnarity in the monkey brain. *Journal of Neuroscience Methods*, 141(2):321–332, May 2005.
- [CRTV05] E. J. Candès, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. In *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 295–308, 2005.
- [CT05] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, December 2005.
- [CT06] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 2006.
- [CW08] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.
- [dE10] A. d’Aspremont and L. El Ghaoui. Testing the nullspace property using semidefinite programming. *Math. Progr.*, February 2010. Special issue on machine learning.
- [dG11] A. d’Aspremont and L. El Ghaoui. Testing the nullspace property using semidefinite programming. *Mathematical Programming*, 127(1):123–144, October 2011.
- [Die10] R. Diestel. *Graph Theory*. Springer, 2010.
- [DMM09] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proc. of the National Academy of Sciences of the United States of America*, 106(45), 2009.

- [Don06] D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.
- [Erm92] B. Ermentrout. Complex dynamics in winner-take-all neural nets with slow inhibition. *Neural Networks*, 5(3):415–431, 1992.
- [FÖ2] P. Földiák. Sparse coding in the primate cortex. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 1064–1068. MIT Press, 2nd edition, 2002.
- [FRG09] A. Fletcher, S. Rangan, and V. Goyal. Necessary and sufficient conditions on sparsity pattern recovery. *IEEE Trans. Inf. Theory*, 55(12):5758–5772, December 2009.
- [FZ07] X. Feng and Z. Zhang. The rank of a random matrix. *Applied Mathematics and Computation*, 185(1):689–694, February 2007.
- [Gal62] R. Gallager. Low-density parity-check codes. *IEEE Trans. Inf. Theory*, 8(1):21–28, January 1962.
- [GB11] V. Gripon and C. Berrou. Sparse neural networks with large learning diversity. *IEEE Trans. Neural Networks*, 22(7):1087–1096, July 2011.
- [GB12] V. Gripon and C. Berrou. Nearly-optimal associative memories based on distributed constant weight codes. In *Proc. Workshop on Information Theory and Applications (ITA)*, pages 269–273, San Diego, February 2012.
- [Gra06] R. M. Gray. *Toeplitz and Circulant Matrices: A Review*. Now Publishers Inc., 2006.
- [GW04] J. Grainger and C. Whitney. Does the human mind read words as a whole? *Trends in Cognitive Sciences*, 8(2):58–59, 2004.

- [Ham50] R. W. Hamming. A mathematical theory of communication. *Bell System Technical Journal*, 29:147–160, April 1950.
- [Hay96] M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. Wiley, 1996.
- [Hay99] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [Hay01] S. O. Haykin. *Adaptive Filter Theory*. Prentice Hall, 2001.
- [Hay05] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas in Commun.*, 23(2):201–220, February 2005.
- [Hay09] S. Haykin. *Communication Systems*. Wiley, 5th edition, 2009.
- [HFYay] F.J. Herrmann, M.P. Friedlander, and O. Yilmaz. Fighting the curse of dimensionality: Compressive sensing in exploration seismology. *IEEE Signal Processing Magazine*, 29(3):88–100, May.
- [IGM99] J. Mitola III and Jr. G.Q. Maguire. Cognitive radio: making software radios more personal. *IEEE Personal Communications*, 6(4):13–18, August 1999.
- [JG05] Shi Jin and Xiqi Gao. Tight lower bounds on the ergodic capacity of ricean fading MIMO channels. In *Proc. IEEE International Conference on Communications (ICC)*, volume 4, pages 2412–2416, May 2005.
- [JKR11] Yuzhe Jin, Young-Han Kim, and B.D. Rao. Limits on support recovery of sparse signals via multiple-access communication techniques. *IEEE Trans. Inf. Theory*, 57(12):7877–7892, December 2011.
- [JMB12] S. Jalali, A. Maleki, and R. Baraniuk. Minimum complexity pursuit: Stability analysis. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2012.

- [Kos88] B. Kosko. Bidirectional associative memories. *IEEE Trans. Syst., Man, and Cyber.*, 18(1):49–60, 1988.
- [LC04] S. Lin and D. J. Costello. *Error Control Coding*. Prentice-Hall, 2nd edition, 2004.
- [LDSP08] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, 25(2):72–82, March 2008.
- [LGGZ12] Shuxing Li, Fei Gao, Gennian Ge, and Shengyuan Zhang. Deterministic construction of compressed sensing matrices via algebraic curves. *IEEE Trans. Neural Networks*, 58(8):5035–5041, August 2012.
- [Loe04] H.-A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, jan. 2004.
- [Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Seventh IEEE Int. Conf. on Computer Vision*, volume 2, pages 1150–1157, 1999.
- [Maa00] W. Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519–2535, November 2000.
- [Mey01] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 2001.
- [Mil56] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, March 1956.
- [MM65] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.

- [Mou97] V. B. Mountcastle. The columnar organization of the neocortex. *Brain : a journal of neurology*, 120(4):701–722, 1997.
- [MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, 1943.
- [MV97] J. K. Merikoski and A. Virtanen. Bounds for eigenvalues using the trace determinant. *Linear Algebra and its Applications*, 264:101–108, October 1997.
- [ONBP02] O. Oyman, R. U. Nabar, H. Bölcskei, and A. J. Paulraj. Tight lower bounds on the ergodic capacity of Rayleigh fading MIMO channels. In *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, pages 1172–1176, November 2002.
- [Pap91] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.
- [PB04] C. A. Perfetti and Donald J. Bolger. The brain might read that way. *Scientific Studies of Reading*, 8(3):293–304, July 2004.
- [PS07] J. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill, 5th edition, 2007.
- [Rad11] K. Rahnema Rad. Nearly sharp sufficient conditions on exact sparsity pattern recovery. *IEEE Trans. Inf. Theory*, 57(7):4672–4679, July 2011.
- [RU08] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.
- [San03] A. Sandberg. *Bayesian Attractor Neural Network Models of Memory*. PhD thesis, Stockholm University, 2003.

- [SBB06] S. Sarvotham, D. Baron, and R. G. Baraniuk. Measurements vs. bits: Compressed sensing meets information theory. In *Proc. 44th Annual Allerton Conf. on Commun., Control, and Computing*, September 2006.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, July 1948.
- [SSP96] F. Schwenker, F. T. Sommer, and G. Palm. Iterative retrieval of sparsely coded associative memory patterns. *Neural Networks*, 9(3), 1996.
- [Sto09] Mihailo Stojnic. Explicit thresholds for approximately sparse compressed sensing via ℓ_1 -optimization. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pages 478–482, June 2009.
- [SXC08] Ji Shihao, Ya Xue, and L. Carin. Bayesian compressive sensing. *IEEE Trans. Signal Processing*, 56(6):2346–2356, 2008.
- [SZ09] V. Saligrama and M. Zhao. Compressed sensing of autoregressive processes. In *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 7446, August 2009.
- [TV11] O. Taheri and S.A. Vorobyov. Segmented compressed sampling for analog-to-information conversion: Method and performance analysis. *IEEE Trans. Inf. Theory*, 59(2):554–572, February 2011.
- [VC10] M. Versace and B. Chandler. The brain of a new machine. *IEEE Spectrum Magazine*, 47(12), 2010.
- [Wai09] M. J. Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Trans. Inf. Theory*, 55(12):5728–5741, December 2009.

- [WCF07] Y. Weiss, H. S. Chang, and W. T. Freeman. Learning compressed sensing. In *Proc. 45th Annual Allerton Conf. on Commun., Control, and Computing*, September 2007.
- [WLW03] Y. Wang, D. Liu, and Y. Wang. Discovering the capacity of human memory. *Brain and Mind*, 4(2):189–198, 2003.
- [WWR10] Wei Wang, M.J. Wainwright, and K. Ramchandran. Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices. *IEEE Trans. Inf. Theory*, 56(6):2967–2979, June 2010.
- [YL03] Young-Han Kim and A. Lapidoth. On the log determinant of non-central wishart matrices. In *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2003.
- [YZ09] L. Ying and Yi Ming Zou. Linear transformations and restricted isometry property. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 2961–2964, 2009.
- [ZS10] M. Zhao and V. Saligrama. On compressed blind de-convolution of filtered sparse processes. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 4038–4041, March 2010.