Exploration of brain-inspired computing with self-organizing neuromorphic architectures

Lyes Khacef

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Exploration du calcul bio-inspiré avec des architectures neuromorphiques auto-organisées

Lyes Khacef

Laboratoire d'Electronique, Antennes et Télécommunications (LEAT)

Présentée en vue de l'obtention du grade de Docteur en Électronique **d'**Université Côte d'Azur

Dirigée par :
Benoît Miramond, Dr. Prof.
Co-encadrée par :
Laurent Rodriguez, Dr. MCF

Soutenue le : 15-12-2020

Devant le jury, composé de :
Frédéric Precioso, Dr. Prof., Université Côte d'Azur
Peter Ford Dominey, Dr. DR, CNRS
Damien Querlioz, Dr. CR, CNRS
Yulia Sandamirskaya, Dr. Chercheure, Intel Labs
Giacomo Indiveri, Dr. Prof., Université de Zurich
Laurent Rodriguez, Dr. MCF, Université Côte d'Azur
Benoît Miramond, Dr. Prof., Université Côte d'Azur
Elisabetta Chicca, Dr. Prof., Université de Groningen

Exploration du calcul bio-inspiré avec des architectures neuromorphiques auto-organisées

Exploration of brain-inspired computing with self-organizing neuromorphic architectures

Président du jury

- Frédéric Precioso, Dr. Professeur des Universités, Université Côte d'Azur.

Rapporteurs

- Peter Ford Dominey, Dr. Directeur de Recherche, CNRS.
- Damien Querlioz, Dr. Chargé de Recherche, CNRS.

Examinateurs

- Yulia Sandamirskaya, Dr. Chercheure Senior, Intel Labs.
- Giacomo Indiveri, Dr. Professeur des Universités, Université de Zurich.
- Laurent Rodriguez, Dr. Maitre de Conférences, Université Côte d'Azur.
- Benoît Miramond, Dr. Professeur des Universités, Université Côte d'Azur.

Invitée

- Elisabetta Chicca, Dr. Professeure des Universités, Université de Groningen.

Résumé

La plasticité corticale du cerveau est l'une des principales caractéristiques qui nous permettent d'apprendre et de nous adapter à notre environnement. En effet, le cortex cérébral a la capacité de s'auto-organiser grâce à deux formes de plasticité : la plasticité structurelle qui crée ou coupe les connexions synaptiques entre les neurones, et la plasticité synaptique qui modifie la force des connexions synaptiques. Ces mécanismes sont très probablement à la base d'une caractéristique extrêmement intéressante du développement du cerveau humain : l'association multimodale. Malgré la diversité des modalités sensorielles, comme la vue, le son et le toucher, le cerveau arrive aux mêmes concepts. De plus, les observations biologiques montrent qu'une modalité peut activer la représentation interne d'une autre modalité lorsque les deux sont corrélées. Pour modéliser un tel comportement, Edelman et Damasio ont proposé respectivement la réentrance et la zone de convergence/divergence où les communications neurales bidirectionnelles peuvent conduire à la fois à la fusion multimodale (convergence) et à l'activation intermodale (divergence). Néanmoins, ces cadres théoriques ne fournissent pas de modèle de calcul au niveau des neurones.

L'objectif de cette thèse est d'abord d'explorer les fondements de l'auto-organisation inspirée par le cerveau en termes (1) d'apprentissage multimodal non supervisé, (2) de calcul massivement parallèle, distribué et local, et (3) de traitement efficace sur le plan énergétique. Sur la base de ces lignes directrices et d'une étude des modèles neuronaux de la littérature, nous choisissons la carte auto-organisée (SOM) proposée par Kohonen comme composant principal de notre système. Nous introduisons la grille itérative, une architecture entièrement distribuée avec une connectivité locale entre les neurones matériels qui permet un calcul cellulaire dans le SOM, et donc un système qui passe à l'échelle en termes de temps de traitement et de connectivité. Ensuite, nous évaluons la performance du SOM dans le problème de l'apprentissage non supervisé post-étiqueté : aucun label n'est disponible pendant l'entrainement, puis très peu de labels sont disponibles pour étiqueter les neurones du SOM. Nous proposons et comparons différentes méthodes d'étiquetage afin de minimiser le nombre d'étiquettes tout en conservant la meilleure précision. Nous comparons nos performances à une approche différente utilisant des réseaux neuronaux à spike (SNN).

Ensuite, nous proposons d'améliorer les performances du SOM en utilisant des caractéristiques extraites au lieu de données brutes. Nous menons une étude comparative sur la classification du SOM avec extraction non-supervisée de caractéristiques à partir de la base de données MNIST en utilisant deux approches différentes : une approche d'apprentissage machine avec des auto-encodeurs convolutionnels et une approche bio-inspirée avec des SNN. Pour prouver la capacité du SOM à classifier des données plus complexes, nous utilisons l'apprentissage par transfert dvec la base de données mini-ImageNet.

Enfin, nous passons au mécanisme d'association multimodale. Nous construisons le modèle bio-inspiré ReSOM basé sur les principes de réentrance en utilisant les SOMs et l'apprentissage Hebbien. Nous proposons et comparons différentes

méthodes de calcul pour l'apprentissage et l'inférence multimodale non supervisée, puis nous quantifions le gain des mécanismes de convergence et de divergence sur trois bases de données multimodales. Le mécanisme de divergence est utilisé pour étiqueter une modalité à partir de l'autre, tandis que le mécanisme de convergence est utilisé pour améliorer la classification globale du système. Nous comparons nos résultats avec des SNNs, puis nous montrons le gain de la plasticité dite matérielle induite par notre modèle, où la topologie du système n'est pas fixée par l'utilisateur mais apprise au fil de l'expérience du système par l'auto-organisation.

Mots clés

Calcul bio-inspiré; réseaux de neurones artificiels; cartes auto-organisatrices réentrantes; apprentissage multimodal non-supervisé; architectures distribuées cellulaires; implémentation neuromorphique.

Résumé vulgarisé

L'auto-organisation, aussi appelée neuro-plasticité, c'est la capacité des neurones biologiques à créer, modifier ou défaire des connexions entre eux pour apprendre et s'adapter à l'environnement. En effet, l'intelligence est souvent définie comme cette capacité d'adaptation au changement à travers l'apprentissage. Dans cette thèse, je modélise des réseaux de neurones artificiels auto-organisés grâce à un mécanisme de plasticité structurelle pour créer ou couper des connections, ainsi qu'un mécanisme de plasticité synaptique qui permet de modifier la force de ces connections. Ainsi, le modèle que je propose est capable de calculer de manière distribuée, d'apprendre de manière non-supervisée et d'exploiter plusieurs modalités sensorielles, telle que la vision, l'audition et le toucher afin d'améliorer sa perception de l'environnement.

Abstract

Lyes KHACEF

Exploration of brain-inspired computing with self-organizing neuromorphic architectures

The brain's cortical plasticity is one of the main features that enable our capability to learn and adapt in our environment. Indeed, the cerebral cortex has the ability to self-organize itself through two distinct forms of plasticity: the structural plasticity that creates (sprouting) or cuts (pruning) synaptic connections between neurons, and the synaptic plasticity that modifies the synaptic connections strength. These mechanisms are very likely at the basis of an extremely interesting characteristic of the human brain development: the multimodal association. In spite of the diversity of the sensory modalities, like sight, sound and touch, the brain arrives at the same concepts. Moreover, biological observations show that one modality can activate the internal representation of another modality when both are correlated. To model such a behavior, Edelman and Damasio proposed respectively the Reentry and the Convergence Divergence Zone frameworks where bi-directional neural communications can lead to both multimodal fusion (convergence) and inter-modal activation (divergence). Nevertheless, these theoretical frameworks do not provide a computational model at the neuron level.

The objective of this thesis is first to explore the foundations of brain-inspired self-organization in terms of (1) multimodal unsupervised learning, (2) massively parallel, distributed and local computing, and (3) extremely energy-efficient processing. Based on these guidelines and a review of the neural models in the literature, we choose the Self-Organizing Map (SOM) proposed by Kohonen as the main component of our system. We introduce the Iterative Grid, a fully distributed architecture with local connectivity amongst hardware neurons which enables cellular computing in the SOM, and thus a scalable system is terms of processing time and connectivity complexity. Then, we assess the performance of the SOM in the problem of post-labeled unsupervised learning: no label is available during training, then very few labels are available for naming the SOM neurons. We propose and compare different labeling methods so that we minimize the number of labels while keeping the best accuracy. We compare our performance to a different approach using Spiking Neural Networks (SNNs) with Spike Timing Dependant Plasticity (STDP) learning.

Next, we propose to improve the SOM performance by using extracted features instead of raw data. We conduct a comparative study on the SOM classification accuracy with unsupervised feature extraction from the MNIST dataset using two different approaches: a machine learning approach with Sparse Convolutional Auto-Encoders using gradient-based learning, and a neuroscience approach with SNNs using STDP learning. To prove the SOM ability to handle more complex datasets, we use transfer learning in the mini-ImageNet few shot classification benchmark to exploit a Wide Residual Network backbone trained on a base dataset as a feature extractor, then we use the SOM to classify the obtained features from the target dataset.

Finally, we move into the multimodal association mechanism. We build the Reentrant SOM (ReSOM), a brain-inspired neural system based on the Reentry principles using SOMs and Hebbian-like learning. We propose and compare different computational methods for multimodal unsupervised learning and inference, then quantify the gain of both convergence and divergence mechanisms on three multimodal datasets. The divergence mechanism is used to label one modality based on the other, while the convergence mechanism is used to improve the overall accuracy of the system. We compare our results to SNNs with STDP learning and different fusion strategies, then we show the gain of the so-called hardware plasticity induced by our model, where the system's topology is not fixed by the user but learned along the system's experience through self-organization.

Keywords

Brain-inspired computing; artificial neural networks; reentrant self-organizing maps; multimodal unsupervised learning; cellular distributed architectures; neuromorphic implementation.

Vulgarized abstract

Self-organization, also called neuro-plasticity, is the ability of biological neurons to create, modify or cut connections amongst them in order to learn and adapt to the environment. Indeed, intelligence is often defined as the ability to adapt to change through learning. In this thesis, I model self-organizing artificial neural networks using a mechanism of structural plasticity to create or cut connections, as well as a mechanism of synaptic plasticity to modify the strength of these connections. Thus, the model I propose is able to compute in a distributed way, to learn in an unsupervised fashion and to exploit several sensory modalities such as sight, sound and touch in order to improve its perception of the environment.

"It's not about how hard you hit,
it's about how hard you can get hit
and keep moving forward…"

Rocky Balboa.

Acknowledgements

First, I want to thank my supervisor Benoit Miramond for the life-time opportunity to find what I love to do. That happens only once. I want to thank him and my co-supervisor Laurent Rodriguez for the guidance, for the countless advices and for the freedom to imagine my thesis as a reflection of my own vision for brain-inspired computing. They gave me the chance to express myself in everything I did, to try, fail and try again, and again, and again. This manuscript is the proof that it ended well, right? I mean, if there ever is or will be an end...

I want to thank my PhD defense jury: Damien Querlioz, Peter Ford Dominey, Giacomo Indiveri, Yulia Sandamirskaya, Frederic Precioso and Elisabetta Chicca, as well as Timothée Masquelier for being part of my PhD individual monitoring committee. I'm deeply honored and truly grateful. I want to thank Vincent Gripon and the SOMA project team with Andres Upegui and Nicolas Rougier. It has been a great opportunity to learn from them in every aspect. I also want to thank Stéphane Lallée for the very encouraging feedback.

I want to thank the amazing eBRAIN group: Nassim Abderrahmane with who I had my first paper, and who has always been there to talk about neurons, football and life, even though he forced me to move from PES to FIFA. I want to thank Yasmina Zaky, even though she's not in the eBRAIN team but I can't separate these two. I want to thank Adrien Russo Inass Bouhichia, couldn't even separate them with a comma, Alexis Arcaya Jordan, Marino Rasamuel, Edgar Lemaire, Pierre-Emmanuel Novac, Loic Cordone and all the others. I learned so much from these guys, and they all gave me some pieces for the puzzle of my thesis.

I want to thank the EDGE team: Alain Pegatoquet, François Verdier, Daniel Gaffé and the whole team for the wonderful team spirit. I want to thank the LEAT members, Nicolas Fortino for the continous help in my teachings, Léonardo Lizzi, Jean-Yves Dauvignac, Robert Staraj, all the permanent and non-permanent members and the beautiful secretary team, Françoise Trucas, Marie-Hélène Proscilico, Sophie Gaffé and Michele Grangier. I also want to thank Régine Saelens, Claire Migliaccio and the EDSTIC doctoral school team. These people's work ethics and thriving for excellence inspired me all the way. A special thank to Konstanze Beck for the help with the great Préfecture, Mohamed Al Khalfioui for the trust and confidence in my project, and to Anne-Laure Simonelli for the greatest leadership lessons.

I had the chance to go further out of my comfort zone and to meet some extraordinary people in my PhD journey. I want to thank the MT180 team and coaches, Laurie Chiara, Franck Rainaut, Ugo Bellagamba, Christophe Rousseau. It was a unique experience that made me construct the purpose of my thesis. I want to thank the BioComp Workshop people, a special thank to Pierre Falez with whom I shared many conferences and the craziest experience in Rio de Janeiro. I want to thank the CapoCaccia Neuromorphic Workshop people, a special thank to my team with Enea Ceolini, Charlotte Frenkel, Gemma Taverni, Melika Payvand and Elisa Donati. It was just great working together.

I've been lucky to be the president of the ADSTIC association for two years, where we gave our best to bring people together around science, football, Karaokes,

beers and Karaokes. Yeah, they love it so much when I sing. I want to thank the ADSTIC team and all our members that I did not mention before, Karyna Gogunska, Amina Ghrissi, Mohamad El Laz, Nathalie Gayraud and Yanis Boussad. A special thank to Rémy Garcia and the new team for taking over the association. And for cooking so well. I also want to thank the ADSFA and AJC06 members and friends, Auréa Cophignon, Sarah Toparslan and Sotherath Seang. A special thank to Claire Lasserre who has been there in some special moments. Together, we did something greater that the sum of what we can do individually.

Now, I want to thank these people who did not fall into the previous clusters: Yacine Dahmani, who was always there for supporting me during the last eight years, in the best and the worst moments. Yasmine Hareb, a living proof that the greatest friendship survives to distances. Yacine Messaoudène, who's still the same after more than fifteen years seg Useqqif gher Paris. Dorine Havyarimana, who made me overcome my darkest demons and pushed me forward in all circumstances. Without her, you would not be reading these words. Roland Kromes and his genuine friendship, I could not be more grateful. Ahmed Oualha and his guitare, Katarzyna Tomasiak and her Vodka, Mircea Moscu for the best flatmate ever (and the best jokes, too), Diana Resmerita for the endless support in my singing career, Paul Jégat for his endless unique energy. I want to thank Luc Guerits, Yassine Chouchane, Marta Ballatorre, Flora Zidane, Khai Nguyen and Luca Santamaria, as well as the six fantastics along with my brother, Julian Roqui, Jonathan Courtois, Lionel Tombakdjian, Floyd Bertagne and Walid Chekkar for the support and for showing us what real team-work is. I want to thank all these people for the great times together and the precious memories that will last forever.

Then, I want to thank a special person with a special connection, Zahra Hadjou. She showed me the courage it takes to be free, and that freedom is meant to be shared with the people we love. "It is your duty in life to save your dream", she thought me what Amedeo Modigliani meant by these words, and she was there to show me the way in the darkest moments. She was there the last day of the writing of my thesis, on September 29, 2020. She gave me the missing pieces to complete this manuscript.

Finally, I want to thank my family. Lina Khacef, she showed us how much we love each other. Melisa Khacef, she became a woman and succeeded in the most difficult times. Yacine Khacef, he proved to be a champion. These two are the best. My Mother, Nouara Khacef, she's the most courageous person in the world. My father, Belkacem Khacef, he's not an ordinary hero, but an extraordinary human being. My family gave me the love, the faith and the unconditional support that I needed to keep moving forward. They are the people that I want to make the most proud of me. This manuscript is dedicated to them.

All of these people and many others that I did not mention helped me grow as a researcher, a leader, a team player, a friend and a person. "You can overcome anything, if and only if you love something enough", Lionel Messi was right. I realize, though, that we sometimes need some people to remind us of why we love what we do. I do research and science to have a great impact, and I have been lucky to have the people that I mentioned here to remind me of that.

Last but not least, I want to thank you for reading these words. Here and now, they are alive because of you, and whether your name was mentioned here or not, you are part of it. Tanemmirt.

With all my love,

Lyes Khacef, Dr.

Contents

List of Figures

List of Tables

List of Abbreviations

AE	**A**uto**E**ncoder
AER	**A**ddress **E**vent **R**epresentation
AI	**A**rtificial **I**ntelligence
ALU	**A**rithmetic and **L**ogic **U**nit
ANN	**A**rtificial **N**eural **N**etwork
AQE	**A**verage **Q**uantization **E**rror
BAM	**B**idirectional **A**ssociative **M**emory
BMU	**B**est **M**atching **U**nit
CA	**C**ellular **A**utomata
CAE	**C**onvolutional **A**uto-**E**ncoder
CDZ	**C**onvergence **D**ivergence **Z**one
CNN	**C**onvolutional **N**eural **N**etwork
CPU	**C**entral **P**rocessing **U**nit
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etworks
DSOM	**D**ynamic **S**elf-**O**rganizing **M**ap
DVS	**D**ynamic **V**ision **S**ensor
DoG	**D**ifference **o**f **G**aussians
EMG	**E**lectro**M**yo**G**raphy
FPGA	**F**ield-**P**rogrammable **G**ate **A**rray
FSM	**F**inite **S**tate **M**achine
GNG	**G**rowing **N**eural **G**as
GPU	**G**raphical **P**rocessing **U**nit
GSC	**G**oogle **S**peech **C**ommands
GSOM	**G**rowing **S**elf-**O**rganizing **M**ap
GWR	**G**rowing **W**hen **R**equired
HMI	**H**uman-**M**achine **I**nteraction
HRI	**H**uman-**R**obot **I**nteraction
IG	**I**terative **G**rid
INN	**I**ncremental **N**eural **N**etwork
IoT	**I**nternet of **T**hings
LIF	**L**eaky **I**ntegrate-and-**F**ire
LTD	**L**ong-**T**erm **D**epression
LTP	**L**ong-**T**erm **P**otentiation
MAV	**M**ean **A**bsolute **V**alue
MFCC	**M**el **F**requency **C**epstral **C**oefficients
ML	**M**achine **L**earning
MMCM	**M**ulti-**M**odal **C**onvergence **M**ap
MNIST	**M**ixed **N**ational **I**nstitute of **S**tandards and **T**echnology
MSE	**M**ean **S**quared **E**rror
NG	**N**eural **G**as
NPU	**N**eural **P**rocessing **U**nit
NoC	**N**etwork-**o**n-**C**hip

PCA	**P**rincipal **C**omponent **A**nalysis
PCSOM	**P**runing **C**ellular **S**elf-**O**rganizing **M**ap
PDP	**P**arallel and **D**istributed **P**rocessing
RMS	**R**oot **M**ean **S**quare
ReSOM	**Re**entrant **S**elf-**O**rganizing **M**ap
SCAE	**S**parse **C**onvolutional **A**uto-**E**ncoder
SNN	**S**piking **N**eural **N**etwork
SOIMA	**S**elf-**O**rganized **I**nternal **M**odels **A**rchitecture
SOM	**S**elf-**O**rganizing **M**ap
STDP	**S**pike-**T**iming-**D**ependent **P**lasticity
SVM	**S**upport **V**ector **M**achine
S-MNIST	**S**poken**-MNIST**
TF	**T**ensor**F**low
TPU	**T**ensor **P**rocessing **U**nit
VDSOM	**V**arying **D**ensity **S**elf-**O**rganizing **M**ap
VLSI	**V**ery **L**arge **S**cale **I**ntegration
WMU	**W**orst **M**atching **U**nit
WRN	**W**ide **R**esidual **N**etwork
WTA	**W**inner-**T**akes-**A**ll

Author's Publication List

Related journal paper

- Lyes Khacef, Laurent Rodriguez, and Benoît Miramond (2020c). "Brain-Inspired Self-Organization with Cellular Neuromorphic Computing for Multimodal Unsupervised Learning". In: *Electronics* 9.10. ISSN: 2079-9292. DOI: 10.3390/electronics9101605. URL: https://www.mdpi.com/2079-9292/9/10/1605.

Related patents

- Lyes Khacef, Laurent Rodriguez, and Benoit Miramond (2020b). "Method and System for multimodal classification based on brain-inspired unsupervised learning". In: *International Patent (Submitted)*.
- Laurent Rodriguez, Lyes Khacef, and Benoit Miramond (2020). "Distributed Cellular Computing System and Method for neural-based Self-Organizing Maps". In: *International Patent (Submitted)*.

Related conference papers

- Lyes Khacef, Vincent Gripon, and Benoît Miramond (2020). "GPU-Based Self-Organizing Maps for Post-labeled Few-Shot Unsupervised Learning". In: *Neural Information Processing*. Ed. by Haiqin Yang et al. Cham: Springer International Publishing, pp. 404–416. ISBN: 978-3-030-63833-7.
- Lyes Khacef, Laurent Rodriguez, and Benoît Miramond (2020a). "Improving Self-Organizing Maps with Unsupervised Feature Extraction". In: *Neural Information Processing*. Ed. by Haiqin Yang et al. Cham: Springer International Publishing, pp. 474–486. ISBN: 978-3-030-63833-7.
- L. Khacef et al. (2019). "Self-organizing neurons: toward brain-inspired unsupervised learning". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. DOI: 10.1109/IJCNN.2019.8852098.
- Lyes Khacef et al. (2018). "Neuromorphic hardware as a self-organizing computing system". In: *2018 IJCNN Neuromorphic Hardware In Practice and Use workshop*. URL: https://arxiv.org/abs/1810.12640.
- L. Rodriguez, L. Khacef, and B. Miramond (2018). "A distributed cellular approach of large scale SOM models for hardware implementation". In: *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 250–255.

Related database

- Lyes Khacef, Laurent Rodriguez, and Benoit Miramond (Oct. 2019). *Written and spoken digits database for multimodal learning*. Version 1.0. DOI: 10.5281/zenodo.3515935. URL: https://doi.org/10.5281/zenodo.3515935.

Unrelated journal paper

- Enea Ceolini et al. (2020). "Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing". In: *Frontiers in Neuroscience* 14, p. 637. ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00637. URL: https://www.frontiersin.org/article/10.3389/fnins.2020.00637.

Unrelated conference papers

- Julian Roqui et al. (2020). "Estimation of Small Antenna Performance Using a Machine Learning Approach". In: *2020 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*.
- M. Rasamuel et al. (2019). "Specialized visual sensor coupled to a dynamic neural field for embedded attentional process". In: *2019 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6.
- E. Ceolini et al. (2019a). "Live Demostration: Sensor fusion using EMG and vision for hand gesture classification in mobile applications". In: *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–1. DOI: 10.1109/BIOCAS.2019.8919163.
- E. Ceolini et al. (2019b). "Sensor fusion using EMG and vision for hand gesture classification in mobile applications". In: *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4. DOI: 10.1109/BIOCAS.2019.8919210.
- L. Khacef, N. Abderrahmane, and B. Miramond (2018). "Confronting machine-learning with neuroscience for neuromorphic architectures design". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. DOI: 10.1109/IJCNN.2018.8489241.

Chapter 1

Introduction, context and motivation

1NT3LL1G3NC3 1S 7H3 4B1L17Y 7O 4D4P7 7O CH4NG3.

ST3PH3N H4WK1NG.

1.1 Brain-inspired self-organization

"The brain is the seat of intelligence". This affirmation was first made in the antiquity (around the 5th century BCE) by Alcmaeon of Croton, a Greek medical writer and philosopher-scientist (Huffman, 2017). He was the first to identify the brain as the seat of understanding and to distinguish understanding from perception. Alcmaeon thought that the sensory organs were connected to the brain by channels and may have discovered the optic nerve that connects the eyes to the brain. We know since then that our brain gives rise to our perceptions, memories, thoughts and actions. However, more than 2,500 years later, we still do not know how these phenomena precisely arise in the brain, and it is considered to be the greatest scientific mystery and challenge of our time (Maass et al., 2019). The only plausible hypothesis that has been strongly suggested over the decades (Marr, 1982) (Valiant, 1984) (Hawkins and Blakeslee, 2004) is that the answer to this question will be at least partly computational (Maass et al., 2019). In this manuscript, we will refer to the brain as a computational system based on neurons and synapses and try to explore its principles to propose a model for brain-inspired computing.

Of course, the goal is ambitious and we do not intend to build an artificial brain, but rather to get inspiration from how the biological brain works and try to use it to approach a certain behavior at a low hardware cost. Indeed, despite the huge scientific and technological advancements in the last decades, we are still unable to produce *intelligent artificial systems* that can autonomously learn and adapt to their environment or integrate 86 billion neurons (Herculano-Houzel, 2009) with 10 thousand connections each at the cost of 20 watts. One main reason for this is certainly grounded in the way biological systems are built, which is very different from the traditional, human way of building things (Bauer, 2013). Instead of having a "construction blueprint" that is implemented by an external observer, biological organisms develop as a result of local cellular behaviors which are specified in the genetic code, and every process relies solely on local information exchange with no global controller or supervisor. While the general structure of the brain is similar in all individuals of the same species (Pfister et al., 2018), local synaptic connections and their

reinforcement are more dependent on experience and interaction with the environment (Varela, Rosch, and Thompson, 1991). The particular case of subjects with early lesions clearly shows that even with different general structures, some individuals are capable of similar tasks (Rita and W. Kercel, 2003).

Early research into the brain revealed a structure comprising a complicated intercommunicating network of billions of neurons, with a relatively simple structure for the neuron itself as modeled by Mcculloch and Pitts (Mcculloch and Pitts, 1943). The field of connectionism (Buckner and Garson, 2019) was thereafter fostered upon the idea that global intelligent behavior, such as memory and pattern recognition emerges from local interactions among a large number of simple processing units working independently. This is the computational basis of *self-organization*. While the simple neuron model has long been overthrown in the neuroscience community, the success of Artificial Neural Networks (ANNs) in diverse application areas has ensured continued interest in them among researchers from different disciplines in Artificial Intelligence (AI) (Ranganathan and Kira, 2003). This self-organized development holds the key to the unique characteristics of the brain (Bauer, 2013). It also leads to the notion of evolution which not only applies to nature but also to artificial systems as long as they are guided by some principle of self-organization (Richter, 1994). After these findings of self-organization in the brain were discovered, they were embraced by the AI community as something that could provide clues as to what intelligence really is (Ranganathan and Kira, 2003). Therefore, understanding brain's self-organization and natural development would extend our biological knowledge but also enable unprecedented technological progress.

This is the main motivation of the Self-Organizing Machine Architecture (SOMA) project where this thesis takes place. The objective of the project is to study neural-based self-organization in computing systems and to prove the feasibility of a self-organizing hardware structure. Today, several current issues such as analysis and classification of major data sources (sensor fusion, Internet of Things, etc.) and the need for adaptability in many application areas (automotive systems, autonomous drones, space exploration, etc.) lead us to study a desirable property from the brain that encompasses all others: the cortical plasticity. This term refers to one of the main developmental properties of the brain where the organization of its structure (structural plasticity) and the learning of the environment (synaptic plasticity) develop simultaneously toward an optimal computing efficiency. In other words, the cortical plasticity enables the self-organization in the brain, that in turn enables the emergence of consistent representations of the world (Varela, Rosch, and Thompson, 1991). We claim that the expected properties of such alternative computing devices could emerge from a close interaction between neural processing (self-organization and adaptation) and cellular computing (decentralization and hardware compliant massive parallelism). Therefore, we propose to combine both principles through a neuro-cellular approach of structural and synaptic self-organization that defines a fully distributed and self-organizing neuromorphic architecture.

1.2 Multimodal unsupervised learning

Intelligence is often defined as the ability to adapt to the environment through learning. "A person possesses intelligence insofar as he has learned, or can learn, to adjust himself to his environment", S. S. Colvin quoted in (Sternberg, 2000). The same definition could be applied to machines and artificial systems in general. Hence, a stronger relationship with the environment is a key challenge for future intelligent

artificial systems that interact in the real-world environment for diverse applications like object detection and recognition, tracking, navigation, etc. The system becomes an "agent" in which the so-called intelligence would emerge from the interaction it has with the environment, as defined in the *embodiement* hypothesis that is widely adopted in philosophy (Clark, 2001), cognitive science (Damasio, 1994), cognitive neuroscience (Varela, Rosch, and Thompson, 1991), developmental psychology (Smith and Gasser, 2005) and developmental robotics (Droniou, Ivaldi, and Sigaud, 2015). In this thesis, we tackle the first of the six fundamental principles for the development of embodied intelligence as defined in (Smith and Gasser, 2005): the multimodality. Indeed, the brain uses multiples sensory and motor modalities to perceive and act on its environment. But how does the brain handle multimodal association? In fact, it is most likely the emergent result of one of the most impressive abilities of the embodied brain that we previously discussed: the self-organization which is enabled by cortical (structural and synaptic) plasticity.

These principles could apply to biological as well as artificial systems, because both can acquire information about the information through various biological and artificial sensors, and the same way act on it. Multimodal data fusion is in fact a direct consequence of the well-accepted paradigm that certain natural processes and phenomena are expressed under completely different physical guises, each of which brings different but complementary information to the others (Baltrusaitis, Ahuja, and Morency, 2019). Multimodal information processing is a vital condition in order for AI to make progress in understanding the world around us. Multimodal ML is thus a vibrant multidisciplinary field of increasing importance with a great potential. Recent works show a growing interest toward multimodal association in several applicative areas such as developmental robotics (Lallee and Dominey, 2013) (Droniou, Ivaldi, and Sigaud, 2015), audio-visual signal processing (Shivappa, Trivedi, and Rao, 2010) (Rivet et al., 2014), spacial perception (Pitti et al., 2012) (Fiack, Cuperlier, and Miramond, 2015), attention-driven selection (Braun et al., 2019) and tracking (Zhao and Zeng, 2019), memory encoding (Tan et al., 2019), emotion recognition (Zhang, Wang, and Du, 2019) (Mansouri-Benssassi and Ye, 2020), human-machine interaction (Turk, 2014), remote sensing and earth observation (Debes et al., 2014), medical diagnosis (Hoeks et al., 2011), understanding brain functionality (Horwitz and Poeppel, 2002), etc.

Following a brain-inspired approach, we couple multimodal association with unsupervised learning. Even though the brain may exhibit several forms of learning, namely supervised learning in the cerebellum, reinforcement learning in the basal ganglia and unsupervised learning in the cerebral cortex (Doya, 1999), the brain seems to be mostly unsupervised (Dayan, 1999). In 1949, Donald Hebb was the first to link neuro-biological experiments on plasticity to a purely unsupervised statistical method (Hebb, 1949). Today, many in the ANN community, including pioneers of DL such as Yann Lecun and Geoffray Hinton, claim that we rely primarily on unsupervised paradigms to construct our representations of the world (Zador, 2019). This is interesting because, indeed, unsupervised learning is becoming one of the most important challenges in ML and AI, as we gather more and more data everyday but we cannot annotate each sample of them. The possibility to learn useful representations from the raw data without labels would create incredible opportunities for many application areas. Hence, from a biological plausibility perspective as well as from the purely pragmatic point of view, unsupervised learning is of great interest. Nevertheless, unsupervised learning is most of the time of a less good performance that supervised learning, especially for classification tasks. We argue here

that the gap of performance between unsupervised learning and supervised learning could be reduced when using multiple modalities that complement each other to improve the overall accuracy of the system. This is one of the most important questions that we want to tackle in this thesis.

1.3 Neuromorphic engineering

ANNs are experiencing today an unprecedented interest in both research and industry thanks to two main changes: the explosion of open data that is necessary for their training, and the increasing computing power of today's computers that makes the training part possible in a reasonable time. The recent results of neural-based Deep Learning (DL) on various classification tasks has given ANNs the leading role in Machine Learning (ML) algorithms and AI technologies. However, in addition to the limits of supervised learning discussed before, most applications such as smart devices or autonomous vehicles require an embedded implementation of ANNs for real-time processing *on the edge*. Their implementation in conventional Von Neumann architectures (von Neumann, 1993) such as CPU and GPU remains too expensive, mostly in energy consumption. This is due to the non-adaptation of the centralized hardware to the distributed computation model. We are today at a turning point, where Moore's law is reaching its end leading to a stagnation of the performance of our computers. Therefore, the research community came to the conclusion that AI needs new hardware, not just new algorithms (Strukov et al., 2019). That's how neuromorphic engineering was born.

In fact, the idea of a brain-inspired computing machine with physical parallel and distributed components was already imagined by Alan Turing. In addition to the work leading to the digital computer, Turing anticipated connectionism and neuron-like computing. In 1948, Turing described in his paper entitled "Intelligent machinery" (Turing, 1948) a machine that consists of artificial neurons connected in any pattern with modifier devices that could be configured to pass or block a signal. The neurons were composed of NAND gates that Turing chose because they are universal gates, i.e. any other gate can be represented as a combination of NAND gates. Nevertheless, it is in the late 1980s that Carver Mead, professor of electrical engineering and computer science at Caltech published "Analog VLSI and Neural Systems" (Mead, 1989) and developed the concept of neuromorphic engineering (Mead, 1990), also known as neuromorphic computing. He described how to design Very Large Scale Integration (VLSI) systems containing electronic analog circuits to mimic neuro-biological architectures present in the nervous system. In recent years, the term neuromorphic has been used to describe analog (Indiveri et al., 2011) (Chicca et al., 2014), digital (Merolla et al., 2014) (Davies et al., 2018), mixed-mode analog/digital (Schemmel et al., 2010) (Moradi et al., 2018) and emerging technologies such as Resistive RAMs and memristors (Bichler et al., 2012) (Querlioz et al., 2013) that implement models of neural systems for perception, motor control, or multisensory integration.

These efforts in both research and industry attest to the necessity of designing neuromorphic architectures, i.e. hardware accelerators that fit to the Parallel and Distributed Processing (PDP) paradigm of neural networks for reducing their hardware cost implementation (Schuman et al., 2017). We tackle this issue by proposing a fully distributed and cellular neuromorphic architecture with local connectivity, which is at the same time the essence of self-organization but also the path toward more energy-efficient implementations. Furthermore, we introduce in this thesis the

concept of *hardware plasticity*, which is the hardware implementation of the structural plasticity for sprouting (creating) and pruning (cutting) synaptic connection amongst neurons. Indeed, as previously discussed, these plasticity mechanisms are the core of self-organization in the brain, and they could enable neuromorphic circuits to calibrate themselves without the need of an external supervisor. The hardware issue addressed by the SOMA project is how the communications within and between the dynamic computing areas self-organize by means of a particular type of dynamically re-configurable Network-on-Chip (NoC) (Moraes et al., 2004) controlled by the neural network, thus transposing structural plasticity principles onto FPGA hardware (Vannel et al., 2018) (Khacef et al., 2018). Even though the hardware implementation is not part of this thesis, the proposed neural model and cellular neuromorphic architecture will be consequent steps toward it. Both contributions will mainly answer the question of how to use structural and synaptic plasticities to build a self-organizing neural system capable of learning from multiple modalities in an unsupervised fashion, and what is the gain of the multimodal association and the so-called hardware plasticity compared to standard approaches.

1.4 Outline

This manuscript is organized as follows: Chapter 2 summarizes the foundations of brain-inspired computing which will guide us to choose the Self-Organizing Map (SOM) as a principal component of the proposed self-organnizing neural system. It also introduces the Iterative Grid, a cellular neuromorphic architecture proposed to distribute the SOM computation with local connectivity. Chapter 3 introduces the post-labeled unsupervised learning problem, then presents the proposed labeling method based on very few labels and confronts SOMs to Spiking Neural Networks (SNNs) in terms of classification accuracy on MNIST and scalability in terms of time and connectivity complexities. Chapter 4 compares different methods for feature extraction in order to improve the SOM performance on MNIST classification, namely AutoEncoders (AEs) and SNNs. Next, it assesses the SOM performance on miniImageNet with transfer learning to figure out if the SOM can be used in the real-world environment to classify complex data. Chapter 5 proposes the Reentrant Self-Organizing Map (ReSOM), a new brain-inspired computational model of self-organization for multimodal unsupervised learning in neuromorphic systems. It first describes the Reentry framework of Edelman (Edelman, 1982) and the Convergence Divergence Zone framework of Damasio (Damasio, 1989), two different theories in cognitive neuroscience for modeling multimodal association in the brain, and then reviews some of their recent computational models and applications. Afterwards, it introduces the ReSOM model and the proposed multimodal unsupervised learning and inference algorithms. Chapter 6 presents the databases, experiments and results with three different case studies, then it relates the ReSOM learning paradigm as well as the convergence and divergence mechanisms to studies on infants development. Finally, it discusses the results and quantifies the gain of the multimodal association and the so-called hardware plasticity through self-organization. Chapter 7 concludes this manuscript with a discussion on the most salient results and the different perspectives for the future works.

Chapter 2

Brain-inspired computing and self-organization

> The whole is greater than the sum of its parts.
>
> *Aristotle.*

2.1 Introduction

Brain-inspired computing is a broadly interdisciplinary field which may refer to multiple paradigms and concepts, some of which we discussed in chapter 1. It may thus lead to some confusion depending from which domain and perspective it is seen. In this chapter, we define *our* foundations for brain-inspired computing and present the biologically plausible properties that will guide us toward the choice of the main component of the proposed self-organizing neural system. Afterwards, we review the main computational models of neural self-organization, and we then choose the one that fits the most for unimodal processing, i.e. processing the information of an independent modality alone. Finally, we present the Iterative Grid (IG), a cellular neuromorphic architecture proposed to distribute the model's computation with local connectivity, a necessary step at this point because the hardware scalability of the model is a necessary condition before going further in the multimodal framework modeling and experimentation process.

2.2 Brain-inspired computing foundations

Brain-inspired computing can be described at different levels of abstraction, going from the hardware implementation into the overall behavior of the system. In order to clearly present the different paradigms and concepts that we claim to be at the foundations of brain-inspired computing and self-organization, we use the taxonomy proposed by David Marr in 1976 (Marr and Poggio, 1976). It is a three-level approach to understand brain's computation as summarized in (Maass et al., 2019):

- At the *behavioural* level, we describe the input-output behavior of the system, i.e. what the system does in a particular context.
- At the *algorithmic* level, we explain the organizations and dynamics of the particular processes used by the system, i.e. how does the system compute.
- At the *hardware* level: we identify the physical elements like Neural Processing Units (NPUs) employed by the system to realize the algorithm.

2.2.1 Behavioral level: embodied computing toward adaptation

At the behavioral level, a brain-inspired computational system should be able to learn in an *unsupervised* fashion by exploiting the *multimodal* information in the environment through a constant *sensori-motor* interaction with it. In this section, we present the biological observations of these three mechanisms and explain why we need them concretely.

Unsupervised learning

It has been shown that the response tuning of the neurons in the cortex is highly dependent on the sensory experience (Blakemore and COOPER, 1970) (Hirsch and Spinelli, 1970). It suggests that the information coding in the cerebral cortical areas is established by the unsupervised learning paradigm in which the synapses are updated by a Hebbian rule (Doya, 1999). In summary, the "learner" must build a probabilistic model of given inputs and use it to generate a recognition distribution for a specific input (Ranganathan and Kira, 2003). In addition to the bioligcal plausibility, unsupervised learning is extremely useful because it would only require the data without the labels. Today, DL models that reach the best performance in classification tasks are mostly based on supervised learning, which implies building huge labeled datasets to implement the gradient-based back-propagation training (Lecun, Bengio, and Hinton, 2015). It suggests that we have to label each sample of each training database depending on the application, and that approach can obviously not be generalized to all types of data and applications. Moreover, it is unlikely that such an algorithm based on neuron-specific error signal would be implemented in the brain (O'Reilly and Munakata, 2000). Hinton, one of the pioneers of DL, acknowledges that "as a biological model, back-propagation is implausible" (Hinton, 1989). Therefore, the neural model we propose should rely on unsupervised learning for the synaptic weights adaptation.

Multimodal association

Multimodality is the first principle for the development of embodied intelligence (Smith and Gasser, 2005). Indeed, biological systems perceive their environment through diverse sensory channels: vision, audition, touch, smell, proprioception, etc. The fundamental reason lies in the concept of degeneracy in neural structures (Edelman, 1987), which is defined by Edelman as the ability of biological elements that are structurally different to perform the same function or yield the same output (Edelman and Gally, 2001). In other words, it means that any single function in the brain can be carried out by more than one configuration of neural signals, so that the system still functions with the loss of one component. It also means that sensory systems can educate each other, without an external teacher (Smith and Gasser, 2005). The same principles can be applied for artificial systems, since the information about the same phenomenon in the environment can be acquired from various types of sensors: cameras, microphones, accelerometers, etc. Each sensory-information can be considered as a modality. Due to the rich characteristics of natural phenomena, it is rare that a single modality provides a complete representation of the phenomenon of interest (Lahat, Adali, and Jutten, 2015). Hence, the multimodality is necessary to have a complete representation of the environment.

Sensori-motor interaction

Another principle for the development of embodied intelligence as defined in (Smith and Gasser, 2005) is the sensory-motor loop or interaction. "The intelligence of babies resides not just inside themselves but is distributed across their interactions and experiences in the physical world" (Smith and Gasser, 2005). This physical world in which we live is rich in regularities that organize perception and action through a continual life-long interaction. This behavior is expecially interesting in the context of robotics, where robotic "agents" have to acquire knowledge via the interaction with their environment. In such a case; learning from the sensori-motor experience would result in a more efficient strategy for a life-long perspective (Droniou, Ivaldi, and Sigaud, 2015). However, the sensori-motor interaction is out of the scope of this thesis. We focus on multimodal perception, and argue that motor skills can be considered as an additional modality but would require a robotic platform or simulation for experimentation (Lallee and Dominey, 2013). This will be discussed in chapter 7.

2.2.2 Algorithmic level: cellular computing toward emergence

The brain is made of billions of neurons with ten thousands of connections each. It forms a self-organizing biological system that relies on local computation which is distributed amongst neurons. There is no central unit that orchestrates the process. The structural and synaptic plasticities that enable the self-organization only depend on local information stored between correlated neurons that can directly connect and communicate to each other, following a Hebbian learning paradigm (Hebb, 1949). The adaptation to the environment is thereby the *emergent* global behavior of the local computations. Such a mechanism is a fundamental aspect for brain-inspired computing, as it is important for both the behavior and the implementation of the self-organizing neural system. Indeed, the self-organization impacts the hardware-efficiency of the system, since the neurons inter-connections are not fixed by the designer but learned via structural and synaptic plasticities. From an algorithmic point of view, self-organization can then be defined as the formation of patterns and structures from the initial state without intervention through the interaction of finite state automata (Bremermann, 1994), or cellular automata. Stephen Wolfram defines cellular automata as “discrete dynamical systems with simple construction but complex self-organizing behaviour” (Wolfram, 1984b). In fact, simple nearest neighbour rules of cellular automata may simulate the complexity of universal computers (Wolfram, 1984a) (Cook, 2004). Therefore, we follow a cellular automata approach where each node represents a neuron to ensure a local and distributed computing.

2.2.3 Hardware level: embedded computing toward efficiency

The idea of neuromorphic engineering is to take inspiration from the brain for designing dedicated chips that merge memory and processing in a distributed non-Von Neumann architecture. In the brain, synapses provide a direct memory access to the neurons that process information. That’s how the brain achieves impressive computational power and speed with very little power consumption (Strukov et al., 2019). Neuromorphic engineering tries to imitate such an architecture for designing dedicated NPUs that are hardware-efficient in terms of electronic components, power consumption and latency. In sum, it leads to extremely energy-efficient processing which is needed for embedded systems where energy is very limited. We merge the cellular automata paradigm and neuromorphic engineering principles to design the

IG, a cellular neuromorphic architecture that supports self-organization to adapt to its environment, i.e. learn useful representations from multimodal information and use them to classify new inputs. The IG will be prototyped into multi-FPGA devices based on the work of Vannel et al. (Vannel et al., 2018), but the hardware implementation is out of the scope of this thesis: we focus on the modeling of the neural system and its architectural design based on the IG substrata.

2.3 Computational models for neural self-organization: historical overview

In this section, we review the most influent ANNs models of self-organization and choose the one that suits best as a main component for the new model that we propose in this thesis.

2.3.1 Cognitron (1975) and Neocognitron (1980)

One of the first models of neural self-organization is the Cognitron (Fukushima, 1975) proposed by Kunihiko Fukushima in 1975. He introduced a new hypothesis for the organization of synapses between neurons in a multilayered neural network, summarized in the following: "The synapse from neuron x to neuron y is reinforced when x fires provided that no neuron in the vicinity of y is firing stronger than y". The primary drawback with the Cognitron model was the high sensitivity to shift in position or any distortion in shape of the input. 15 years before in the 1960's, Hubel and Wiesel have already introduced the concept of simple cells (S-cells) and complex cells (C-cells) as a biological model for the structure of the visual cortex (Hubel and Wiesel, 1959). This model inspired Fukushima's new model in 1980: the Neocognitron (Fukushima, 1980). The Fukushima's Neocognitron divides indeed the neural network layers into S-cell layers and the C-cell layers. The input connections to the S-cell are plastic and could have their weights modified or even cut. The S-cells acted as input to the C-cells that were in contrary static and could not be varied. This way, the S-cell layers are responsible for forming weights to recognize patterns (synaptic strengths are adjusted layer by layer following the original Cognitron self-organization) while the C-cell layers ensure that the recognition was possible even after a shift of position or distortion in shape for the input. The Neocognitron is the predecessor of modern Convolutional Neural Networks (CNNs): the S-cell layer became the convolution layer while the C-cell layer became the max pooling layer.

2.3.2 Self-Organizing Map (SOM) (1982)

The Self-Organizing Map (SOM) algorithm was proposed by Teuvo Kohonen in 1982 (Kohonen, 1982) as a computational model for synaptic plasticity in the brain. It is one of the most popular models in the field of computational neurosciences, as it gives a plausible account on the self-organization of sensory areas in the cortex where adjacent neurons share similar representations (Kohonen, 1990). Indeed, it is possible to identify seemingly specialized cortical areas that, for example, encode information about faces (Strukov et al., 2019). The SOM defines an ANN in a two-dimentional map topology where each neuron is connected to the input. Technically, the SOM is a vector quantization algorithm, as it models the probability density function of the training dataset into a set of prototype vectors that are represented by the neurons afferent weights (Kohonen, Schroeder, and Huang, 2001) (Rougier

and Boniface, 2011). SOMs apply unsupervised learning in a form of competitive learning that uses a neighborhood function to preserve the topological properties of the input space. Basically, for each input stimulus, the SOM neurons compete to "win", i.e. to be the Best Matching Unit (BMU) that is the neuron representing best the stimulus. Then, the winning neuron and its neighborhood neurons adapt their afferent synaptic weights in order to improve the representation of the input pattern as detailed in section 2.4. The idea is that each neuron becomes a "prototype" that represents an average of similar-enough inputs. Many variants of the SOM have been proposed like the Receptive Field Laterally Interconnected Synergetically SOM (RF-LISSOM) (Miikkulainen et al., 1997) where the neurons receive inputs from local receptive fields on the input instead of the entire input, and are also laterally connected via excitatory and inhibitory synapses.

2.3.3 Neural Gas (1991) and Growing Neural Gas (1995)

The Neural Gas (NG) is an ANN inspired by the SOM and introduced in 1991 by Thomas Martinetz and Klaus Schulten (Martinetz and Schulten, 1991). The NG is an algorithm for finding optimal data representations based on feature vectors. The main difference with the SOM is that the modulation of learning in the NG is not dependent on the structure of the network, it is in the reverse the structure of the network that becomes dependent on the distance of each prototype from the stimuli. The algorithm was indeed called a "neural gas" because of the dynamics of the neurons during learning such that they distribute themselves like a gas within the data space. In 1995, Bernd Fritzke introduced the Growing NG (GNG) (Fritzke, 1994) as an incremental network model that learns topological relations by using a Hebbian-like learning. The GNG follows the same concept as the NG with two main differences: first, unlike the NG, the GNG has no hyper-parameters that change over time, it is hence more dynamic and capable of continuous learning. Second, the GNG can "grow" as indicated in its name, i.e. create neurons or inversely cut them when necessary, depending on the error to the input stimuli. The GNG grows and learns to represent the data until a given criterion is satisfied, e.g. a number of epochs or a maximum number of neurons.

2.3.4 Spiking Neural Network (SNN) with Spike-Timing-Dependent Plasticity (STDP) (1997)

Spiking Neural Networks are a particular model of ANNs in terms of information coding, as they use impulsion-based or *spike*-based coding amongst neurons, as further discussed in chapter 3. With such a coding scheme where information about the neurons activities is not frame-based but event-based, the time of the spike emission between two neurons can be used to modify the weight of the synapse connecting them. This is the basic principle of Spike-Timing-Dependent Plasticity (STDP) introduced by Henry Markram in 1997 (Markram et al., 1997), even though it is in 2000 that Sen Song et al. formalized the mechanism and named it as STDP (Song, Miller, and Abbott, 2000). In short, STDP is a form of Hebbian learning where the strengthening of a synapse occurs not only when two neurons spike at the same time, but also when one spikes just before the other as further detailed in chapter 3. It introduces therefore a form of causality in a particular topology. Indeed, Markram showed that in neocortical slices, Long-Term Potentiation (LTP) of synapses occurs if pre-synaptic spikes precede post-synaptic firing by no more than about 50 ms. In the reverse, Long-Term Depression (LTD) of synapses occurs when pre-synaptic

spikes follow the post-synaptic spikes (Markram et al., 1997). As all Hebbian models of development and learning, STDP requires both an activity-dependent synaptic plasticity and a competition mechanism (Song, Miller, and Abbott, 2000). This competition was further discussed and deployed in the form a Winner-Takes-All (WTA) mechanism in more recent works including Querlioz et al. in 2013 (Querlioz et al., 2013) and Diehl and Cook in 2015 (Diehl and Cook, 2015). The WTA induces a global inhibition so that each neuron learns a different pattern. A "soft" WTA mechanism inspired from the SOM mechanism was introduced by Hazan et al. in 2018 (Hazan et al., 2018), where the authors showed that it reaches a better performance in a classification task. It is further described in chapter 3.

2.3.5 Summary

We have reviewed the most important models of self-organization that exhibit some of the brain-inspired computing foundations that we previously discussed in section 2.2. At this point, we have to choose the model that will be used as the main component for unimodal processing in our multimodal self-organizing neural system. Thus, the model has to learn useful representations from unlabeled data and use them to classify new samples. The Neocognitron is therefore not a suitable model because it is mainly useful for feature extraction through its multi-layered architecture. The GNG, on the other hand, lacks the topological properties that are inherent to the cortical areas since the structure of the GNG is dependent on the distance of each prototype from the stimuli. Hence, the choice is between the SOM and the SNN with STDP. Both models show interesting properties like unsupervised learning and distributed computing, but at the cost of massive interconnections amongst neurons. In fact, even though the learning process of the SOM (Kohonen, 1982) and SNN (Diehl and Cook, 2015) can be distributed with local computing, the competition mechanism based on inhibition requires either a centralized unit or an all-to-all connectivity amongst neurons, as further discussed in chapter 3.

Following the work initiated by Laurent Rodriguez in 2015 (Rodriguez, 2015), we propose in section 2.5 to distribute the SOM computing based on the IG, a cellular neuromorphic architecture with local connectivity amongst neighbor neurons. Moreover, it has been shown that SOMs have a better performance in representing overlapping structures compared to classical clustering techniques such as partitive clustering or K-means (Budayan, Dikmen, and Birgonul, 2009). It partly explains why the SOM is one of the most popular ANNs in the unsupervised learning category (Kohonen, Schroeder, and Huang, 2001), used in a large range of applications going from high-dimensional data analysis (Kohonen et al., 1996) to more recent developments such as identification of social media trends (Silva et al., 2018), incremental change detection (Nallaperuma et al., 2018) and energy consumption minimization on sensor networks (Kromes et al., 2019). Therefore, we choose to explore the SOM and two of its major variants with the IG substrata, which we confront thereafter to the SNN in chapter 3 in terms of classification accuracy, learning dynamicity and hardware scalability.

2.4 SOM models

We present in this section three models of SOMs: the original Kohonen SOM (KSOM) (Kohonen, 1982), the Dynamic SOM (DSOM) (Rougier and Boniface, 2011) proposed by Rougier et al. in 2011 and the Pruning Cellular SOM (PCSOM) (Upegui et al.,

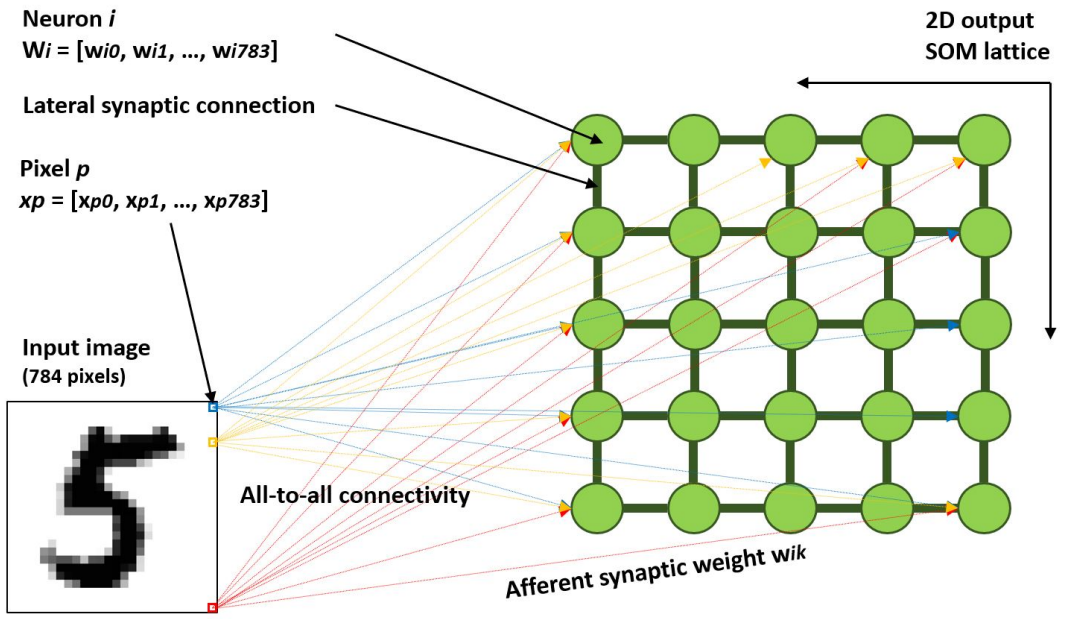

FIGURE 2.1: Self-Organizing Map (SOM) topology.

2018) proposed by Upegui et al. in 2018. Each SOM uses a two-dimensional grid of neurons, where each neuron has a respective two-dimensional position in the grid and is connected to the input stimulus through afferent synapses that carry the weights where the learning occurs. The weights of the neuron are represented as an m-dimensional vector where m is defined by the dimensions of the input stimuli. Each neuron is then connected to its four neighbors from north, east, south and west through lateral synapses without weights, as shown in figure 2.1. The mechanism by which the neuron communicates with its neighbors is detailed in section 2.5.

2.4.1 Kohonen SOM

The original KSOM algorithm introduced by Kohonen (Kohonen, 1982) is described in algorithm 1, where we use a two-dimensional array of k neurons.

It is to note in algorithm 1 that t_f is the number of epochs, i.e. the number of times the whole training dataset is presented. We introduced a new activity computation from the distance d in equation 2.2 so that this activity could later be used for the Hebbian learning in the multimodal association mechanism presented in chapter 5. The α hyper-parameter in equation 2.2 is the width of the Gaussian kernel. Its value is fixed to 1 in the SOM training, but it does not have any impact in the training phase since it does not change the neuron with the maximum activity. Its value becomes critical though in the labeling process presented in chapter 3.

The KSOM has a decaying learning rate ϵ which regulates the weights update and a decaying neighborhood width σ which defines a Gaussian neighborhood around the BMU where neurons learn, so that the learning stabilizes after a certain number of iterations. When $t = t_f$, the KSOM is almost unable to learn any change in the input stimuli, as $\epsilon_f << \epsilon_i$ and $\sigma_f << \sigma_i$. Therefore, the learning is stable but not dynamic. It can be considered as an off-line unsupervised learning algorithm.

Algorithm 1: Kohonen SOM algorithm

Initialize the network as a two-dimensional array of k neurons, where each neuron n with m inputs is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
for t from 0 to t_f **do**
  for every input vector v **do**
    for every neuron n in the network **do**
      Compute the afferent activity a_n from the distance d:

$$d = \|v - w_n\| \tag{2.1}$$

$$a_n = e^{-\frac{d}{\alpha}} \tag{2.2}$$

    end for
    Compute the winner s such that:

$$a_s = \max_{n=0}^{k-1} (a_n) \tag{2.3}$$

    for every neuron n in the network **do**
      Compute the neighborhood function $h_\sigma(t, n, s)$:

$$h_\sigma(t, n, s) = e^{-\frac{\|p_n - p_s\|^2}{2\sigma(t)^2}} \tag{2.4}$$

      Update the weight w_n of the neuron n:

$$w_n = w_n + \epsilon(t) \times h_\sigma(t, n, s) \times (v - w_n) \tag{2.5}$$

    end for
  end for
  Update the learning rate $\epsilon(t)$:

$$\epsilon(t) = \epsilon_i \left(\frac{\epsilon_f}{\epsilon_i}\right)^{t/t_f} \tag{2.6}$$

  Update the width of the neighborhood $\sigma(t)$:

$$\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{t/t_f} \tag{2.7}$$

end for

2.4.2 Dynamic SOM

The DSOM introduced by Rougier et al. (Rougier and Boniface, 2011) is a variation of the KSOM algorithm where the time dependency of the learning rate and neighborhood function has been replaced by the distance between the winning neuron and the input stimulus, as shown in algorithm 2.

It is to note in algorithm 2 that η is the elasticity or plasticity parameter. In the DSOM algorithm, if a neuron is close enough to the stimulus, then this neuron is

Algorithm 2: Dynamic SOM algorithm

Initialize the network as a two-dimensional array of neurons, where each neuron n is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
for every new input vector v **do**
  Compute the winner s such that w_s is the closest to v following equations 2.1, 2.2 and 2.3.
  for every neuron n in the network **do**
    Compute the neighborhood function $h_\sigma(t, n, s)$:
    if $v = w_s$ **then**

$$h_\eta(n, s, v) = 0 \tag{2.8}$$

    else

$$h_\eta(n, s, v) = e^{-\frac{1}{\eta^2}\frac{\|p_n - p_s\|^2}{\|v - w_s\|^2}} \tag{2.9}$$

    end if
    Update the weight w_n of the neuron n:

$$w_n = w_n + \epsilon \times \|v - w_n\| \times h_\eta(n, s, v) \times (v - w_n) \tag{2.10}$$

  end for
end for

already representing well the stimulus, hence there is no need for any neuron to learn (the extreme case where $v = w_s$ and thus $h_\eta(n, s, v) = 0$). In the other case, if there is no neuron close enough to the stimulus, all neurons learn according to their weight distance to this new stimulus and their topological distance to the BMU.

This mechanism allows a dynamic learning to adapt to any change in the environment at any moment. The DSOM can therefore be used for on-line unsupervised learning. However, in contrast with SOM, the DSOM self-organizes into the support of the distribution of the input stimuli and does not try to match the density (Rougier and Boniface, 2011).

2.4.3 Pruning Cellular SOM

The PCSOM introduced by Upegui et al. (Upegui et al., 2018) is also abstracted from the time dependency of the KSOM, it is hence made for continuous on-line unsupervised learning. In addition, it models a specific mechanism of biological neurons: the synaptic pruning. Indeed, each neuron of the PCSOM has a number of associated lateral synaptic connections varying from 0 to 4 that define its lateral influence during training.

These lateral synapses can be seen as interconnection matrices that are initially interconnecting every neuron to its four physical neighbors. Afterwards, during the network lifetime where the network learns to represent the input stimuli, some of these synapses will be pruned (removed) in order to allow the prototype vectors to better fit their density function. After a certain number of iterations, the PCSOM begins to create clusters, i.e. group of topologically close neurons which represent variations of the same class. The idea is then to prune the synaptic connections between the frontiers neurons of different topological clusters and isolate them, so

that one class does not affect the learning of the other. The PCSOM algorithm is described in algorithm 3.

Algorithm 3: Pruning Cellular SOM algorithm

1: **Initialize** the network as a two-dimensional array of neurons, where each neuron n is defined by a two-dimensional position, a randomly initialized m-dimensional weight w_n vector and a set of synapses defining connections to other neurons with respect to its position.
2: **for** every new input vector v **do**
3: **Compute** the winner s such that w_s is the closest to v following equations 2.1, 2.2 and 2.3.
4: **Update** the weight of the winner w_s:

$$w_s = w_s + \alpha \times (v - w_s) \times \|v - w_s\| \tag{2.11}$$

5: **for** every other neuron n in the network **do**
6: **Update** the weight w_n of the neuron n:

$$w_n = w_n + \alpha \times (w_i - w_n) \times e^{(-\frac{1}{\eta}\frac{hops}{\|w_i - w_n\|})} \tag{2.12}$$

7: **end for**
8: **for** every synapse in the network **do**
9: **Apply pruning** following the probability:

$$P_{ij} = e^{(-\frac{1}{\omega}\frac{1}{\|w_i - w_j\| t_i t_j})} \tag{2.13}$$

10: **end for**
11: **end for**

It is to note in algorithm 3 that α is the learning rate, w_n is the weight vector of the neuron to be updated, $hops$ is the number of propagation hops from the winner, w_i is the weight vector of the influential neuron, η is the elasticity of the network, P_{ij} is the probability of pruning the synapse interconnecting n_i and n_j, ω is the pruning rate, and t_i is the time from the last winning of neuron n_i. With respect to the neuron n to be updated, the influential neuron i is the connected neighbor neuron that is closest to the winner neuron s in terms of number of $hops$. In the case where two connected neighbor neurons have the same number of $hops$, the choice of the influential neuron is made randomly.

Thanks to the propagation of the neurons update through the neurons neighbors, the overall network weights are influenced by every new input vector depending on the network connectivity: if a neuron n is a topological neighbor of the influential neuron i, it is only updated if the synapse connecting n to i has not been pruned. The PCSOM is therefore modifying the connectivity of the neural network by pruning the useless synapses that connect two neurons whose activities are poorly correlated, as shown in figure 2.2 where $\omega = 3e - 07$. It allows the network to better fit the probability density function of the input stimuli. Preliminary results in (Upegui et al., 2018) have shown that the proposed pruning mechanism improves the network performance by reducing the Average Quantization Error (AQE) of the system for two-dimensional stimuli.

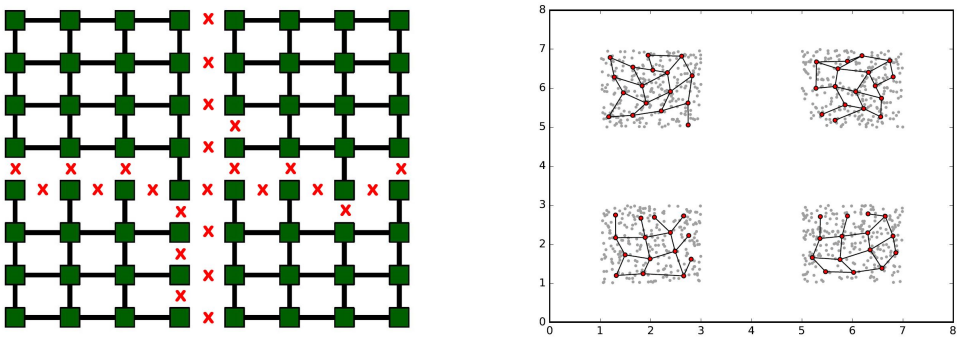

FIGURE 2.2: PCSOM after training: (left) cellular connections; (right) weights and inputs probability density.

The PCSOM is thus a dynamic and evolving (with respect to pruning) SOM that may reach a better performance than DSOM for on-line unsupervised learning thanks to the pruning mechanism. The comparative results on image classification for the three SOM models will be discussed in chapter 3.

2.5 Cellular neuromorphic architecture

As discussed in chapter 1, the centralized neural models that run on classical computers suffer from the Von-Neumann bottleneck due to the overload of communications between computing memory components, leading to a an over-consumption of time and energy. One attempt to overcome this limitation is to distribute the computing amongst neurons as done in (Kohonen, 1982) and (Diehl and Cook, 2015), but it implies an all-to-all connectivity to calculate the global information, i.e. the BMU and the topological distance of each neuron to it. Therefore, this solution does not completely solve the initial problem of scalability.

An alternative approach to solve the scalability problem can be derived from the Cellular Automata (CA) which was originally proposed by John von Neumann (Kemeny, 1967) and further formalized by Wolfram (Wolfram, 1984b). The CA paradigm relies on locally connected cells with local computing rules which define the new state of a cell depending on its own state and the states of its neighbors. All cells can then compute in parallel as no global information is needed. Therefore, the model is massively parallel and is an ideal candidate for hardware implementations (Halbach and Hoffmann, 2004).

A recent FPGA implementation to simulate CA in real time has been proposed in (Kyparissas and Dollas, 2019), where authors show a speed-up of 51$\times$ compared to a high-end CPU (Intel Core i7-7700HQ) and a comparable performance with recent GPUs with a gain of 10$\times$ in power consumption. With a low development cost, a low cost of migration to future devices and a good performance, FPGAs are suited to the design of cellular processors (Walsh and Dudek, 2012). Cellular architectures for ANNs were common in early neuromorphic implementations and have recently seen a resurgence (Schuman et al., 2017). Such implementation is also refered as near-memory computing where one embeds dedicated co-processors in close proximity to the memory unit, thus getting closer to the PDP paradigm (Blazewicz et al., 2000) formalized in the theory of ANNs. An FPGA distributed implementation model for SOMs was proposed in (Sousa and Del-Moral-Hernandez, 2017), where the local computation and the information exchange among neighboring neurons enable a global self-organization of the entire network.

Similarly, following the work initiated in (Rodriguez, 2015), we proposed a cellular formulation of the related neural models which would be able to tackle the full connectivity limitation by iterating the propagation of the information in the network (Rodriguez, Khacef, and Miramond, 2018). This particular cellular implementation, named the Iterative Grid (IG), reaches the same behavior as the centralized models but drastically reduces their computing complexity when deployed on hardware. Indeed, as detailed in chapter 3, the time complexity of the IG is $O(\sqrt{n})$ with respect to the number of neurons n in a squared map, while the time complexity of a centralized implementation is $O(n)$. In addition, the connectivity complexity of the IG is $O(n)$ with respect to the number of neurons n, while the connectivity complexity of a distributed implementation with all-to-all connectivity is $O(n^2)$ (Diehl and Cook, 2015). The principles of the IG are summarized in this section followed by a new SOM implementation over the IG substrata which takes in account the needs of the multimodal association learning and inference.

2.5.1 Iterative Grid (IG) substrata

Let's consider a two-dimensional grid shaped Network-on-Chip (NoC). This means that each node (neuron) of the network is physically connected (only) to its four closest neighbors. At each clock edge, each node reads the data provided by its neighbors, computes on these data and then propagates the result to its own neighbors on the next clock edge. The data is propagated (or broadcasted) in a certain amount of time to all the nodes. The maximum amount of time T_p which is needed to cover all the NoC (worst case reference) depends on its size: for a $N \times M$ grid, $T_p = N + M - 2$. After T_p clock edges, new data can be sent. A set of T_p iterations can be seen as a *wave of propagation.*

For the SOM afferent weights learning, the data to be propagated is the maximum activity for the BMU election, plus its distance with respect to every neuron in the map. The maximum activity is transmitted through the wave of propagation, and the distance to the BMU is computed in the same wave thanks to this finding: "When a data is iteratively propagated through a grid network, the propagation time is equivalent to the Manhattan distance between the source and each receiver".

The cellular propagation wave algorithm executed by each cell synchronously is detailed in Algorithm 4, where T_i is the iteration time that goes from 0 to T_p. This T_i is to distinguish from t in algorithm 1 which is relative to the training epoch. R the data stored in the node, D_j is the data given by the neighbor j with $j \in [\![0;3]\!]$ and the output buffers are memories used to keep the data consistency during the process. Each connection to neighbor nodes is provided with output double buffers, since we need to save the data of both current and previous clock edges.

A number of generic functions have been defined and explained in section 2.5.2. In summary, the IG substrata allows to implement a cellular architecture able to distribute the centralized behavior of SOMs into each node of the NoC, transforming the connectivity complexity into a scalable time complexity in $O(\sqrt{n})$ with respect to the number of neurons n regardless of the simulated SOM model.

2.5.2 IG for cellular distributed SOM

The SOM implementation on the IG has to take in consideration the needs of the multimodal association that will be presented in chapter 5: (1) we add the Worst Matching Unit (WMU) activity needed for the activities min-max normalization in the convergence step, and (2) we use the Gaussian kernel in equation 2.2 to transform

Algorithm 4: Cellular propagation wave algorithm

T_0: Let D_c the initial data of the cell.
Compute $R \leftarrow g_1(D_c)$
Write R on the output buffer.
for all T_i **do**
  for all D_j **do**
    Compute $R_j = f(D_j, T_i)$
  end for
  Compute $R = g_4(R, R_0, R_1, R_2, R_3)$
  Write R on the output buffer.
  Switch output buffers.
end for

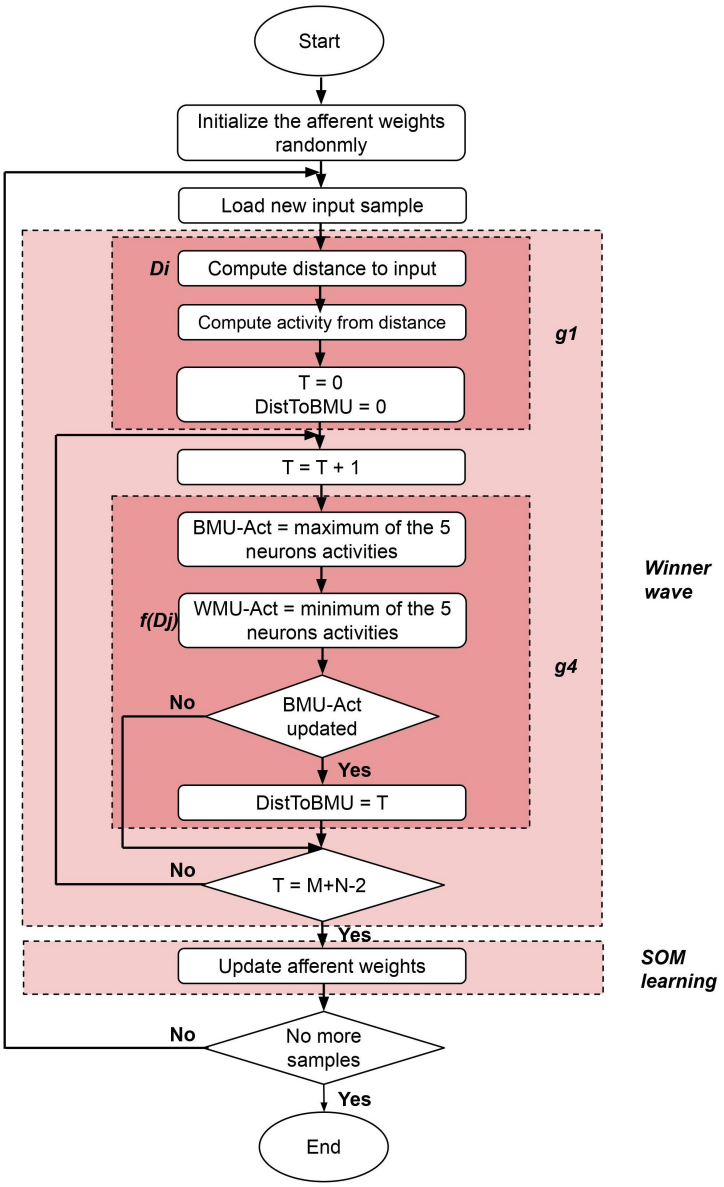

FIGURE 2.3: Flowchart: BMU and WMU distributed computing for each neuron.

the euclidean distances into activities. Therefore, the BMU is the neuron with the maximum activity and the WMU the neuron with the minimum one. The winner search wave and learning step are summarized as a flowchart in figure 2.3. This flowchart describes the KSOM learning, but the winner wave is applied the same way for all steps of the multimodal learning while the learning part can be replaced by Hebbian-like learning or inference.

BMU/WMU search wave

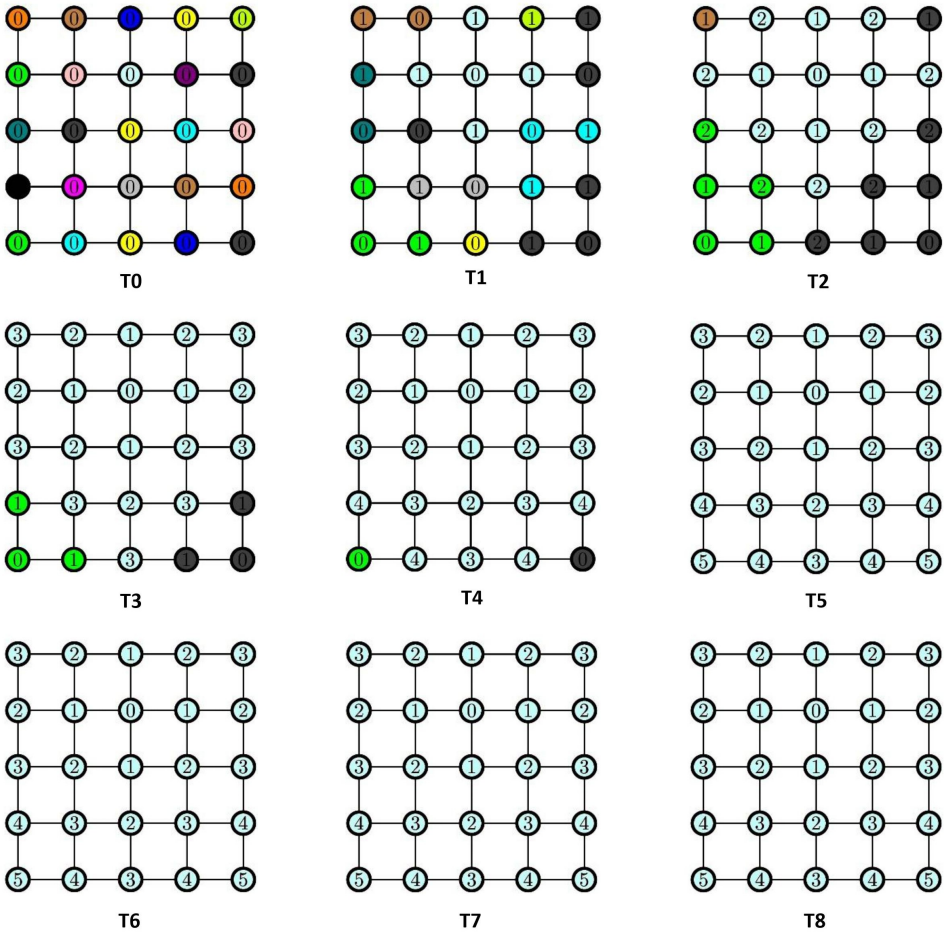

FIGURE 2.4: Iterative Grid BMU/WMU search wave in a 5×5 SOM.

In order to compute the BMU and WMU search, we have to define D_c, R, g_1, f, and g_4. Figure 2.3 shows the BMU/WMU search IG implementation with a flowchart representation. Here, D_c is the activity A computed by the neuron before the wave, as defined in equation 2.2. R contains A_{MIN} and (A_{MAX}, T_M) with A_{MIN} and A_{MAX} the current known WMU and BMU activities respectively which are detected by the neuron at the iteration of propagation T_M. g_1 initializes R with $A_{MIN} = A_{MAX} = D_c$ and $T_M = 0$. Because the propagation time T is equivalent to the Manhattan distance, T must be coherently coupled with A_{MAX} for the learning equation computation. i is the radius of the propagation which goes from 1 to half the perimeter of the grid. f is very similar to g_1 and sets R_j values to the respective value $[A_{MIN_j}, (A_{MAX_j}, T_i)]$. Finally, g_4 selects the minimum activity A_{MIN} and the maximum one A_{MAX} between R, R_0, R_1, R_2, R_3 and stores it as a result in the neuron's output buffer. After this propagation wave, each neuron n contains its own

$[A_{MIN}, (A_{MAX}, T_M)]$ with A_{MIN} and A_{MAX} common for each of them and T_M distinct values depending on their respective Manhattan distances to the BMU. Hence, equation 2.3 is implemented without using a central controller or a full connectivity, but with a simple iterative cellular method based on a local connectivity.

Figure 2.4 shows how the BMU information (color) and the topological distance to it (index) evolves with the cycles for each neuron of the SOM implemented with the IG. At T0, each of the 25 neurons compute their respective distance and activity to the input; then from T1 to T8, during (N+M-2 = 8 cycles), each neuron operates simultaneously and synchronously the operations defined in algorithm 4; and at T9, each neuron operates its afferent synaptic weights. Even though the propagation is complete at T5, its computation must continue until the pre-defined number of cycles corresponding to the worst case.

SOM learning

From the winner propagation wave, all useful data are present in each neuron to compute the learning equation. No propagation wave is necessary at this step. We notice that the A_{MIN} information is not necessary for the KSOM learning, but it is needed for the upcoming multimodal convergence step.

Behavioral study

The iterative grid is based on the time to distance transformation. This implies the use of a Manhattan distance in the models despite of the Euclidean one often used in software or centralized hardware implementations. Otherwise, if the Manhattan distance decreases the performance of the SOM, it is also possible to include the two-dimensional position of the BMU with its activity so that each neuron can calculate its Euclidean distance to the BMU. It would, however, increase the size of the data to be shared amongst neurons.

In order to prove the same behavior between the centralized and the distributed implementations of the SOM, we run three scenarii with three different distributions of two-dimensional data, keeping the same random seed on both architectures. Afterwards, we compare the AQE and the afferent weights between the same neurons after the same number of training iterations. When we use the Manhattan distance for the centralized version of the KSOM, the two implementations lead to exactly the same results for the AQE and a zero distance for the afferent weights. This illustrates that the model behaves the same way with or without using the IG formalism and substrata. The same result is obtained when the IG is used with the DSOM. More generally, the IG substrate can distribute any SOM-like model without extra design efforts. If we need to use the Euclidean distance as originally defined by Kohonen (Kohonen, 1982), then we only need to transmit the topological position of the BMU along its activity to the four neighbor neurons. There is then a compromise between the classification accuracy and the hardware efficiency of the system.

2.5.3 FPGA hardware support

The multi-FPGA implementation of the IG is a work in progress based on a previously implemented Neural Processing Unit (NPU) (Rodriguez, Fiack, and Miramond, 2013) (Fiack, Rodriguez, and Miramond, 2015). As shown in figure 2.5, the

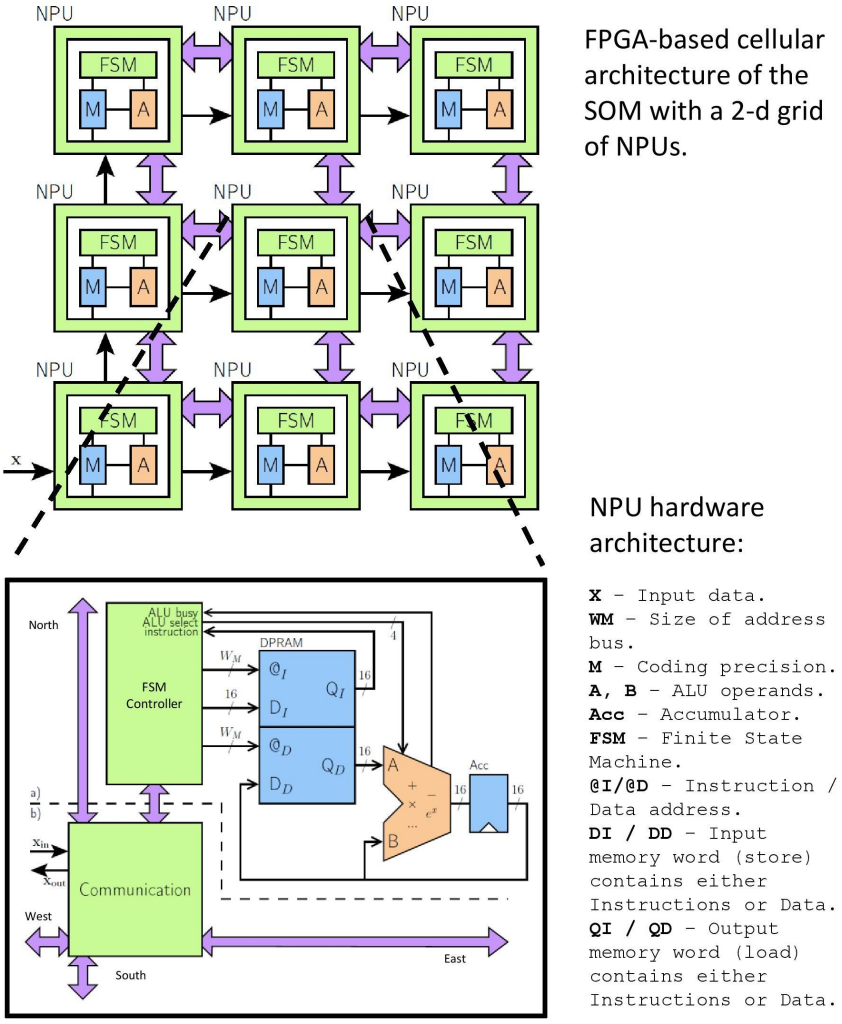

FIGURE 2.5: Neural Processing Units (NPUs) grid on FPGA.

NPU is made of two main parts: the computation core and the communication engine. The computation core is a lightweight Harvard-like accumulator-based microprocessor where a central dual-port RAM memory stores the instructions and the data, both separately accessible from its two ports. A Finite State Machine (FSM) controls the two independent ports of the memory and the Arithmetic and Logic Unit (ALU), which implements the needed operations to perform the operations of algorithm 4. The aim of the communication engine is to bring the input stimuli vector and the neighbors activities to the computation core at each iteration. The values of the input vector flow across the NPUs through their x_{in} and x_{out} ports which are connected as a broadcast tree. The output activity ports of each NPU are connected to the four cardinal neighbors through a dedicated hard-wired channel. Originally implemented on an Altera Stratix V GXEA7 FPGA, the resources (LUT, Registers, DSP and memory blocks) consumption is indeed scalable as it increases linearly in function of the size of the NPU network (Rodriguez, Fiack, and Miramond, 2013) (Fiack, Rodriguez, and Miramond, 2015). Future works will focus on configuring the new model in the NPU and implementing it on a more recent and adapted FPGA device, particularly for the communication part between multiple FPGA boards that will be based on (Vannel et al., 2018). It will be further discussed in chapter 7.

In terms of scalable FPGA designs for neural networks, we find in the literature

the work of Moore et al. (Moore et al., 2012) with the Bluehive project, a custom 64-FPGA machine made for large-scale real-time neural network simulation with a reconfigurable communication topology. Moore et al. showed that FPGAs perform much better than current CPUs/GPUs for cellular architectures due to the low-latency and high-bandwidth communication needs. More recently, Wang et al. proposed an advanced multipurpose neuromorphic engine that breaks the *Liebig's law*, i.e. the problem that the performance of the system is limited by the component in shortest supply. The authors implemented an array of identical components, each of which can be configured as a Leaky Integrate-and-Fire (LIF) neuron, a learning-synapse or an axon with trainable delay. Wang et al. also proposed an FPGA implementation of parallel and scalable neuromorphic cortex simulator (Wang, Thakur, and Schaik, 2018), arranged in minicolumns and hypercolumns. Similarly to (Fiack, Rodriguez, and Miramond, 2015), the cortex simulator can be reconfigured for simulating different neural networks without any change in hardware structure by programming the memory. However, works in (Moore et al., 2012), (Wang and Schaik, 2018) and (Wang, Thakur, and Schaik, 2018) target SNNs with no on-chip learning for (Moore et al., 2012) and (Wang, Thakur, and Schaik, 2018). Our goal in terms of multi-FPGA communication is similar to (Moore et al., 2012), while neurons interconnections are local and thus different from (Wang and Schaik, 2018) (central controller) and (Wang, Thakur, and Schaik, 2018) (hierarchical communication).

2.5.4 Comparison to state of the art approaches

Finally, the only cellular approach for implementing SOM models is proposed by Sousa et al. (Sousa and Del-Moral-Hernandez, 2017). It is an FPGA implementation that shares the same approach as the IG with distributed cellular computing and local connectivity. However, the IG has two main advantages over the proposed cellular model in (Sousa and Del-Moral-Hernandez, 2017):

- Waves complexity: The ”smallest of 5” and ”neighborhood” waves in (Sousa and Del-Moral-Hernandez, 2017) have been coupled into one wave called the ”winner wave”, as the iterative grid is based on time to distance transformation to find the Manhattan distance between the BMU and each neuron. We have therefore a gain of about $2\times$ in the time complexity of the SOM training.
- Sequential vs. combinatory architecture: The processes of calculating the neuron distances to the input vector, searching for the BMU and updating the weight vectors are performed in a single clock cycle. This assumption goes against the iterative computing paradigm in the SOM grid to propagate the neurons information. Hence, the hardware implementation in (Sousa and Del-Moral-Hernandez, 2017) is almost fully combinatory. It explains why the maximum operating frequency is low and decreases when increasing the number of neurons, thus being not scalable in terms of both hardware resources and latency.

2.6 Conclusion

In this chapter, we have defined *our* foundations for brain-inspired computing with the biologically plausible properties that we aim to model in our proposed self-organizing neural system, namely the multimodal unsupervised learning at the behavioral level, the cellular computing at the algorithmic level and the neuromorphic

implementation at the hardware level. Next, we have made a historical review for the most successful computational models of neural self-organization, then we have made the choice of the SOM as a main component for unimodal processing in our multimodal framework. Finally, we introduced the IG, a cellular neuromorphic architecture used to distribute the SOM computation with local connectivity. The idea is to propagate the neurons information through the neurons grid in a certain number of iterations until the global information, i.e. the BMU and the distance of each neuron to it, emerges from the local interactions of connected neurons. We showed the generalization of the mechanism to any SOM-like model, the IG can therefore be used as a computing substrata to distribute neural models that use competition-based learning with excitation/inhibition mechanisms. We further discuss in chapter 3 its impact on scalability in terms of time complexity and connectivity complexity for hardware implementation.

Chapter 3

Confronting SOMs to SNNs for unsupervised learning

> We can't afford that all of our research is devoted to the machine: what we are trying to learn about isn't the machine that we are building, it's the brain.
>
> *Misha Mahowald.*

3.1 Introduction

During the last years, Deep Neural Networks (DNNs) have reached the highest performance in classification tasks. However, as previously discussed, such a success is mostly based on supervised and off-line learning: they require huge labeled datasets for learning, and once it is done, they cannot adapt to any change in the data from the environment. Consequently, with the increasing amount of unlabeled data gathered everyday through Internet of Things (IoT) devices and the difficult task of labeling each sample, DNNs are slowly reaching the limits of supervised learning (Droniou, Ivaldi, and Sigaud, 2015) (Chum et al., 2019). Hence, unsupervised learning is becoming one of the most important and challenging topics in ML and AI. In the context of brain-inspired computing, we apply the KSOM (Kohonen, 1982) for unsupervised learning without labels, and we explore two of its major extensions: the DSOM (Rougier and Boniface, 2011) that enables continuous learning and the PC-SOM (Upegui et al., 2018) that adds lateral synaptic pruning. First, we present the spiking neuron and the STDP learning rule of the SNNs that will be our reference for the comparative study. Afterwards, we introduce the *post-labeled unsupervised learning* problem and propose an automatic labeling method with three different variants to assign the class of each neuron, trying to reach the best accuracy while minimizing the number of labeled images we need. Finally, we confront the performance of the Kohonen-based SOMs with STDP-based SNNs in terms of classification accuracy, learning dynamicity and hardware scalability.

3.2 Spiking Neural Networks (SNNs)

3.2.1 Spiking neurons

Figure 3.1 ("Neurons and glial cells") represents a simplified anatomy of the biological neuron. It contains long and short extensions called axon and dendrites, respectively. Dendrites carry electric potentials towards the cell (inputs) and the axon carry

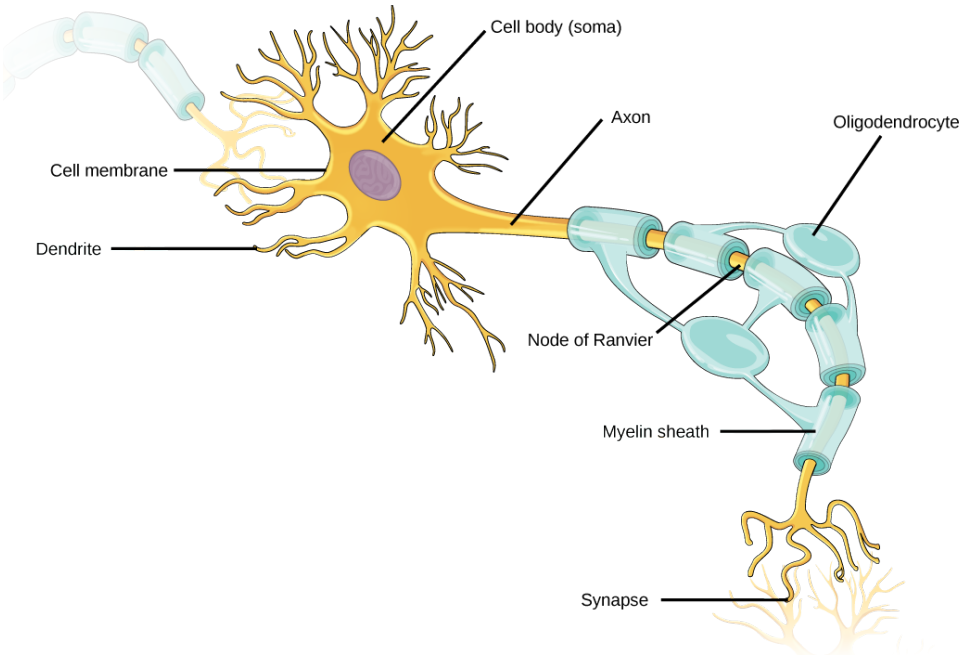

FIGURE 3.1: Anatomy of a biological neuron.

them away from the cell (output). The dendrite of one cell is connected to the axon of another, with a small gap in between called the synaptic gap or the synapse. In order to transmit information between cell, the cell transmits electrical signals called "spikes" that travel down the axon and causes the release of neurotransmitters that travel through the synapse to the other cell (Ranganathan and Kira, 2003). Although numerous models have been proposed by drawing inspiration from neuroscience like the biologically plausible complex Hodgkin–Huxley model (Hodgkin and Huxley, 1952) and the Izhikevich model (Izhikevich, 2003), most SNNs rely on the simple Integrate-and-Fire (IF) or Leaky IF (LIF) neuron models that provide reduced complexity, especially for hardware implementation, while producing the required key dynamics for computation. Equations 3.1 and 3.2 describe the discrete IF neuron computation in a multi-layer SNN topology.

$$s_j^l(t) = p_j^l(t-1) + \sum_{i=0}^{N_{l-1}-1} w_{ij}^l \times \gamma_i^{l-1}(t) \tag{3.1}$$

$$p_j^l(t) = \begin{cases} s_j^l(t) & \text{if } s_j^l(t) \leq \theta \\ 0 & \text{otherwise} \end{cases} \qquad \gamma_j^l(t) = \begin{cases} 0 & \text{if } s_j^l(t) \leq \theta \\ 1 & \text{otherwise} \end{cases} \tag{3.2}$$

It is to note in equations 3.1 and 3.2 that N_{l-1} is the number of neurons in the layer $l-1$, w_{ij} is the synaptic weight between the neuron i the in layer $l-1$ and the neuron j in the layer l, p_j^l is the potential of the neuron j in the layer l, θ is the threshold and γ_i^{l-1} is the spike state (0 or 1) of the neuron i in the layer $l-1$. Therefore, in an event-based hardware implementation, the multiplication in equation 3.1 can be represented as a simple chip enable that triggers the hardware neuron whenever $\gamma_i^{l-1} = 1$.

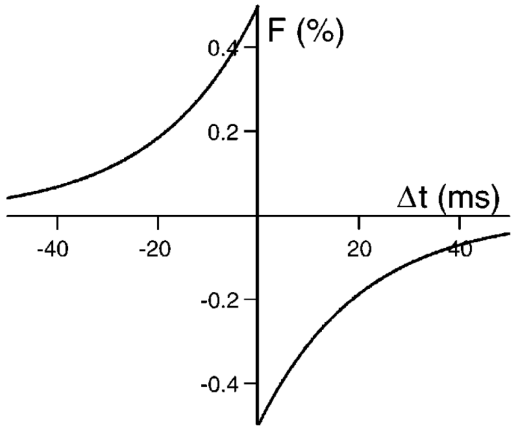

FIGURE 3.2: STDP modification function.

3.2.2 Spike Timing Dependant Plasticity (STDP)

As previously discussed, connections among neurons are plastic, i.e. locally strengthened and weakened so that a global learning could emerge. This synaptic plasticity in spiking neurons is modeled in the STDP, a brain-inspired unsupervised learning algorithm where the idea is to detect the causality between the neurons for each input: if a neuron spikes soon (before the expiry of a time Δt) after receiving a spike from a given synapse, it suggests that synapse played an important role in the triggering of the neuron, and therefore it reinforces that synapse by increasing its corresponding weight (LTP). In the other case, if a neuron spikes just before or a long time after receiving a spike from a given synapse, it suggests that synapse has no impact in the spike of the neuron, and therefore it decreases its corresponding weight (LTD). Song et al. (Song, Miller, and Abbott, 2000) explain that "the largest changes in synaptic efficacy occur when the time difference between pre- and postsynaptic action potentials is small, and there is a sharp transition from strengthening to weakening as this time difference passes through zero". This is illustrated in figure 3.2 (Song, Miller, and Abbott, 2000), where Δt is the time of the presynaptic spike minus the time of the postsynaptic spike. The change of the peak conductance at a synapse due to a single pre- and postsynaptic action potential pair is $F(\Delta t) \times C_{max}$, where C_{max} is the maximum conductance. The original STDP learning rule proposed by Song et al. (Song, Miller, and Abbott, 2000) is expressed in equation 3.3.

$$F(\Delta t) = \begin{cases} A^+ e^{(\Delta t/\tau^+)} & if \Delta t < 0 \\ -A^- e^{(-\Delta t/\tau^-)} & if \Delta t > 0 \end{cases} \tag{3.3}$$

It is to note in equation 3.3 that τ^+ and τ^- determine the ranges of pre- to postsynaptic interspike intervals over which synaptic strengthening and weakening occur, while A^+ and A^- determine the maximum amounts of synaptic modification which occur when Δt is close to zero. The STDP formalized in equation 3.3 provides a reasonable approximation of the dependence of synaptic modification on spike timing seen in the experimental data (Song, Miller, and Abbott, 2000), but other models of STDP with different levels of biological plausibility and computational complexity have been proposed and used, like in (Querlioz et al., 2013), (Diehl and Cook, 2015), (Kheradpisheh et al., 2018) and (Vigneron and Martinet, 2020).

3.2.3 SNN models

Baseline SNN

In 2015, Diehl and Cook (Diehl and Cook, 2015) proposed a SNN structure implementing STDP for unsupervised learning with lateral inhibition that generates competition amongst neurons, so that each neuron learns a different pattern. Indeed, as discussed in chapter 2, STDP requires both an activity-dependent synaptic plasticity and a competition mechanism (Song, Miller, and Abbott, 2000). This competition is implemented through a WTA inhibition mechanism. The SNN architecture is made of input, excitatory and inhibitory layers. Each excitatory neuron receives an SDTP-modifiable synapse from the input layer, and is connected to one neuron of the inhibitory layer which connects back to all the other excitatory neurons. Consequently, the first neuron that spikes inhibits all others. In order to have a first assessment and confrontation of the SOM and SNN classification accuracies, we choose the smallest topology that could be compared in the literature, i.e. 10×10 map of excitatory neurons. It implies a 10×10 map of inhibitory neurons as well, but we do not count them as *effective* neurons since they only represent an all-to-all connectivity amongst excitatory neurons for the inhibition mechanism. The baseline SNN accuracy on MNIST classification using 100 neurons is 80.71% (Hazan et al., 2018).

Lattice Map SNN (LMSNN)

In 2018, Hazan et al. proposed the Lattice Map SNN (LMSNN) (Hazan et al., 2018), an extension of (Diehl and Cook, 2015) that combines STDP with a inherent characteristic of the SOM: the topological neighborhood. To do so, the degree of competition imposed by the connections from the inhibitory layer is curved by increasing the level of inhibition with the distance between neurons. This form of competition, called soft competition, has two main advantages over hard competitive methods: first, in a hard competitive learning system, it is possible to have some "dead units", in this case neurons that are never winners for any input signal and, therefore, keep their initial random weights and remain as unused network resources (Ranganathan and Kira, 2003). Second, different random initializations may lead to widely differing results, because the purely local adaptations may not be able to get the system out of the local minimum where it started (Ranganathan and Kira, 2003). Hence, the soft WTA mechanism is preferred to the hard WTA for most purposes, including classification tasks. Indeed, the LMSNN achieved an accuracy of $85.71\% \pm 0.85\%$ on MNIST classification with 100 neurons, which improves the performance of the baseline SNN with hard WTA (Hazan et al., 2018).

Nevertheless, the works of (Diehl and Cook, 2015) and (Hazan et al., 2018) are limited in their labeling approach: since the training is unsupervised, we need a labeled subset of the training dataset in order to label the neurons for evaluation purposes and inference. Still, we must not use the whole training dataset for labeling. In the literature, the labeling is performed using one presentation of the whole training dataset (Diehl and Cook, 2015) (Hazan et al., 2018). It means that the DNNs initial limit of using labels is only shifted from the training part to the labeling part. In our work, we evaluate the minimal subset of labeled samples necessary to reach the best accuracy, as described in section 3.3.

3.3 SOM labeling and test

3.3.1 Post-labeled unsupervised learning problem

With the fast expansion of IoT devices, a huge amount of unlabeled data is gathered everyday. While it is a big opportunity for AI and ML, the difficult task of labeling DL techniques slowly reaching the limits of supervised learning (Droniou, Ivaldi, and Sigaud, 2015; Chum et al., 2019). Hence, unsupervised learning is becoming one of the most important and challenging topics in ML. In this thesis, we introduce the problem of post-labeled unsupervised learning: no label is available during training and representations are learned in an unsupervised fashion, then very few labels are available for assigning each representation the class it represents. The latter is called the labeling phase. The labeling phase is to distinguish from the fine-tuning process in semi-supervised learning where a labeled subset is used to re-adjust the synaptic weights and improve the neurons representations. Here, the synaptic learning is fully unsupervised, and labels are only used to name the class that each neuron represents. In certain cases such as classification tasks on the edge where the representations can be qualitatively named (e.g. written digits), this labeling phase can be replaced by an expert that should nevertheless not be prompted often. We consider the general case where the labeling phase is based on a labeled subset, then we try to minimize its size while maximizing the classification accuracy on the MNIST dataset (LeCun and Cortes, 1998).

3.3.2 Proposed labeling and test methods

At the end of the training process, each neuron of the SOM corresponds to a cluster prototype in the considered problem. At this stage, these prototypes are anonymous and cannot be directly used to perform classifications. The next step explains the neurons labeling process for transforming the SOM into a classifier. The labeling is the step between training and test (or inference) where we assign each neuron the class it represents in the training dataset. In our case with MNIST, each neuron has to be assigned a digit label from 0 to 9.

We propose in this section a labeling algorithm based on very few labels. The idea is the following: we consider a randomly chosen labeled subset of the training dataset, and we try to minimize its size while keeping the best classification accuracy. The proposed labeling is illustrated as a flowchart in figure 3.3. It can be summarized in five steps:

- First, we calculate the neurons activations based on the labeled input samples from the euclidean distance following Equation 2.2, where v is the input vector, w_n and a_n are respectively the weights vector and the activity of the neuron n. The parameter α is the width of the Gaussian kernel that becomes a hyper-parameter for the method.
- Second, the Best Matching Unit (BMU), i.e. the neuron with the maximum activity is elected.
- Third, each neuron accumulates its normalized activation (simple division) with respect to the BMU activity in the corresponding class accumulator, and the three steps are repeated for every sample of the labeling subset.
- Fourth, each class accumulator is normalized over the number of samples per class.

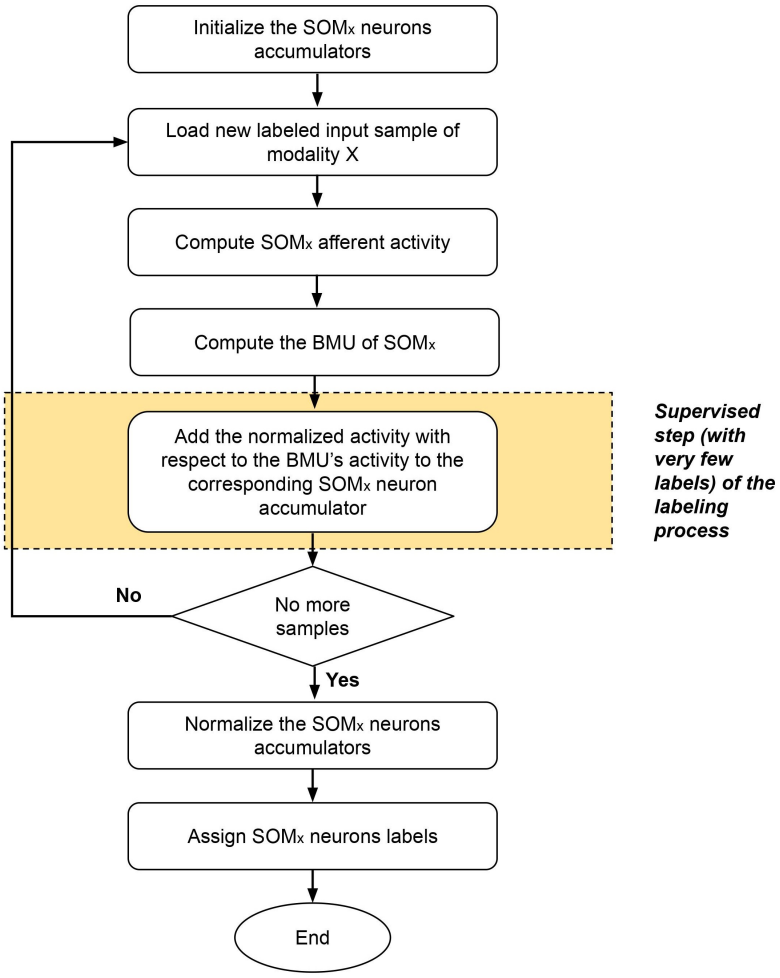

FIGURE 3.3: Flowchart: SOM labeling.

- Fifth and finally, the label of each neuron is chosen according to the class accumulator that has the maximum activity.

One could think about a simpler process where we simply count the number of times each neuron has been BMU for each class, and assign its label accordingly. The problem with this method is that some neurons are never elected as a BMU within the labeling subset, especially when this subset is very small and when the number of neurons grow. Our proposed labeling process prevents from this issue and guarantees that all neurons are labeled at the end of the labeling phase regardless of the labeling subset or SOM sizes.

The generic labeling process is detailed in algorithm 5. We tried three different methods that define the functions we use in the generic algorithm: *Activation*, *Distance* and *Gaussian*. For each of the three methods, we need a labeled subset as we need to know the class of each sample of it. Hence, we try to find the labeling method that requires the minimum number of labeled samples without reducing the best accuracy.

The associated functions of the three methods are detailed in table. 3.1, and their impact on accuracy with different labeling subset sizes is explained in section 3.3.3.

The parameter σ that we use in the function *dist_method* of the *Gaussian* method is the width of the Gaussian function that weighs the relevance of each neuron to

Algorithm 5: SOM labeling algorithm

Initialize label_count = zeros[number_of_classes];
accumulator = zeros[number_of_neurons, number_of_classes].
for every input sample in the labeling subset **do**
  for every neuron in the SOM network **do**
    dist_matrix[neuron.index] = ***dist_method***(neuron.weights, image.pixels)
  end for
  best_dist = ***best_dist_method***(dist_matrix)
  label_count[image.label] += 1
  accumulator[neurons, image.label] += ***norm_acc_method***(dist_matrix, best_dist)
end for
for i in range(number_of_classes) **do**
  accumulator[neurons, i] /= label_count(i)
end for
for every neuron in the SOM network **do**
  neuron_label[neuron.index] = ***find_best***(accumulator[neuron.index])
end for

TABLE 3.1: SOM labeling and test methods.

Function	**Method**		
	Activation	***Distance***	***Gaussian***
dist_method(x, y)	$\sum_{i=0}^{max}(x[i] \times y[i])$	$\sqrt{\sum_{i=0}^{max}(x[i] - y[i])^2}$	$e^{\frac{-\sqrt{\sum_{i=0}^{max}(x[i]-y[i])^2}}{\sigma}}$
best_dist_method(x)	max(x)	min(x)	max(x)
norm_acc_method(x, y)	x / y	x - y	x / y
find_best(x)	argmax(x)	argmin(x)	argmax(x)

the input sample with respect to its weights distance to the sample. We propose a method to approximate σ in algorithm 6.

Algorithm 6: σ computation algorithm for gaussian labeling method

Initialize dist_to_origin = zeros[number_of_neurons].
for every neuron in the SOM network **do**
  dist_to_origin[neuron.index] = $\sqrt{\sum_{i=0}^{max}(neuron.weights[i])^2}$
end for
σ = ***standard_deviation***(dist_to_origin)

After training and labeling, the remaining step is the test for accuracy which is detailed in algorithm 7. Here again, we try three test methods for inference, each of them corresponding to one of the three labeling methods whose functions are detailed in table 3.1.

3.3.3 Labeling methods: comparative study

After training the KSOM in 10 epochs on MNIST, we performed the labeling on the same trained network using three methods as shown in figure 3.4. The results are averaged over 10 different subsets of the same size each time.

Algorithm 7: SOM test algorithm

```
Initialize accuracy_count = 0; confusion_matrix = zeros[number_of_classes, number_of_classes]
for every input sample in the test subset do
    for every neuron in the SOM network do
        dist_matrix[neuron.index] = dist_method(neuron.weights, image.pixels)
    end for
    best_neuron = find_best(dist_matrix)
    if best_neuron.label = image.label then
        accuracy_count += 1
    end if
    Update confusion_matrix
end for
accuracy = (accuracy_count ÷ test_subset_size) × 100%
```

On the one hand, the *Activation* method does not perform well because of the confusion it creates between two different samples during labeling: for example, the activity (multiplication and activation) between the prototype neuron 1 and the two digits 1 and 7 is nearly the same, because the horizontal bar of 7 that makes the difference is multiplied by zero. Hence, the *Activation* method can hardly recognize which of the digits 1 and 7 the prototype neuron 1 represents. This confusion may also happen for other digits where one of them is "part of the other", like the digits 6 and 8. In contrary, the *Distance* method does not cancel this difference. On the other hand, the *Gaussian* method leads to a better accuracy than the *Distance* method, because of the distance modulation mechanism that gives more importance to the neurons whose weights are closer to the input sample during labeling. Moreover, the *Gaussian* method reaches the best accuracy with only 1% of labeled data, and the performance does not fluctuate a lot with different labeling subsets (± 0.23).

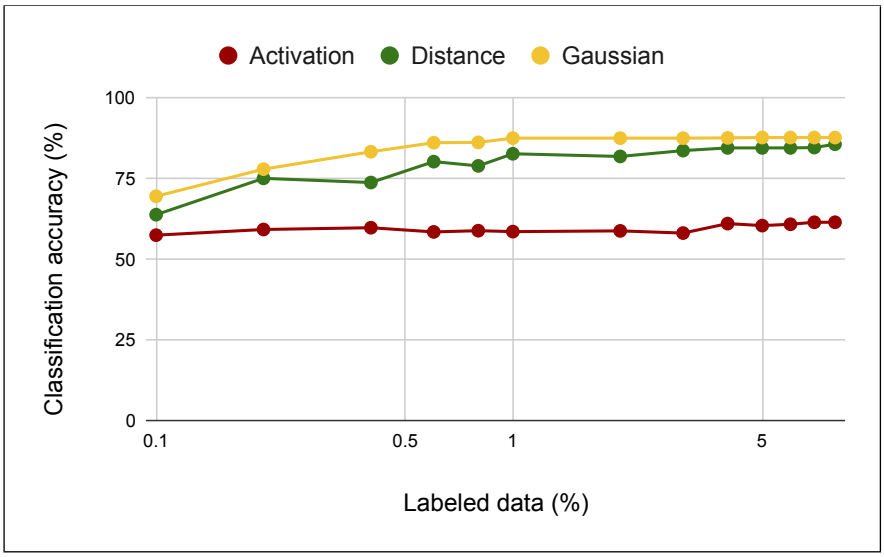

FIGURE 3.4: SOM labeling methods comparison.

In order to qualitatively asses our best labeling method with the minimal labeled subset size, i.e. the *Gaussian* method with 1% of labeled data, we displayed the KSOM trained weights in figure 3.5-a and their corresponding labels in figure 3.5-b.

We see that we would manually do the same labeling, except for one neuron which didn't converge well (neuron in the bottom right corner of the KSOM grid, labeled as 4). We could only hesitate between some labels whose representations are very close, like 4 and 9, and this is an interesting point that we discuss further in section 3.4. Hence, we tried to manually label the KSOM, modifying some labels of the *Gaussian* method: the accuracy decreased by approximately 1%. It shows that the *Gaussian* method performs better than a qualitative manual labeling.

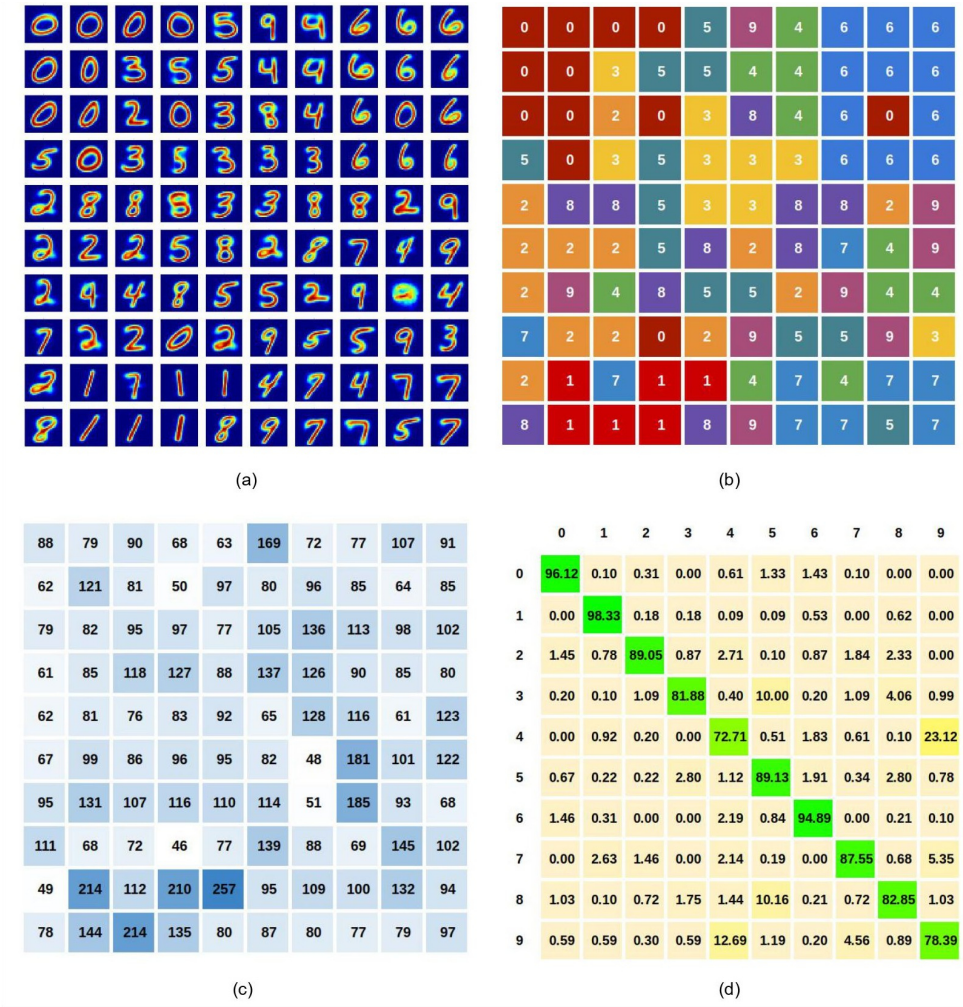

FIGURE 3.5: KSOM trained on MNIST: (a) neurons synaptic weights; (b) neurons labels; (c) neurons BMU counters; (d) confusion matrix.

3.4 MNIST unsupervised classification performance

3.4.1 Confronting KSOM, DSOM and PCSOM

In order to compare the accuracy performance of our three SOMs with each other and with respect to the state of the art, the first step is to perform the training of each of the KSOM, DSOM and PCSOM over the MNIST training dataset for the same number of iterations, and then to perform the labeling and the test. Even though the DSOM and PCSOM enable on-line learning, we stop it during labeling and test to assess the performance of our three SOMs with the same number of images for training. Once again, we do not use any label for training, even if they are actually

available in the MNIST training dataset, as the goal of our work is to generalize our method for unlabeled datasets and data from the real-world environment where there is no label.

A important step was to determine the number of iterations that we needed for learning in order to achieve the best accuracy. In other words, we had to set a convergence criterion that would guide us to the number of iterations we need, and we have chosen the Average Quantization Error (AQE) as done in (Upegui et al., 2018). It is calculated after each iteration on the same randomly chosen subset of 10% of the training dataset following equation 3.4.

$$AQE = \frac{1}{K}\sum_{i=1}^{K}\min_{1\leq n\leq N}\left(\sqrt{\sum_{p=0}^{max}(image_i[p] - neuron_n[p])^2}\right)^2 \tag{3.4}$$

It is to note in equation 3.4 that K is the number of input vectors used as reference for computing the AQE (we use 10% of the training dataset, so $K = 6000$), N is the total number of neurons and p is the index of the pixel in the image and the corresponding synaptic weight in the neuron. We notice that we only consider the minimum euclidean distance between the image and the neurons, i.e. the distance between the image and the BMU.

TABLE 3.2: SOMs training hyper-parameters.

KSOM				**DSOM**		**PCSOM**		
ϵ_i	ϵ_f	σ_i	σ_f	ϵ	η	α	η	ω
1	0.01	5	0.01	0.005	0.1	0.01	0.1	0

We trained the three SOMs over 10 epochs on the 60,000 training images of MNIST database. The hyper-parameters were found with a grid search and are reported in table 3.2. To assess the learning convergence, we calculated the AQE after each iteration on the same randomly chosen subset of 10% of the training dataset. The results in figure 3.6, averaged over 10 runs of training, show that the three SOMs learnings converge with different speeds: the KSOM converges after approximately 7 iterations, while the DSOM and PCSOM converge after only 3 iterations, with the two plots almost superposed. This is due to the absence of temporal dependency in both DSOM and PCSOM. Nevertheless, the KSOM converges to a slightly better AQE value that is reflected by a better accuracy, as shown in figure 3.7. The AQE, though, is not directly proportional to the accuracy, as two different representations may have the same AQE but different test (generalization) accuracies.

We computed the test accuracy after each training epoch, as shown in figure 3.7, averaged on 10 runs of training. We see that the DSOM and PCSOM learn faster than the KSOM, but the KSOM reaches the highest accuracy of 87.36% $\pm$ 00.23%. Like for STDP-based learning (Diehl and Cook, 2015), one good property of Kohonen-based learning is the absence of over-fitting even when performed for 100 epochs, i.e 6 million samples (accuracy of approximately 87%). Indeed, unlike many ANNs which tend to over-fit the data (Diehl and Cook, 2015), the SOMs learning is stable over time. We calculated the average and standard deviation of the test accuracy for each of the three models over 10 runs of training (10 epochs each, with random initialization and training dataset shuffle after each epoch). The results are summarized in table 3.3.

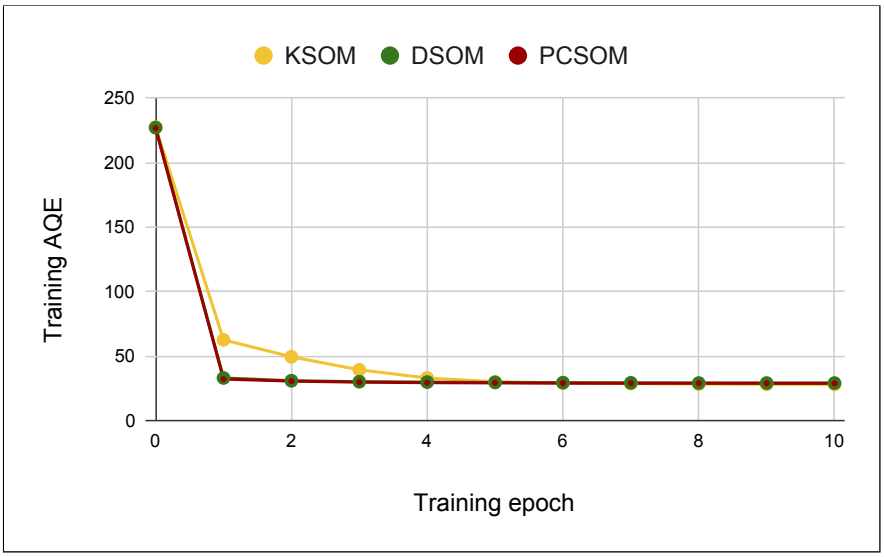

FIGURE 3.6: SOMs training AQE on MNIST.

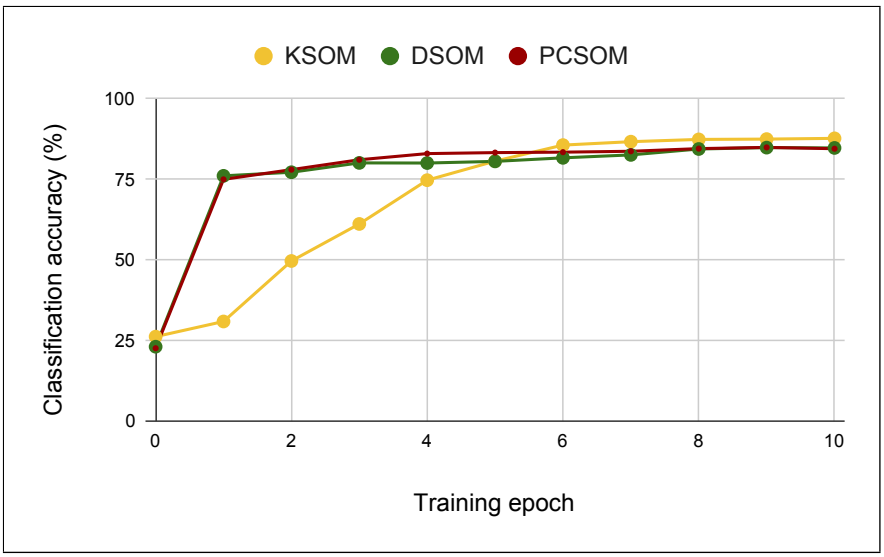

FIGURE 3.7: SOMs classification accuracy on MNIST.

Figure 3.5-c shows the number of times each neuron has been BMU during test. Since the number of digits per class in the MNIST test dataset is nearly equal, the neurons which have been BMUs the most are the neurons that are the fewest to represent a class, like the digit class 1. This is an interesting behavior which shows that the more diverse the class is, the more neurons it needs to represent it. The digit 1, in contrary, does not have a high diversity, it can thus be represented by fewer neurons which are then BMUs more often. To understand the KSOM mistakes, we plot the confusion matrix in figure 3.5-d where the left column represents the image class and the top row represents the BMU label. We notice that the biggest values occur in the diagonal, with some peak values between the digits whose representations are close: 16.75% of the digits 9 are classified as a 4, and 14.46% of the digits 4 are classified as 9. We find the same mistakes with a lower percentage between the digits 3, 5 and 8, because of their proximity in the 784-dimensional vector space. Our hypothesis is that the multimodal association can overcome these mistakes by adding a different but complementary modality like sound, and exploiting both modalities to improve

the overall accuracy. This will be explored in detail in chapter 5.

3.4.2 Confronting SOMs to SNNs

Both SOMs and SNNs are brain-inspired neural models that were intitially proposed to model the cortical plasticity as presented in chapter 2. They both learn a compressed representation of the data in an unsupervised fashion by using a two-dimensional neural map topology. The neurons of SOMs and SNNs have excitatory afferent connections and a lateral competition mechanism necessary for the learning, which is based on two phases: the global election of the winning neuron (called a BMU in the SOM and a WTA in the SNN) and the local synaptic weights update. SOMs and SNNs only differ in the way they encode information. SNNs use spike coding in a certain temporal window while SOMs use activities that can be considered as an average spiking rate. In spite of the different information coding schemes, the SOM and SNN neurons prototypes are very similar and the learning converges in a similar way. The work of Hazan et al. (Hazan et al., 2018) where the authors merged the baseline SNN model with the topolgical neighborhood inherent to the SOM to improve the SNN accuracy is another evidence of the strong link between the two models. This is further discussed in the comparative study of this section, where we first compare the SOM models then confront them to SNNs.

In terms of accuracy, the KSOM outperforms the DSOM and PCSOM. However, the DSOM and PCSOM enable on-line learning and can continuously learn from a dynamic environment. The PCSOM can also enable the lateral synaptic pruning, but this mechanism does not achieve a better accuracy on MNIST classification, and that's why we disabled it ($\omega = 0$ in table 3.2). Indeed, the hyper-parameter ω is very difficult to adjust so that we have the desired amount of pruning that separates the neurons clusters. Moreover, the neurons representing the same class are not always in the same topological neighborhood, as shown in figure 3.5-a. This is the natural behavior of the KSOM with the chosen hyper-parameters and the small number of neurons. It causes topologically close neurons whose weights are close at the beginning of the training to converge toward different classes at the end. Therefore, the learning with pruning is inefficient for our application with high-dimensional data. Furthermore, in order for the PCSOM to be totally dynamic, it needs to implement in addition a form of sprouting so that two neurons that would have been disconnected could reconnect to each other if the input stimuli change accordingly. However, one could think the opposite way: the pruning can be "stable" and even be done manually if we know from the beginning the number of classes that are present in our data. The idea is that the topological pruning will separate during training the neurons clusters of different classes and thus prevent them from learning from one another. It can be a solution to the catastrophic forgetting that occurs in ANNs, but this topic is out of the scope of this manuscript.

Some previous works attempted to use SOMs for MNIST classification, using complex multi-layered SOM structures to behave as receptive field in a local region of the input. They achieved classification accuracies of 80.46% (Liu, Wang, and Gong, 2015) and 82.10% (Wickramasinghe, Amarasinghe, and Manic, 2017) at the cost of thousands of neurons. We showed that we can achieve a better performance with a standard SOM architecture plus a well defined labeling mechanism with very few labels.

Compared to SNNs, the KSOM ($87.36\% \pm 00.23\%$) achieves a better accuracy than the the state of the art LMSNN ($85.71\% \pm 00.85\%$) for the same number of neurons (100). In addition, the SOMs only needs 1% of labeled samples for the neurons

TABLE 3.3: MNIST unsupervised learning with SOMs and SNNs.

Learning	ANN Model	# neurons	Labeled images		Test accuracy (%)
			Number	%	
STDP	SNN (Diehl and Cook, 2015)	100	60,000	100	80.71 ± 1.66
	LMSNN (Hazan et al., 2018)	100	60,000	100	85.71 ± 0.85
Kohonen	KSOM [Our work]	100	600	1	$\mathbf{87.36 \pm 0.23}$
	DSOM [Our work]	100	600	1	85.19 ± 0.54
	PCSOM [Our work]	100	600	1	84.53 ± 0.95

labeling. However, the LMSNN, just like the DSOM and PCSOM, enables dynamic learning, and the three models reach approximately the same accuracy. From a hardware perspective, the SNNs computation is shown to be hardware-efficient (Khacef, Abderrahmane, and Miramond, 2018). Nevertheless, even though the STDP itself is local, the excitatory/inhibitory mechanism that allows the STDP learning convergence implies an all-to-all connectivity that is not scalable, especially for embedded applications. We discuss the scalability performance of the SOM with the IG in section 3.5.

3.5 Scalability performance for hardware implementation

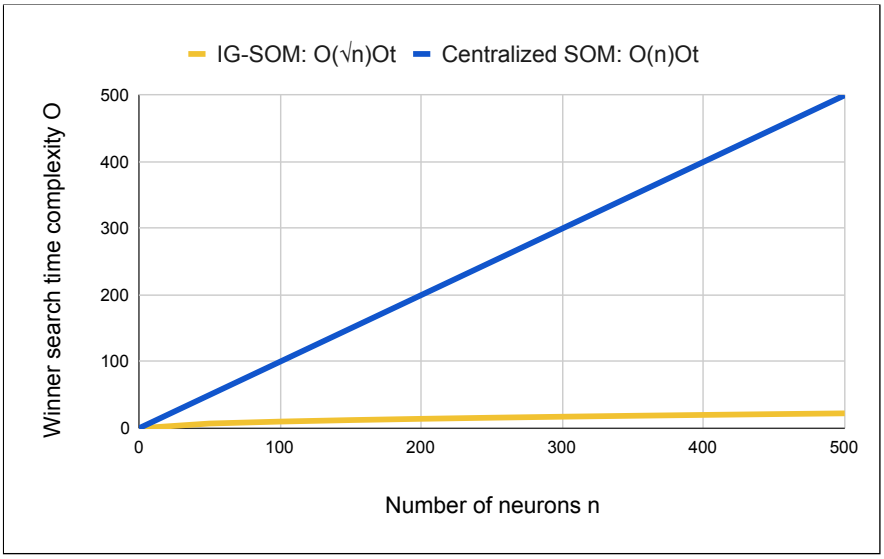

FIGURE 3.8: IG time complexity.

First, in order to express the time complexity of the IG formalism, we consider the case of square maps of $n = m \times m$ neurons. Let O_t be the computing time complexity of a neuron regardless of the simulated model. The standard centralized implementation must sequentially compute the equation of each neuron. That is to say $m \times m \times O_t$ and thus a time complexity in $O(m^2)O_t = O(n)O_t$. The IG implementation in software (sequentially simulated) adds $2m - 2$ propagation iterations for which each cell computes O_t, i.e. $2m \times m \times m \times O_t$ and thus a time complexity in $O(m^3)O_t = O(n\sqrt{n})O_t$. The fully distributed IG hardware implementation allows the cells to compute (the O_t) in parallel. It means that the m^2 factor is transformed into hardware resources and the time complexity becomes $O(m)O_t$. It is thus a square

root complexity in $O(\sqrt{n})O_t$ when expressed relatively to the number of neurons n. These results are illustrated in figure 3.8.

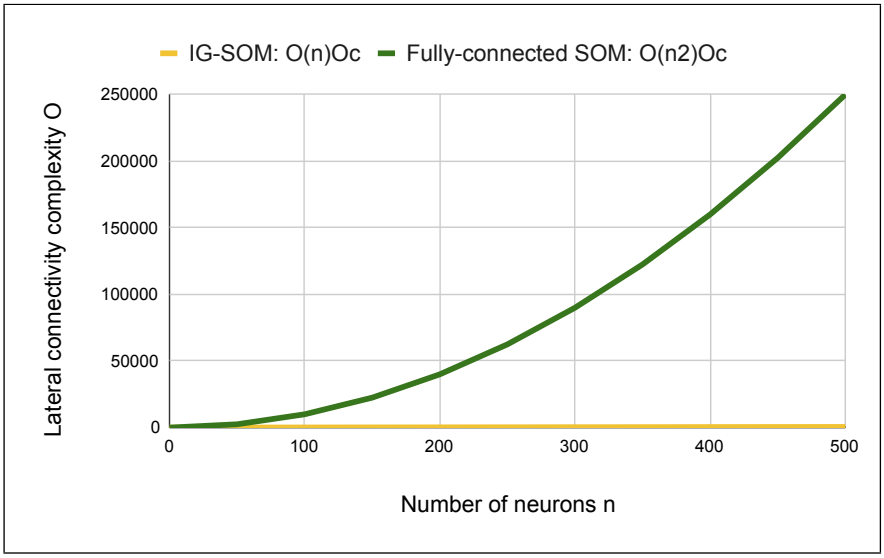

FIGURE 3.9: IG connectivity complexity.

Second, in order to express the connectivity complexity of the IG formalism, we consider the same case of square maps of n neurons. Let O_c be the connectivity complexity of a synapse between two neurons. The original distributed SOM (Kohonen, 1982) as well as the SNN (Diehl and Cook, 2015) with fully-connected architectures have $n \times n$ connections and thus a connectivity complexity in $O(n^2)O_c$. The fully distributed IG hardware implementation requires 4 connections per neuron (except for the border neurons which need less), so that each neuron is connected to each of its local neighbors by two connections. This way, two connected neurons can simultaneously read information from each other. The total number of connections is therefore $4n$, which means a linear connectivity complexity in $O(n)O_c$.

Here, the $O(n)$ complexity is studied in terms of time and connectivity, but since all cells are identical and because of the grid mesh structure of the IG substrata, the complexity in terms of chip area and power consumption also follows a linear curve (Rodriguez, Fiack, and Miramond, 2013) (Fiack, Rodriguez, and Miramond, 2015) (Khacef, Abderrahmane, and Miramond, 2018). Therefore, we can use the IG concept to design a scalable implementation for SOM-like models.

3.6 Conclusion

In the context of brain-inspired computing for unsupervised learning, we have reviewed the state of the art SNN models with STDP learning and different inhibition mechanisms. Then we have introduced the *post-labeled unsupervised learning* problem and proposed an automatic labeling method with three different variants, in order to assign each neuron the class it represents. Afterwards, we applied the KSOM, DSOM and PCSOM for training without labels, and showed that the best labeling method needs only 1% of labeled data and outperforms the qualitative manual labeling in terms of accuracy. We have then confronted the Kohonen-based SOMs with STDP-based SNNs on MNIST classification, and the KSOM achieves the best accuracy ($87.36\% \pm 00.23\%$) with the same number of neurons (100), as summarized in table 3.4.

TABLE 3.4: SOM vs. SNN: comparative study summary.

Criteria	SNNs	SOMs	SOMs with IG
Accuracy (100 neurons)	85.71%	87.36%	87.36%
Labeled training subset	100%	1%	1%
Unsupervised learning	Yes	Yes	Yes
Dynamic learning	Yes	Yes	Yes
Dsitributed computing	Yes	No	Yes
Local connectitivy (scalable)	No	No	Yes
Hardware cost	Very low	High	Low

Overall, the SOMs with IG offer a better compromise than SNNs with a better accuracy using very few labeled samples for labeling, as well as a scalable neuromorphic architecture. Nevertheless, SNNs remain interesting to their very low hardware cost (Abderrahmane, Lemaire, and Miramond, 2020) that needs to be further quantified. Before going into the multimodal association mechanism in chapter 5, we want to show that the SOM can achieve better results in MNIST and deal with more complex datasets such as natural images without increasing exponentially the number of neurons: we will instead use feature extraction.

Chapter 4

Improving the SOM performance with feature extraction

> Brains operate not by logic but by pattern recognition. This process is not precise [...]. Instead, it trades off specificity and precision, if necessary, to increase its range.
>
> *Gerald Edelman.*

4.1 Introduction

We propose in this chapter to improve the SOM performance by using extracted features instead of raw data in two different contexts: fully unsupervised learning and transfer learning. In the first part, we will conduct a comparative study on the SOM classification accuracy with unsupervised feature extraction using two different approaches: a *machine learning* approach with Sparse Convolutional Auto-Encoders using gradient-based learning, and a *neuroscience* approach with convolutional SNNs using STDP learning. The SOM is trained on the extracted features, then very few labeled samples are used to label the neurons with their corresponding class, as explained in chapter 3. We investigate the impact of the feature maps, the features sparsity, the SOM size and the labeled subset size on the MNIST handwritten digits classification accuracy using the different feature extraction methods. We also experiment a supervised Convolutional Neural Network (CNN) with the same topology for approximating the best accuracy we can expect from the feature extraction. In the second part, we use the SOM to classify more complex data: natural images in the mini-ImageNet few-shot classification task, a state of the art ML challenge where the goal is to train a classifier using a very limited number of labeled examples. This scenario is likely to occur frequently in real life when data acquisition or labeling is expensive. To address this problem, we consider an algorithm consisting in the concatenation of transfer learning with clustering using SOMs, then we demonstrate the ability of the proposed method in reaching top performance with the challenging benchmark of mini-ImageNet classification task. To speedup the SOM training process, we propose in appendix A a GPU-based SOM implementation with TensorFlow capable of running 100× faster on average on Nvidia GPUs compared to the standard CPU implementation. The complete GPU-based source code for the SOM training, labeling and test is available in `https://github.com/lyes-khacef/GPU-SOM`.

4.2 SOM on MNIST unsupervised classification

4.2.1 Unsupervised feature extraction

In this section, we review the related work and present the proposed methodology for unsupervised feature extraction.

Sparse Convolutional AutoEncoders (SCAE)

Introduced by Rumelhart, Hinton and Williams (Rumelhart, Hinton, and Williams, 1988), AEs were designed to address the problem of back propagation without supervisor via taking the input data itself as the supervised label (Baldi, 2012). Today, AEs are typically used for dimensionality reduction or weights initialization in CNNs to improve the classification accuracy (Masci et al., 2011) (Kohlbrenner, 2017). In this work, we want to use AEs as feature extractors with unsupervised learning. In such cases, the feature map representation of a Convolutional AE (CAE) is most of the time of a much higher dimensionality than the input image. While this feature representation seems well-suited in a supervised CNN, the so-called overcomplete representation becomes problematic in an AE since it gives the autoencoder the possibility to simply learn the identity function by having only one weight "on" in the convolutional kernels (Masci et al., 2011). Without any further constraints, each convolutional layer in the AE could easily learn a simple point filter that copies the input onto a feature map (Kohlbrenner, 2017). While this would later simplify a perfect reconstruction of the input, the CAE does not find any more suitable representation for our data. To prevent this problem, some constraints have to be applied in the CAE to increase the sparsity and therefore the saparability of the features.

The concept of sparsity was introduced in computational neuroscience, as sparse representations resemble the behavior of simple cells in the mammalian primary visual cortex which is believed to have evolved to discover efficient coding strategies (Olshausen and Field, 1997). It has been proven that encouraging sparsity when learning the transformed representation can improve the performance of classification tasks (Hoyer, 2004). Indeed, the overcomplete architecture of a CAE allows a larger number of hidden units in the code, but this requires that for the given input, most of hidden neurons result in very little activation (Ng, 2011). In a Sparse CAE (SCAE), activations of the encoding layer need to have low values in average. Units in the hidden layers usually do not fire (Charte et al., 2018) so that the few non-zero elements represent the most salient features (Ng, 2011).

In order to increase the sparsity of the CAE's feature representation, several methods can be found in the literature. In (Masci et al., 2011), the authors use max-pooling to enforce the learning of plausible filters, but the filters are then fine-tuned with supervised learning for the classification. Since we do not want to use any label in the training process, we apply additional constraints in the SCAE, namely weights and activity constraints of types L2 and L1, respectively (Nan Jiang et al., 2015).

Convolutional SNNs (CSNNs)

As explained in the previous chapters, SNNs are a brain-inspired family of ANNs used for large-scale simulations in neuroscience (Furber et al., 2014) and efficient hardware implementations for embedded AI (Davies et al., 2018). They are characterized by the spike-based information coding, a computational model of the electrical impulses amongst the biological neurons. The amplitude and duration of all spikes are almost the same, so they are mainly characterized by their emission time

(Kheradpisheh et al., 2018). Furthermore, spiking neurons appear to fire a spike only when they have to send an important message, which leads to the fast and extremely energy-efficient neural computation in the brain.

Moreover, SNNs have a great potential for unsupervised learning through STDP (Diehl and Cook, 2015), a biologically plausible local learning mechanism that uses the spike-timing correlation to update the synaptic weights. Kheradpisheh et al. proposed in (Kheradpisheh et al., 2018) a SNN architecture that implements convolutional and pooling layers for spike-based unsupervised feature extraction. The SNN processes image inputs as follow. The first layer of the network uses Difference of Gaussians (DoG) filters to detect contrasts in the input image. It encodes the strength of the edges in the latencies of its output spikes, i.e. the higher the contrast, the shorter the latency. On the one hand, neurons in convolutional layers detect complex features by integrating input spikes from the previous layer, and emit a spike as soon as they detect their "preferred" visual feature. A WTA mechanism is implemented so that the neurons that fire earlier perform the STDP learning and prevent the others from firing. Hence, more salient and frequent features tend to be learned by the network. On the other hand, neurons in the pooling layers provide translation invariance by using a temporal maximum operation, and help the network to compress the flow of visual data by propagating the first spike received from neighboring neurons in the previous layer which are selective to the same feature. However, in (Kheradpisheh et al., 2018), the extracted features were classified using a supervised Support Vector Machine (SVM). In this work, we use the unsupervised SOM classifier to keep the unsupervised training from end to end.

4.2.2 CNN, SCAE and SNN training methods

In order to compare the feature extraction performance, we use the topologies shown in table 4.1 for the three approaches.

TABLE 4.1: CNN, SCAE and SNN feature extractors topologies.

Model	Topology
CNN	$28 \times 28 \times 1 - 64c5 - Xc5 - p5$
SCAE	$28 \times 28 \times 1 - 64c5 - Xc5 - p5$
SNN	$28 \times 28 \times 1 - 64c5 - p2 - Xc5 - p2$

We therefore use two convolutional layers of 64 maps and X maps respectively. Each one uses 5×5 kernels followed by a max-pooling layer. The reason for the different pooling mechanism of the SNN is explained in the following. We explore the impact of the number of features X on the classification accuracy.

CNN training

The CNN is modeled in TensorFlow/Keras (Abadi et al., 2016) (Chollet et al., 2015) and trained with Adadelta (Zeiler, 2012) gradient-based algorithm for 100 epochs with a learning rate of 1.0. Since the goal is to estimate the maximum accuracy we can expect from each topology, the CNN is trained with the labeled training set by using 10 neurons with a Softmax activation function on top of the last pooling layer. Even though the CNN uses the sparsity constraints described in the following, this network is simply noted as CNN+MLP in the rest of this section.

SCAE training

The SCAE is also modeled in TensorFlow/Keras and trained using Adadelta (Zeiler, 2012) gradient-based algorithm for 100 epochs with a learning rate of 1.0. However, no label is used in the training process, as the goal of the SCAE is to reconstruct the input in the output. The complete SCAE topology is $28 \times 28 \times 1 - 64c5 - Xc5 - p5 - u5 - 64d5 - 1d5$, where u stands for up-sampling and d stands for deconvolution (or transposed convolution) layers. The complete architecture is thus symetric. We add to every convolution and deconvoltion layer a weight constraint of type $L2$ ($\lambda \sum_{i=0}^{max} w_i^2$), and we add to the second convolution layer that produces the features an activity constraint of type $L1$ ($\lambda \sum_{j=0}^{max} \|a_j\|$). The weights and activity regularisation rates λ are set to 10^{-4}. Therefore, the objective function of the SCAE takes in account both the image reconstruction and the sparsity constraints.

SNN training

The SNN is modeled in SpykeTorch (Mozafari et al., 2019), an open-source simulator of convolutional SNNs based on PyTorch (Paszke et al., 2019). The SNN is trained with STDP layer by layer, with a different pooling mechanism than the CNN and SCAE. Except for the number of feature maps and kernel sizes, we kept the same hyper-parameters as the original implementation of (Kheradpisheh et al., 2018) that can be found on (Mozafari et al., 2019). Hence, we used a pooling layer of 2×2 after each convolutional layer, with a padding of 1 before the second convolutional layer. The threshold of the neurons in the last convolutional layer were set to be infinite so that their final potentials can be measured (Kheradpisheh et al., 2018). Finally, the global pooling neurons compute the maximum potential at their corresponding receptive field and produce the features that will be used as input for the SOM. Our experimental study showed that the added padding and the pooling mechanism proposed in (Mozafari et al., 2019) performs better than the one used in the CNN and SCAE (i.e. no padding and one pooling layer), with a gain of 1.43% on the maximum achievable accuracy.

4.2.3 Confronting SCAE, SNN and CNN feature extraction with a SOM classifier

Comparative study: feature maps, SOM neurons and labels

For simplicity, we refer as SOM to the original KSOM described in chapters 2 and 3. The following SOM training hyper-parameters for the different settings were found with a grid search: $\epsilon_i = 1.0$, $\epsilon_f = 0.01$, $\eta_i = 10.0$, $\eta_f = 0.01$, $\alpha = 1.0$ and the number of epochs is 10.

First, figure 4.1 shows the impact of the number of feature maps in the second convolutional layer, using 256 neurons in the SOM and 10% of labels. We deliberately use a large number of labels to avoid any bias due to the labeling performance, and focus on the impact of the feature maps. The accuracy of the CNN+SOM and SCAE+SOM is increasing with respect to the number of feature maps, reaching a maximum at 256 maps. Interestingly, the CNN+SOM performs better with 8 maps (97.56%) than with 16 (97.25%), 32 (97.00%), 64 (97.26%) or 128 (97.31%) maps. This is due to the tradeoff between additional information and additional noise induced by more feature maps according to the SOM classification. In fact, the CNN+MLP supervised baseline accuracy is increasing from 98.7% to 99% when the feature maps increase from 8 to 512. This observation is more pronounced when we look at the

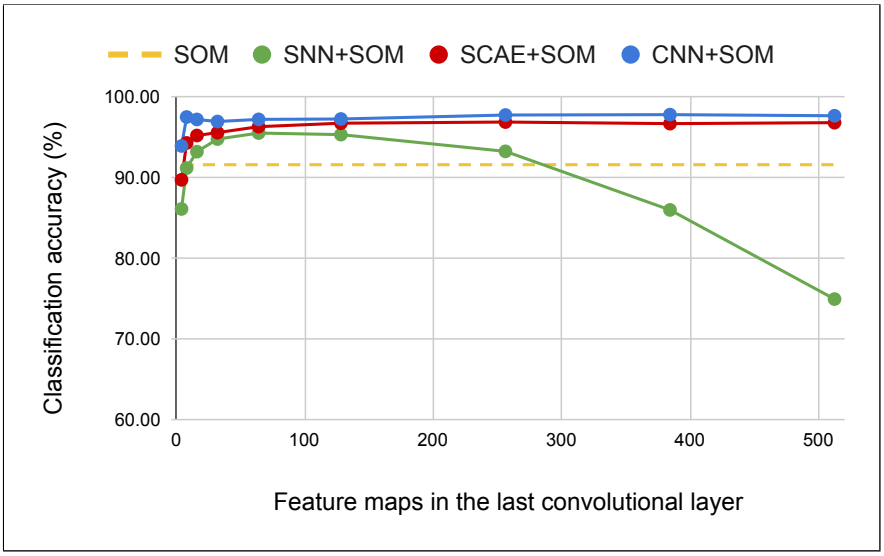

FIGURE 4.1: SOM classification accuracy on MNIST using CNN, SCAE and SNN feature extraction vs. number of feature maps with 256 SOM neurons and 10% of labels.

SNN+SOM that reaches a maximum accuracy for 64 maps then drastically decreases with more feature maps. Following the approach of (Kheradpisheh et al., 2018), we used a SNN+SVM supervised baseline and its accuracy increases from 97% to 98% when the feature maps increase from 64 to 512. It means that the increasing number of feature maps for the SNN produces noisy features that do not affect the supervised classification but do decrease the unsupervised classification accuracy. This is due to the overlapping of the SOM prototypes overlap that become then less descriminative. In fact, this behavior is either due to the STDP learning or to the spike coding paradigm of SNN. In order to eliminate the wrong answer, we trained a SNN with spike-based surrogate gradient in SpykeTorch (Mozafari et al., 2019) and varied the number of feature maps for comparison with the SNN+SOM in figure 4.1. We found that the gradient-based SNN+SOM is reaching an accuracy plateau of 98.2% starting from 256 feature maps with 256 neurons. Thus, the decreasing performance with large features for the SNN+SOM is not due the spike coding but to the STDP local learning. We observe that the STDP-based SNN+SOM with large features does not converge well with the chosen hyper-parameters. Finally, we choose 256 maps for the CNN and SCAE that produce a feature size of 4096, and 64 maps for the SNN that produces feature maps of size 3136. We remark that the SNN features size is different from the CNN/SCAE features size, which is due to the added padding and the different pooling mechanism as explained in section 4.2.2.

Second, with the above mentioned topologies, we investigated the impact of the SOM size with 10% of labels, from 16 to 10,000 neurons. We see in figure 4.2 that the accuracy of the four systems is increasing with respect to the number of neurons. We notice that the SNN-SOM reaches the same accuracy as the SCAE+SOM starting from 1024 neurons. Nevertheless, for the next step of the study, it is important to keep the same number of neurons. Hence, we have chosen the number of neurons for which one of the SCAE+SOM or SNN+SOM reaches the maximum accuracy, which is equal to 256 neurons with respect to the SCAE+SOM accuracy.

Third, using 256 neurons for the SOM, we investigated the impact of the labeling subset size in terms of % of the training set. Figure 4.3 shows that the accuracy increases when the labeled subset increases. Interestingly, the CNN+SOM and

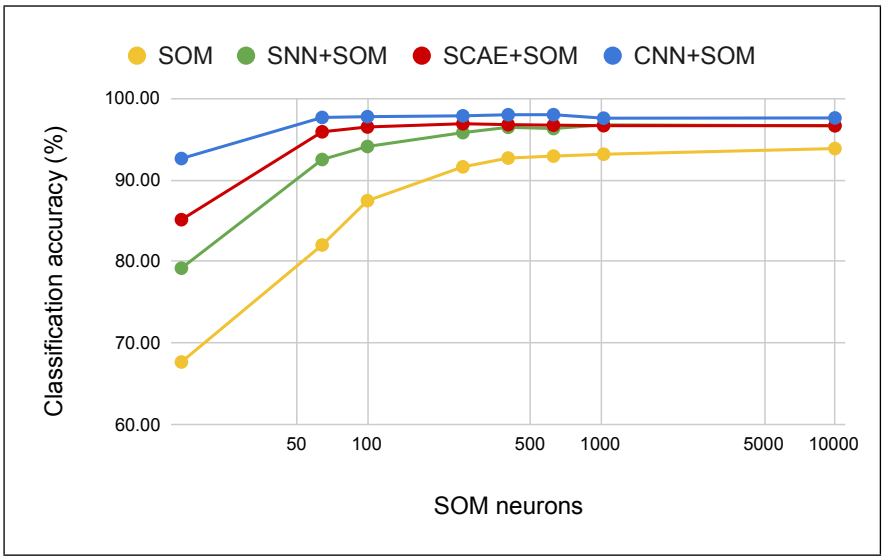

FIGURE 4.2: SOM classification accuracy on MNIST using CNN, SCAE and SNN feature extraction vs. number of SOM neurons with the optimal topologies and 10% of labels.

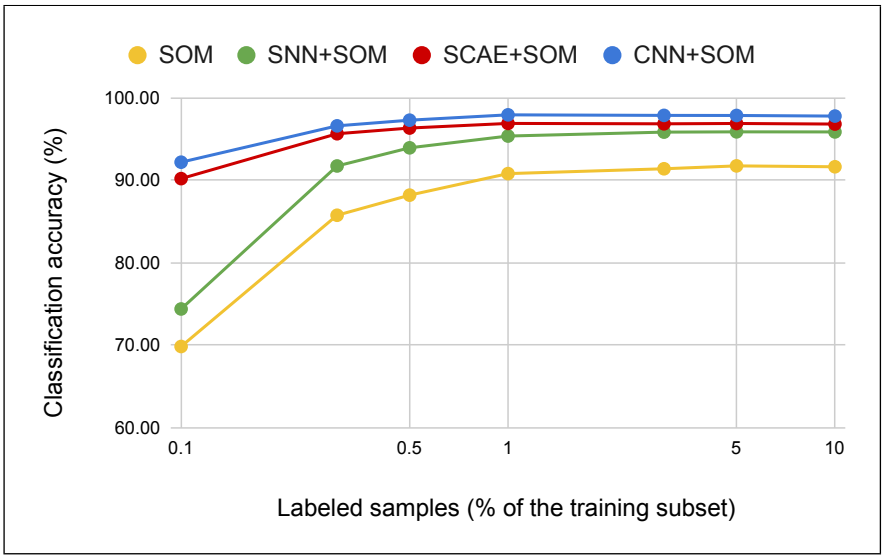

FIGURE 4.3: SOM classification accuracy on MNIST using CNN, SCAE and SNN feature extraction vs. % of labeled data from the training subset for the neurons labeling with the optimal topologies and 256 SOM neurons.

SCAE+SOM reach their maximum accuracies with only 1% of labeled data, while the SNN+SOM and SOM need approximately 5% of labeled data. Since the SCAE+SOM performs better than the SNN+SOM, we only need 1% of labeled data. It confirms the results obtained in chapter 3.

Finally, the comparative study of the four settings with their best topologies, using 256 neurons for the SOM and 1% of labeled data for the neurons labeling is summarized in figure 4.4. As expected, the SOM without feature extraction has the worst accuracy of 90.81% $\pm$ 0.15 and the CNN+SOM with supervised feature extraction reaches the best accuracy of 97.94% $\pm$ 0.22. More interestingly, with fully unsupervised learning, the SCAE performs better than the SNN (+1.53%), with 96.9% $\pm$ 0.24

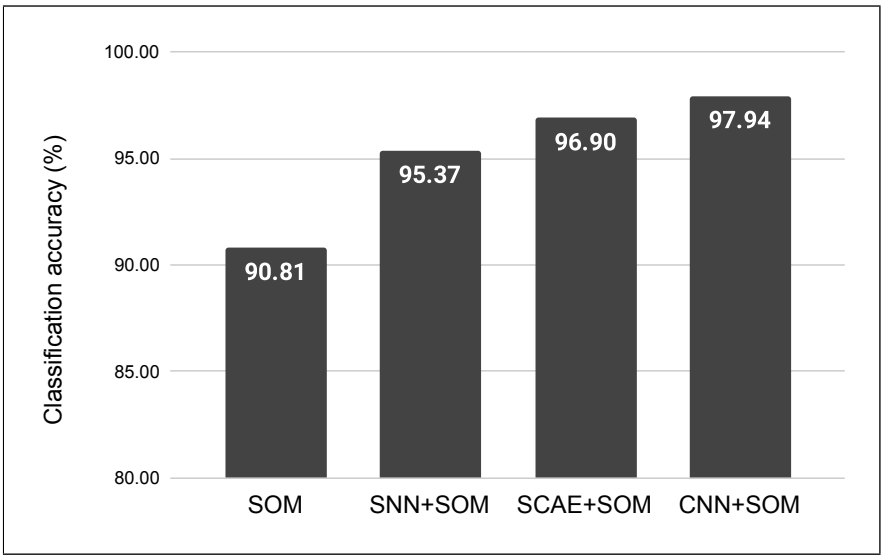

FIGURE 4.4: SOM classification accuracy on MNIST using CNN, SCAE and SNN feature extraction: summary of the comparative study with the optimal topologies, 256 SOM neurons and 1% of labels.

and 95.37% $\pm$ 0.58 respectively. In the following section, we try to investigate the role of sparsity in the features separability and classification with the SOM.

Features sparsity investigation

Sparsity is commonly imposed in AEs, especially when the set of features is overcomplete, i.e. the dimension of the feature is larger than the dimension of the input (Nan Jiang et al., 2015). As discussed before, it implies that most units take values close to zero while only few take significantly non-zero values. In this section, we quantifity the sparsity of the extracted features of the CNN, SCAE and SNN using the sparseness measure proposed in (Hoyer, 2004). It is based on the relationship between the L1 norm and the L2 norm as expressed in equation 4.1.

$$Sparsity(f) = \frac{\sqrt{n_f} - \frac{\sum_{i=0}^{n_f} |f_i|}{\sqrt{\sum_{i=0}^{n_f} f_i^2}}}{\sqrt{n_f} - 1} \tag{4.1}$$

It is to note in equation 4.1 that f is the feature vector and n_f is the dimension of f. The sparsity value lies between 0 and 1: a vector with all elements equal has a sparsity of 0, whereas a vector with only a single non-zero component has a sparsity of 1. It means that larger values indicate sparser features.

The sparsity results are reported in table 4.2. First, if we look at the unsupervised models, we see that the SCAE is sparser than the SNN which is in turn sparser than the CAE. The SNN is not the sparsest model because even though each neuron can only spike once, it does not prevent the other neurons to spike as well. The SNN features are therefore binary with only 0 and 1 values but less sparse compared to the SCAE.

TABLE 4.2: Features sparsity: comparative study.

Features learning	Model	Sparsity constraints	Sparsity	SOM accuracy (%)
Unsupervised	CAE	No	0.35	94.9
	SNN	Yes[1]	0.57	95.4
	SCAE	Yes[2]	0.78	96.9
Supervised	CNN	No	0.42	97
		Yes[2]	0.92	97.9

[1] WTA inhibition mechanism.
[2] L1 and L2 constraints on activities and weights, respectively.

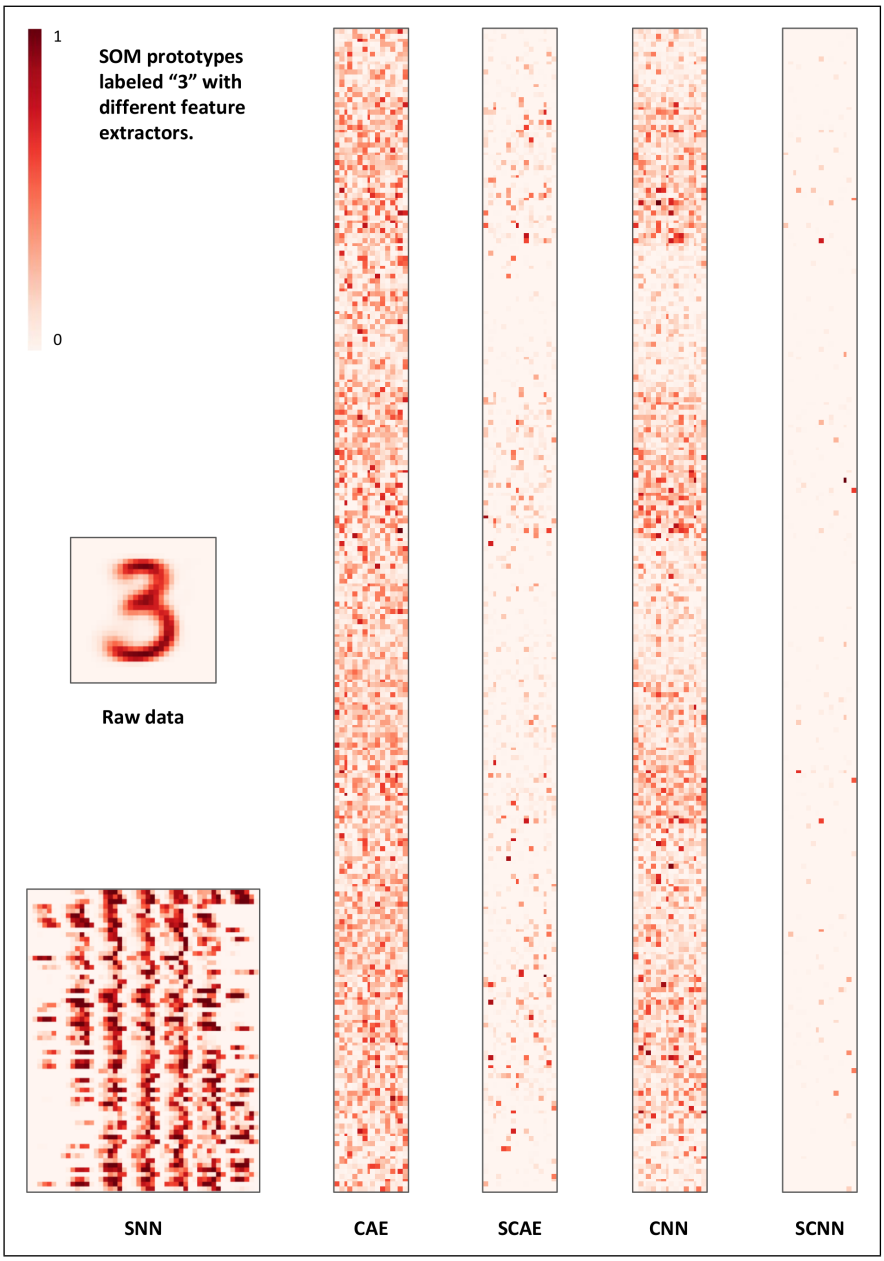

FIGURE 4.5: SOM prototypes with different features extractors on MNIST.

This is illustrated in figure 4.5 that shows SOM prototypes labeled "3" with the different feature extractors. The SCNN stands for sparse CNN which is simply noted as CNN in the rest of the section. It is to note that the SNN features ($64 \times 49 = 3136$) are smaller than the others ($256 \times 16 = 4096$) because of the different pooling mechanism explained in section 4.2.2. Also, even though the SNN features are binary, their prototypes are not binary since they are centroids of a cluster of features. Furthermore, we can notice some similar characteristics amongst the CAEs and CNNs with or without sparsity constraints, in the form of horizontal bars corresponding to the activations of the same weighted neurons connected to different visual fields.

Overall, the most important remark for the unsupervised models is that the increase in sparsity is reflected in a better accuracy. The same remark is correct when we look at the supervised models alone, namely the CNN without and with sparsity constraints. We can therefore conclude that within the same learning paradigm for the feature extraction, unsupervised or supervised, the features sparsity measure is a good indicator that is strongly correlated with the SOM classification accuracy.

However, this conclusion cannot be extended in general, because of the counter-example of the CNN without sparsity: its features are less sparse (0.42) than those of the SNN (0.57) and the SCAE (0.78), but the CNN features classification accuracy with the SOM is better. Our results are in the continuity of the literature findings in two aspects. First, although sparsity is a desirable property for good representation, an excessive level of sparsity can be detrimental (Nan Jiang et al., 2015) (Falez et al., 2019), that's why we had to optimize the λ hyper-parameter for the CNN and SCAE trainings in section 4.2.2. Second, the sparsity measure alone is not a sufficient indicator to assess for the quality of features before classification. Indeed, we have seen that the sparsity is not always correlated with the classification accuracy, especially when comparing features that are extracted with different learning paradigms, i.e. supervised and unsupervised learning.

Summary

TABLE 4.3: Comparison of unsupervised feature extraction and classification techniques in terms of accuracy and hardware cost.

Feature extraction		Classification		Performance		
Model	**Learning**	**Model**	**Learning**	**Accuracy (%)**	**Error (%)**	**Hardware cost**
CNN	Supervised	MLP	Supervised	99.00	1.00	High
CNN	Supervised	SOM	Unsupervised	97.94	2.06	Medium
SCAE	Unsupervised	SOM	Unsupervised	96.90	3.10	Medium
SNN	Unsupervised	SOM	Unsupervised	95.37	4.63	Low

To summarize, we can see in table 4.3 the gap between supervised and unsupervised methods for feature extraction and classification. Interestingly, we only lose about 1% of accuracy when going from CNN+MLP to CNN+SOM, and another 1% when going from CNN+SOM to SCAE+SOM. The gap is slightly higher when going from SCAE+SOM to SNN+SOM, which is about 1.5%. In return, the hardware cost decreases when using SOMs and SNNs, thanks to the brain-inspired computing paradigm (distributed and local). Indeed, we showed in (Khacef, Abderrahmane, and Miramond, 2018) that the SNN has a gain of approximately 50% in hardware resources and power consumption when implemented in dedicated FPGA and ASIC hardware. However, this study was performed on fully connected layers only, and

needs to be extended to convolutional and pooling layers to have a more precise quantification of the hardware gain we can expect from convolutional SNNs. Moreover, the SNN features are binary, which provides an additional gain in the input memory footprint.

TABLE 4.4: MNIST unsupervised learning with AE-based feature extraction: state of the art reported from (Ji, Vedaldi, and Henriques, 2018) and completed.

Method	Accuracy (%)
AE + K-means (Bengio et al., 2006)	81.2
Sparse AE + K-means (Ng, 2011)	82.7
Denoising AE + K-means (Vincent et al., 2010)	83.2
Variational Bayes AE + K-means (Kingma and Welling, 2013)	83.2
SWWAE + K-means (Zhao et al., 2015)	82.5
Adversarial AE (Makhzani et al., 2015)	95.9
Sparse CAE + SOM [Our work]	**96.9**

Overall, the SCAE+SOM reaches the best accuracy of $96.9\% \pm 0.24$ on MNIST classification with unsupervised learning. As shown in table 4.4, we achieved state of the art accuracy compared to similar works that followed an AE-based approach. The sparsity constraints of the SCAE through the weights and activities regularization significantly improved the SOM classification accuracy. Indeed, without these constraints, the CAE+SOM with the same configuration achieves an accuracy of $94.9\% \pm 0.24$, which means a loss of -2% and a less good performance compared to the SNN+SOM.

A similar comparative study was conducted in (Falez et al., 2019), but the study was limited to one layer SCAE and SNN, and a supervised SVM was used to assess the classification accuracy. The authors concluded that the SCAE reaches a better classification accuracy. Our study extends their finding to multiple convolutional layers by using unsupervised learning for both feature extraction and classification. Nevertheless, the SNN+SOM remains attractive due to the hardware-efficient computation of spiking neurons (Khacef, Abderrahmane, and Miramond, 2018) and the possible association to the cellular neuromorphic architecture of the SOM presented in chapter 2.

4.3 SOM on mini-ImageNet few shot classification

4.3.1 Few-shot classification: state of the art approaches

In the last decade, DL techniques have achieved state of the art performance in many classification problems. However, DL heavily relies on supervised learning with abundant labeled data. As discussed before, the fast expansion of IoT devices gathers a huge amount of unlabeled data everyday, but labeling these data is a very difficult task because of the human annotation cost as well as the scarcity of data in some classes (Chen et al., 2019). Finding methods to learn to generalize to new classes with a limited amount of labeled examples for each class is therefore a very active topic of research in ML. This is the main motivation for few-shot learning (Hu, Gripon, and Pateux, 2020). Recently, three main approaches have been proposed in the literature:

- **Hallucination methods** where the aim is to augment the training sets by learning a generator that can create novel data using data-augmentation techniques (Chen et al., 2019). However, these methods lack precision which results in coarse and low-quality synthesized data that can sometimes lead to very poor gains in performance (Wang et al., 2019).
- **Meta-learning** where the goal is to train an optimizer that initializes the network parameters using a first generic dataset, so that the model can reach good performance with only a few more steps on the new dataset (Thrun and Pratt, 2012). This type of solution suffers from the domain shift problem (Chen et al., 2019) as well as the sensitivity of hyper-parameters.
- **Transfer learning** where a model developed for a given task is reused as the starting point for a model on a different task. In real-world problems, it happens that we have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest. Therefore, knowledge transfer would greatly improve the performance of learning by avoiding much expensive data-gathering and data-labeling efforts (Pan and Yang, 2010). Hence, transfer learning has emerged as the new learning framework for the few-shot classification task.

The problem becomes even harder when facing technical limitations, such as using embedded implementations for real-time processing on the edge. As a matter of fact, in many real-world scenarios, the training data is acquired using the same device that will later be used for training and inference, and labels could be given at any time of the process. To encompass for this added difficulty, we consider in this work the problem of post-labeled few-shot unsupervised learning. In this problem, learning algorithms can be deployed using no annotated data, for example to learn representations using the data acquired by the considered device. These algorithms can later be "adjusted" using a few labeled samples so that they become able to make predictions, at the condition that this adjustment comes with almost no added complexity to the process, so that it can be performed on the edge. Compared to the post-labeled unsupervised learning problem introduced in chapter 3, the only difference is that we know at the beginning the "few shot" labels we can use.

To address this problem, we propose a solution that combines transfer learning and unsupervised learning with the SOM. On the one hand, transfer learning is used to exploit a DNN trained on a large collection of labeled data as a "universal" feature extractor. On the other hand, the SOM is used to leverage the obtained features and make predictions. Following the same approach defined in chapter 3, this algorithm works in two steps: first, clusters prototypes are learned using no annotated data, then the prototypes are labeled using the few available annotated samples.

4.3.2 Transfer learning for feature extraction

We consider that we are given an unlabeled dataset $X = \{x, x \in X\}$. Our first step consists in extracting relevant features from these inputs. For this purpose, we follow the approach proposed by (Hu, Gripon, and Pateux, 2020) and train a supervised feature extractor f_φ that we call a *backbone* on a large annotated dataset. The parameters of the backbone are then fixed and used to obtain *generic* features from any input. In our case, we therefore transform X into $V = f_\varphi(X) = \{f_\varphi(x), x \in X\}$.

We perform our experiments using the mini-ImageNet (Vinyals et al., 2016) benchmark. mini-ImageNet is a subset of ImageNet (Russakovsky et al., 2015) that contains 60,000 images divided into 100 classes of 600 images, each image has 84×84 pixels. Following the standard approach (Ravi and Larochelle, 2017), we use 64 base classes with labels to train the backbone, 16 base classes for validation and 20 novel classes to draw the novel datasets from. For each run, 5 classes are drawn uniformly at random among these 20 classes, then q unlabeled inputs and s labelled inputs per class are chosen uniformly at random among the 5 drawn classes. The features of the $(q+s) \times 5$ samples are used to train the SOM, then the s labeled samples are used to label the SOM neurons. Finally, the $Q = q \times 5$ unlabeled samples are classified and produce a classification accuracy for each run. We run 10,000 random draws to obtain a mean accuracy score and indicate the confidence scores (95%) when relevant.

The feature extractor we use is the same as in (Hu, Gripon, and Pateux, 2020). It is mostly based on a Wide Residual Network (WRN) (Zagoruyko and Komodakis, 2016) as a backbone extractor, with 28 convolutional layers and a widening factor of 10. As a result, the output feature size (the dimension of a vector $v \in V$) is 640. Let us insist on the fact the backbone is trained on a completely disjoint dataset with the tasks we consider thereafter.

The next steps consist in training, labeling and testing the SOM using the transformed representations in V, i.e. the extracted features. In transfer learning, the backbone feature extractor is trained and validated with 80 classes that are different from the 20 classes we classify using the SOM. Hence, the features amplitude is not relevant, and the Euclidean distance of the SOM does not provide the best performance. Therefore, we replace the Euclidean distance in equation 2.1 with the Cosine distance in Equation 4.2.

$$d = 1 - \cos(v, w_n) = 1 - \frac{v.w_n}{\|v\| \times \|w_n\|} \tag{4.2}$$

The Cosine distance is also used in the labeling and test phases. The comparison to Euclidean distance is discussed in section 4.3.3.

4.3.3 mini-ImageNet few-shot classification performance

The following SOM training hyper-parameters for transfer learning were found with a grid search: $\epsilon_i = 1.0$, $\epsilon_f = 0.01$, $\eta_i = 10.0$, $\eta_f = 0.1$, $\alpha = 1.0$ and the number of epochs is 10.

First, we investigated the impact of the SOM size on the classification accuracy for the commonly used number of unlabeled samples $q = 15$ and labeled samples $s = [1, 3, 5]$ (Hu, Gripon, and Pateux, 2020). Figure 4.6 shows that there is an optimal point at 25 neurons for $s = 1$ and 100 neurons for $s = 3$ and $s = 5$. There is a tradeoff between the number of neurons that learn different prototypes and the quality of the learning/labeling of these neurons. The more neurons we have, the more potential to learn different prototypes of the data but the more fuzzy the prototypes become, which makes the labeling part more difficult. For example, a neuron may be assigned a class "A" with respect to the labeled subset, but will be more active for a class "B" with respect to the test set. When we only have one labeled sample per class, i.e. $s = 1$, then a SOM of only 25 neurons achieves the best accuracy because more neurons will not converge as well.

Next, we varied the number of unlabeled data $Q = q \times 5$ with the above mentioned SOM sizes. Figure 4.7 shows that even though the labels are only used for the neurons class assignment and not in the training process, they still have a large

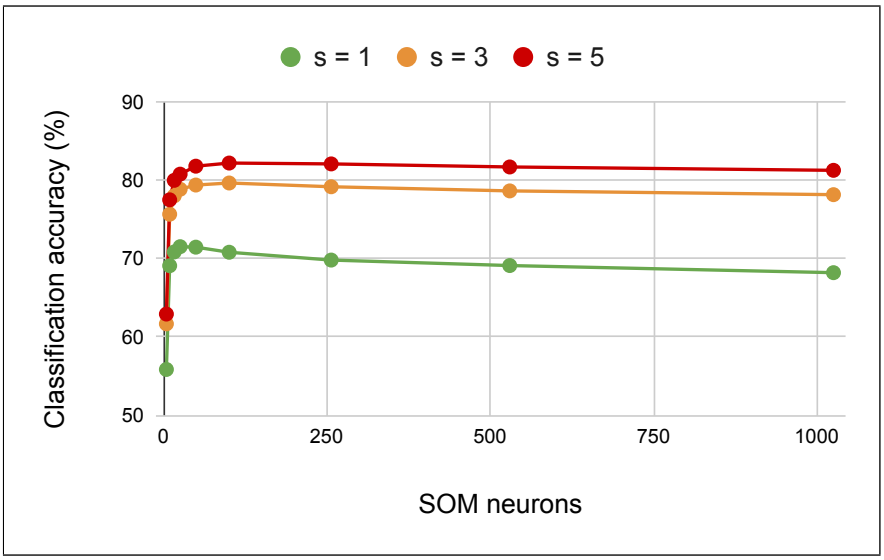

FIGURE 4.6: SOM classification accuracy on mini-ImageNet transfer learning for different numbers of labeled samples s vs. number of SOM neurons.

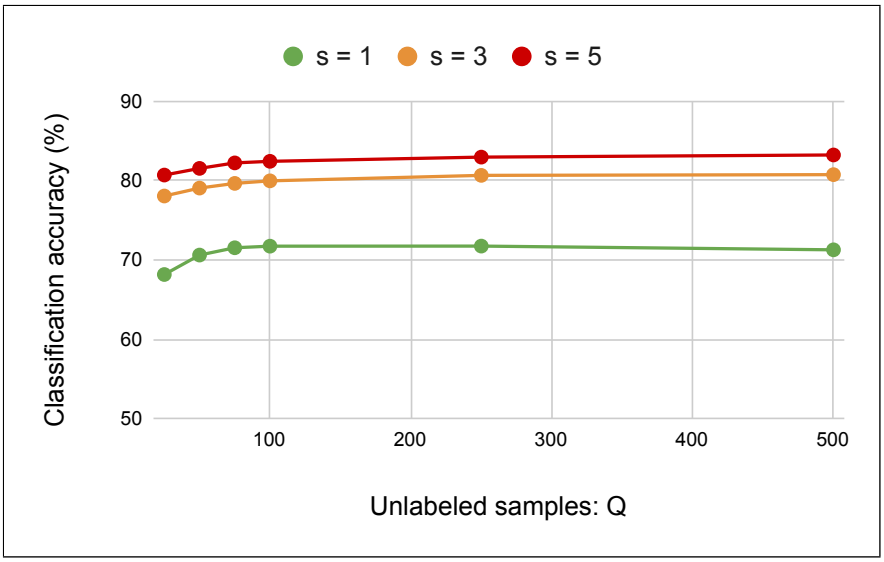

FIGURE 4.7: SOM classification accuracy on mini-ImageNet transfer learning for different numbers of labeled samples s vs. number of unlabeled samples to classify Q.

impact on the accuracy. Naturally, the more labeled data we have, the better accuracy we get. A second remark is that the more unlabeled data we have, the better accuracy we get too. This is not intuitive, because the unlabeled data are the queries, i.e. the samples to classify, so the more we have the harder the classification task becomes. However, since the SOM is trained on these data, its adaptation capabilities makes the accuracy increase with the number of unlabeled data for the same number of labels. The only exception is when $s = 1$, where there is a small decrease in accuracy between $Q = 250$ ($71.74\% \pm 0.21$) and $Q = 500$ ($71.27\% \pm 0.21$). A third remark is that the SOM reaches the same accuracy of 80.6% for $[s = 5, Q = 25]$ and $[s = 3, Q = 250]$, which means that the lack of labeled data can be compensated by more unlabeled data. In fact, it is a very interesting property since unlabeled

data can be gathered much more easily, and no extra-effort for labeling these data is needed.

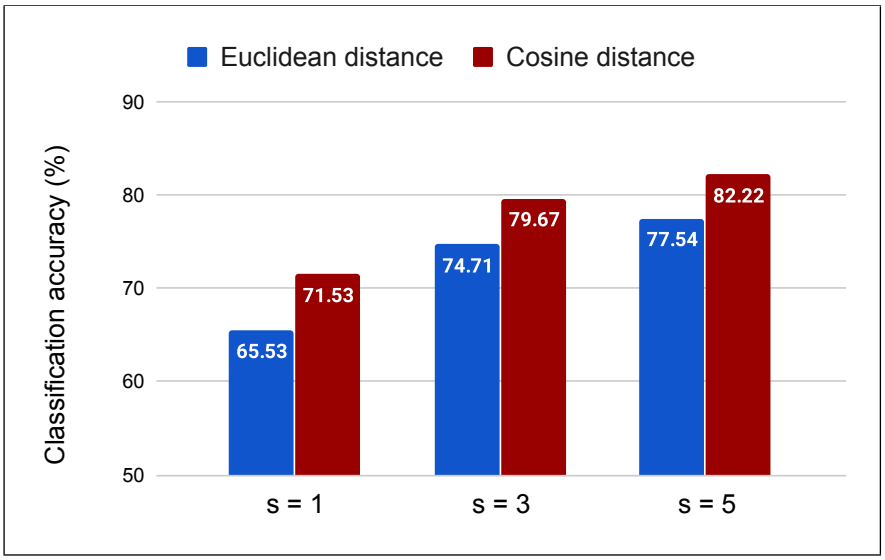

FIGURE 4.8: SOM classification accuracy on mini-ImageNet transfer learning with few labels using Euclidean distance and Cosine distance.

The choice of using the Cosine distance in the SOM computation (training, labeling and test) was inspired from the work of (Hu, Gripon, and Pateux, 2020). In fact, figure 4.8 shows that replacing the Euclidean distance by the Cosine distance significantly improves the SOM classification accuracy, with a gain of +5.9%, +4.96% and +4.68% for s = 1, s = 3 and s = 5, respectively. It validates our hypothesis about the non-effectiveness of the Euclidean distance when using transfer learning.

TABLE 4.5: mini-ImageNet few labels transfer learning with a WRN backbone and q = 15 (Q = 75): state of the art reported from (Hu, Gripon, and Pateux, 2020) and completed.

Method	**Classifier**	**1-shot (%)**	**5-shot (%)**
wDAE-GNN (Gidaris and Komodakis, 2019)	Supervised	61.07 ± 0.15	76.75 ± 0.11
ACC+Amphibian (Snell, Swersky, and Zemel, 2017)	Supervised	64.21 ± 0.62	$\mathbf{87.75 \pm 0.73}$
BD-CSPN (Liu, Song, and Qin, 2019)	Supervised	70.31 ± 0.93	81.89 ± 0.60
Transfer+SGC (Hu, Gripon, and Pateux, 2020)	Supervised	$\mathbf{76.47 \pm 0.23}$	85.23 ± 0.13
Transfer+SOM [Our work]	**Unsupervised**	71.53 ± 0.23	82.22 ± 0.15

Finally, table 4.5 reports the recent works that proposed solutions to the mini-ImageNet few labels classification problem using transfer learning with the WRN backbone feature extractor. The SOM reaches top-2 accuracy for $s = 1$ and top-3 accuracy for $s = 5$, which is a good result that proves the SOM ability to handle complex datasets. Nevertheless, one has to keep in mind that while the other works use the few labels in the training process, we only use them for neurons labeling phase. Our accuracy performance is therefore obtained with fully unsupervised learning followed by post-labeling. We argue that this is the right approach for the few-shot classification problem, especially in the context of embedded systems on the edge.

4.4 Conclusion

In this chapter, we have demonstrated the ability of the SOM to improve its accuracy on MNIST and achieve top performance on the challenging mini-ImageNet task when using extracted features instead of the raw data. First, in the context of unsupervised learning, we conducted a comparative study for unsupervised feature extraction, and concluded that the SCAE+SOM achieves a better accuracy thanks to the sparsity constraints that were applied to the SCAE through weights and activities regularization. However, the SNN+SOM remains interesting due to the hardware efficiency of spiking neurons. We improved the SOM classification by $+6.09\%$ and achieved state of the art performance on MNIST unsupervised classification, using post-labeled unsupervised learning. Second, in the context of transfer learning, we proposed a solution that combines transfer learning and SOMs. Transfer learning was used to exploit a WRN backbone trained on a base dataset as a generic feature extractor, and the SOM was used to classify the obtained features from the target dataset. The SOM is trained with no label, then labeled with the few available annotated samples. We showed that we reach a good performance on the mini-ImageNet few shot classification benchmark with an unsupervised learning method. Furthermore, the SOM is suitable for hardware implementations based on a cellular neuromorphic architecture, which enables its application on the edge.

Chapter 5

Reentrant Self-Organizing Map (ReSOM): Proposed model

> Learning is about stabilizing
> pre-established synaptic combinations.
> It also means eliminating the others.
>
> *Jean-Pierre Changeux.*

5.1 Introduction

Cortical plasticity is one of the main features that enable our capability to learn and adapt in our environment. Indeed, as discussed in chapters 1 and 2, the cerebral cortex has the ability to self-organize itself through two distinct forms of plasticity: the structural plasticity that creates (sprouting) or cuts (pruning) synaptic connections between neurons, and the synaptic plasticity that modifies the synaptic connections strength. These mechanisms are very likely at the basis of an extremely interesting characteristic of the human brain development: the multimodal association. The brain uses spatio-temporal correlations between several modalities to structure the data and create sense from observations. Thus, in spite of the diversity of the sensory modalities, like sight, sound and touch, the brain arrives at the same concepts. Moreover, biological observations show that one modality can activate the internal representation of another modality when both are correlated. To model such a behavior, Gerald Edelman and Antonio Damasio proposed respectively the Reentry and the Convergence Divergence Zone frameworks where bi-directional neural communications can lead to both multimodal fusion (convergence) and inter-modal activation (divergence). Nevertheless, these frameworks do not provide a computational model at the neuron level, and only few works tackle this issue of brain-inspired multimodal association (Althaus and Mareschal, 2013) which is yet necessary for a complete representation of the environment. In this chapter, we propose the Reentrant Self-Organizing Map (ReSOM), a brain-inspired neural system based on the Reentry principles, using Self-Organizing Maps and Hebbian-like learning. We propose different computational methods for multimodal unsupervised learning and inference, with both divergence and convergence mechanisms. The divergence mechanism is used to label one modality based on the other, while the convergence mechanism is used to improve the overall accuracy of the system. Finally, we propose an extension of the IG introduced in chapter 2 to the multimodal framework.

5.2 Reentry and Convergence Divergence Zone (CDZ)

Brain's plasticity, also known as neuro-plasticity, is the key to humans capability to learn and adapt their behaviour. The plastic changes happen in neural pathways as a result of the multimodal sensori-motor interaction in the environment (Escobar-Juárez et al., 2016). But since most of the stimuli are processed by the brain in more than one sensory modality (Meyer and Damasio, 2009), how do the multimodal information *converge* in the brain?

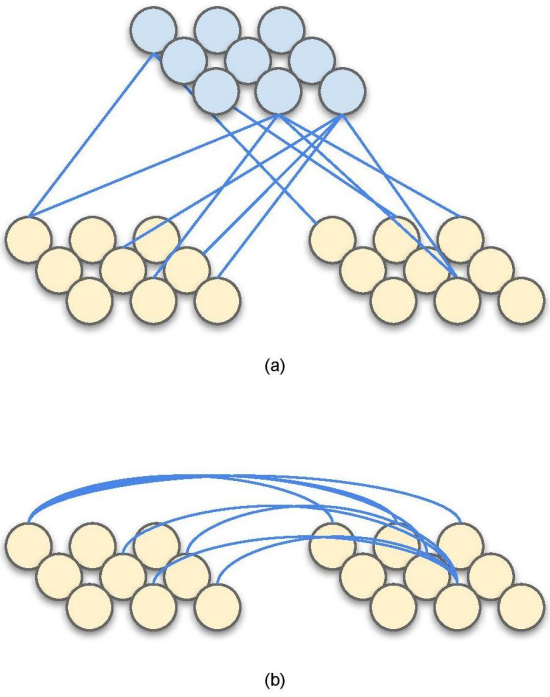

FIGURE 5.1: Schematic representation of the (a) CDZ and the (b) reentry frameworks. The CDZ paradigm (Damasio) implies hierarchical neurons that connect unimodal neurons, while the reentry paradigm (Edelman) states that unimodal neurons connect to each other through direct connections.

Indeed, we can recognize a dog by seeing its picture, hearing its bark or rubbing its fur. These features are different patterns of energy at our sensory organs (eyes, ears and skin) that are represented in specialized regions of the brain. However, we arrive at the same concept of the "dog" regardless of which sensory modality was used (Man et al., 2015). Furthermore, modalities can *diverge* and activate one another when they are correlated. Recent studies have demonstrated cross-modal activation amongst various sensory modalities, like reading words with auditory and olfactory meanings that evokes activity in auditory and olfactory cortices (Kiefer et al., 2008) (González et al., 2006), or trying to discriminate the orientation of a tactile grid pattern with eyes closed that induces activity in the visual cortex (Sathian and Zangaladze, 2002). Both mechanisms rely on the cerebral cortex as a substrate. "We see with the brain, not the eyes", Paul Bach-y-Rita quoted in (rita, 1972). But even though recent works have tried to study the human brain's ability to integrate inputs from multiple modalities (Calvert, 2001) (Kriegstein and Giraud, 2006), it is not clear how the different cortical areas connect and communicate with each other.

To answer this question, Edelman proposed in 1982 the Reentry (Edelman, 1982) (Edelman, 1993): the ongoing bidirectional exchange of signals linking two or more brain areas, one of the most important integrative mechanisms in vertebrate brains (Edelman, 1982). In a recent review (Edelman and Gally, 2013), Edelman defines reentry as a process which involves a localized population of excitatory neurons that simultaneously stimulates and is stimulated by another population, as shown in figure 5.1-b. It has been shown that reentrant neuronal circuits self-organize early during the embryonic development of vertebrate brains (Singer, 1990) (Shatz, 1992), and can give rise to patterns of activity with Winner-Take-All properties (Douglas and Martin, 2004) (Rutishauser and Douglas, 2009). When combined with appropriate mechanisms for synaptic plasticity, the mutual exchange of signals amongst neural networks in distributed cortical areas results in the spatio-temporal integration of patterns of neural network activity. It allows the brain to categorize sensory inputs, remember and manipulate mental constructs, and generate motor commands (Edelman and Gally, 2013). Thus, reentry would be the key to multimodal integration in the brain.

Damasio proposed another answer in 1989 with the Convergence Divergence Zone (CDZ) (Damasio, 1989) (Damasio and Damasio, 1994), another biologically plausible framework for multimodal association. In a nutshell, the CDZ theory states that particular cortical areas act as sets of pointers to other areas, with a hierarchical construction: the CDZ merges low level cortical areas with high level amodal constructs, which connects multiple cortical networks to each other and therefore solves the problem of multimodal integration. The CDZ convergence process integrates unimodal information into multimodal areas, while the CDZ divergence process propagates the multimodal information to the unimodal areas, as shown in figure 5.1-a. For example, when someone talks to us in person, we simultaneously hear the speaker's voice and see the speaker's lips move. As the visual movement and the sound co-occur, the CDZ would associate (convergence) the respective neural representations of the two events in early visual and auditory cortices into a higher cortical map. Then, when we only watch a specific lip movement without any sound, the activity pattern induced in the early visual cortices would trigger the CDZ and the CDZ would retro-activate (divergence) in early auditory cortices the representation of the sound that usually accompanied the lip movement (Meyer and Damasio, 2009).

The bidirectionality of the connections is therefore a fundamental characteristic of both reentry and CDZ frameworks, that are likewise in many aspects. Indeed, we find computational models of both paradigms in the literature. We review the most significant ones to our work in Section 5.3.

5.3 Models and applications

In this section, we make a chronological review of the recent works that explored brain-inspired multimodal learning for two main applications: sensori-motor mapping and multisensory classification.

5.3.1 Sensori-motor mapping

Lallee and Dominey (Lallee and Dominey, 2013) proposed one of the first models of brain-inspired multimodal association: the MultiModal Convergence Map (MMCM) that applies the SOM (Kohonen, 1990) to model the CDZ framework. A hierarchy

of SOMs is used to reduce the dimensionality of the input, using the coordinates of the most active unit of each unimodal map as input of the multimodal map. The MMCM was applied to encode the sensori-motor experience of a robot based on the language, vision and motor modalities. This "knowledge" was used in return to control the robot behaviour. The experiments were conducted in both a simulated and a real humanoid robot, the iCub (Metta et al., 2008). In a nutshell, the MMCM provides an implemented framework in which multiple modalities are represented in distinct and converging maps. Activation in one modality can be used to generate a mental image in the other modalities. Lallee and Dominey demonstrated how this can be used to increase the performance of the iCub in the recognition of its hand in different postures.

A quite similar approach is followed by Escobar-Juarez et al. (Escobar-Juárez et al., 2016) who proposed the Self-Organized Internal Models Architecture (SOIMA) that models the CDZ framework based on internal models, where sensory and motor information merge in a natural way and create a multimodal representation (Wolpert and Kawato, 1998). The work focused on the pair formed by inverse-forward models. The inverse model is a controller that generates the motor command (Mt) needed to achieve a desired sensory state (St+1) given a current sensory state (St), while the forward model is a predictor that predicts the sensory state entailed (St+1) by some action of the agent (Mt) given a current sensory state (St). The necessary property of bidirectionality is once again pointed out by the authors. SOIMA relies on two main learning mechanisms: the first one consists in SOMs that create clusters of unimodal information coming from the environment. The second one codes the internal models by means of connections between the first maps using Hebbian learning that generates sensory–motor patterns. As in (Lallee and Dominey, 2013), a hierarchy of SOMs is used such that the inputs to the top multimodal map are the coordinates of the winning neurons in the unimodal maps. The SOIMA architecture was successfully experimented on a saccadic control and hand-eye coordination tasks.

A different approach is used by Droniou et al. (Droniou, Ivaldi, and Sigaud, 2015) where the authors proposed an architecture based on DNNs, which is used by the iCub (Metta et al., 2008) to learn a task from multiple perceptual modalities: proprioception, vision and audition. The DNN is based on the auto-encoder paradigm for both reducing the dimensionality of data as in a standard auto-encoding approach and for clustering, adding a Softmax activation function (Memisevic et al., 2010) to make the compressed representation sparser and cluster the data. Globally, the system of Droniou et al. relates to the CDZ framework even if the actual purpose was not to provide a computational model for the theory. First, for a bi-modal task and given one modality alone, the network was able to infer a classification and a parametrization which can be used to reconstruct the missing modality. Second, the proposed network was able to exploit multimodal correlations to improve the representation of each modality alone.

Following the reentry paradigm, Zahra et al. (Zahra and Navarro-Alarcon, 2019) proposed the Varying Density Self-Organizing Map (VDSOM) for characterizing sensorimotor relations in robotic systems with bidirectional connections. The proposed method relies on self-organizing properties through SOMs and associative properties through Oja's learning (Oja, 1982) that enables it to autonomously obtain sensori-motor relations without any prior knowledge of either the motor (e.g. mechanical structure) or perceptual (e.g. sensor calibration) models. This solution relies on collecting data samples by motor babbling and is therefore suitable for various robotic manipulators without prior information about robot kinematics. Even though the paper (Zahra and Navarro-Alarcon, 2019) does not state so explicitly, the

VDSOM is closer to the reentry paradigm where direct bidirectional connections are learned amongst neurons.

5.3.2 Multisensory classification

Parisi et al. (Parisi et al., 2017) proposed a hierarchical architecture with Growing When Required (GWR) networks (Marsland, Shapiro, and Nehmzow, 2002) for learning human actions from audiovisual inputs. The neural architecture consists of a self-organizing hierarchy with four layers of GWR for the unsupervised processing of visual action features. The fourth layer of the network implements a semi-supervised algorithm where action–word mappings are developed. This is done by binding co-occurring audiovisual inputs using bidirectional inter-layer connectivity, and thus learning multimodal representations of actions. The direct bidirectional connections follow the reentry paradigm.

With the same paradigm, Jayaratne et al. (Jayaratne et al., 2018) proposed a multisensory neural architecture that consists of multiple self-organizing neural layers of Growing SOMs (GSOM) (Alahakoon, Halgamuge, and Srinivasan, 2000) for modelling the respective cortical areas of each sensory modality, and inter-sensory associative connections representing the co-occurrence probabilities of the modalities. Here again, there is no hierarchy in the bidirectional connections, thus referring to the reentry paradigm. The system was implemented in Apache Spark (Zaharia et al., 2016) to distribute the GSOM computing with respect to data, i.e. distribute data across a cluster of computers to process them in parallel, and thus improving its scalability to big datasets. The system's principle is to supplement the information on a single modality with the corresponding information on other modalities with the Tulips1 audio-visual dataset (Movellan, 1995) (not available), exploiting the co-occurrence relationship across the modalities for a better classification accuracy.

Using spike coding, Rathi and Roy (Rathi and Roy, 2018) proposed an STDP-based multimodal unsupervised learning for SNNs. The goal of this work was to learn the cross-modal connections between areas of single modality in SNNs to improve the recognition accuracy and make the system robust to noisy inputs. Each modality is represented by a specific SNN trained with its own data following the learning framework proposed in (Diehl and Cook, 2015). The SNN computation is distributed, but requires an all-to-all connectivty amongst neurons. As discussed and quantified in chapter 3, this full connectivity goes against the scalability of the network. In addition, the cross-modal connections between the two SNNs are trained along with the unimodal connections. The cross-modal connections are sparsely connected following the reentry paradigm and initialized with random weights. Afterwards, STDP is used to update these weights as both SNNs are presented with two inputs of the same class at the same time. The correlation between neurons of different modalities is captured in the cross-modal connections, which assist the network in making the right decision by increasing the spikes for the correct class. The proposed method was experimented with a written/spoken digit classification task, and the collaborative learning results in an accuracy improvement of 2% compared to the best unimodal accuracy. Furthermore, the multimodal approach makes the network noise tolerant. The work of Rathi and Roy (Rathi and Roy, 2018) is the first to train SNNs with multimodal inputs, and is the closest to our work. Hence, a detailed comparison is presented in section 6.4.2.

Finally, Cholet et al. (Cholet, Paugam-Moisy, and Regis, 2019) proposed a modular architecture for multimodal fusion using Bidirectional Associative Memories (BAMs), which were initially proposed by Kosko (Kosko, 1988) as an adaptation of

the Hopfield network (Hopfield, 1982) for hetero-association. The BAMs is composed of two fully and bidirectionally connected layers. The proposed architecture can be summarised in three stages: unimodal data are first processed by as many independent prototype-based Incremental Neural Networks (INNs) (Azcarraga and Giacometti, 1991) as the number of modalities to be combined. The second stage consists of multiple BAMs that achieve the fusion of modalities by learning pairs of unimodal prototypes towards the integrative layer which builds an abstract representation. Finally, the third stage is an INN that performs supervised classification. Even though we can see a form of hierarchy in the third stage with the INN that takes the BAMs as input for classification, the multimodal association itself is made with direct BAMs between uni-modal representations, thus following the reentry paradigm.

5.3.3 Summary

TABLE 5.1: Models and applications of brain-inspired multimodal learning.

Application	Work	Paradigm	Learning	Computing
Sensori-motor mapping	(Lallee and Dominey, 2013)	CDZ	Unsupervised	Centralized
	(Droniou, Ivaldi, and Sigaud, 2015)	CDZ	Unsupervised	Centralized
	(Escobar-Juárez et al., 2016)	CDZ	Unsupervised	Centralized
	(Zahra and Navarro-Alarcon, 2019)	Reentry	Unsupervised	Centralized
Multisensory classification	(Parisi et al., 2017)	Reentry	Semi-supervised	Centralized
	(Jayaratne et al., 2018)	Reentry	Semi-supervised	Distributed[1]
	(Rathi and Roy, 2018)	Reentry	Unsupervised	Centralized **
	(Cholet, Paugam-Moisy, and Regis, 2019)	Reentry *	Supervised	Centralized
	(Khacef et al., 2020) [Our work]	Reentry	Unsupervised	Distributed[2]

[1] data level.
[2] system level.
* with an extra layer for classification.
** learning is distributed but inference for classification is centralized.

Overall, the rentry and CDZ frameworks share two key aspects: the multimodal associative learning based on the temporal co-occurrence of the modalities, and the bidirectionality of the associative connections. We summarize the most relevant papers to our work in Table 5.1, where we classify each paper with respect to the application, the brain-inspired paradigm, the learning type and the computing nature. We notice that sensori-mapping is based on unsupervised learning, which is natural as no label is necessary to map two modalities together. However, classification is based on either supervised or semi-supervised learning, as mapping multisensory modalities is not sufficient: we need to know the corresponding class to each activation pattern. That's why we proposed in chapter 3 a labeling method based on a small labeled subset, so that we do not use any label in the learning process. The same approach is used in (Rathi and Roy, 2018), but the authors rely on the complete labeled dataset, as further discussed in section 6.4.2. Finally, all previous works rely on the centralized Von Neumann computing paradigm, except (Jayaratne et al., 2018) that attempts a partially distributed computing with respect to data, i.e. using the MapReduce computing paradigm to speed up computations. It is based on Apache Spark (Gu and Li, 2013), mainly used for cloud computing. Also, STDP learning in (Rathi and Roy, 2018) is distributed, but the inference for classification

requires a central unit, as discussed in section 6.4.2. We propose a fully distributed computing on the edge with respect to the system, i.e. the neurons computing itself to improve the SOMs scalability for hardware implementation thanks to the IG presented in chapter 2.

Consequently, we chose to follow the reentry paradigm where multimodal processing is distributed in all cortical maps without dedicated associative maps for two reasons. First, from the brain-inspired computing perspective, more biological evidences tend to confirm the hypothesis of reentry as reviewed by (Barth et al., 1995), (Allman, Keniston, and Meredith, 2009) and (Lefort, Boniface, and Girau, 2013). Indeed, biological observations highlight a multimodal processing in the whole cortex including sensory areas (Calvert, Spence, and Stein, 2004) which contain multimodal neurons that are activated by multimodal stimuli (Barth et al., 1995) (Bizley and King, 2008). Moreover, it has been shown that there are direct connections between sensory cortices (Cappe, Rouiller, and Barone, 2009) (Schroeder and Foxe, 2005), and neural activities in one sensory area may be influenced by stimuli from other modalities (Allman, Keniston, and Meredith, 2009) (Dehner et al., 2004). Second, from a pragmatical and functional perspective, the reentry paradigm fits better to our cellular architecture based on the IG, and thus increases the scalability and fault tolerance thanks to the fully distributed processing (Lefort, Boniface, and Girau, 2013). Nevertheless, we keep the *convergence* and *divergence* terminology to distinguish between, respectively, the integration of two modalities and the activation of one modality based on the other.

5.4 Reentrant Self-Organizing Map (ReSOM)

In this section, we introduce the Reentrant Self-Organizing Map (ReSOM) neural system described in figure 5.2, a new model composed of two or more SOMs with afferent connections each and lateral connections amongst them, following the reentry paradigm. The ReSOM will be used for learning multimodal associations, labeling one modality based on the other and converging the two modalities with *cooperation* and *competition* for a better classification accuracy.

The initial convergence zone model was proposed by Moll and Miikkulainen in 1997 (Miikkulainen and Moll, 1997), but it lacked the self-organizing and topographical property inherent to cortical maps (Lallee and Dominey, 2013). Hence, we use SOMs and Hebbian-like learning in two times to perform multimodal learning as illustrated as a flowchart in figure 5.3: first, unimodal representations are obtained with SOMs and, second, multimodal representations develop through the association of unimodal maps via bidirectional synapses that can be seen as BAMs. The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence (Fiebelkorn, Foxe, and Molholm, 2010), and follow the reentry theory (Edelman and Gally, 2013).

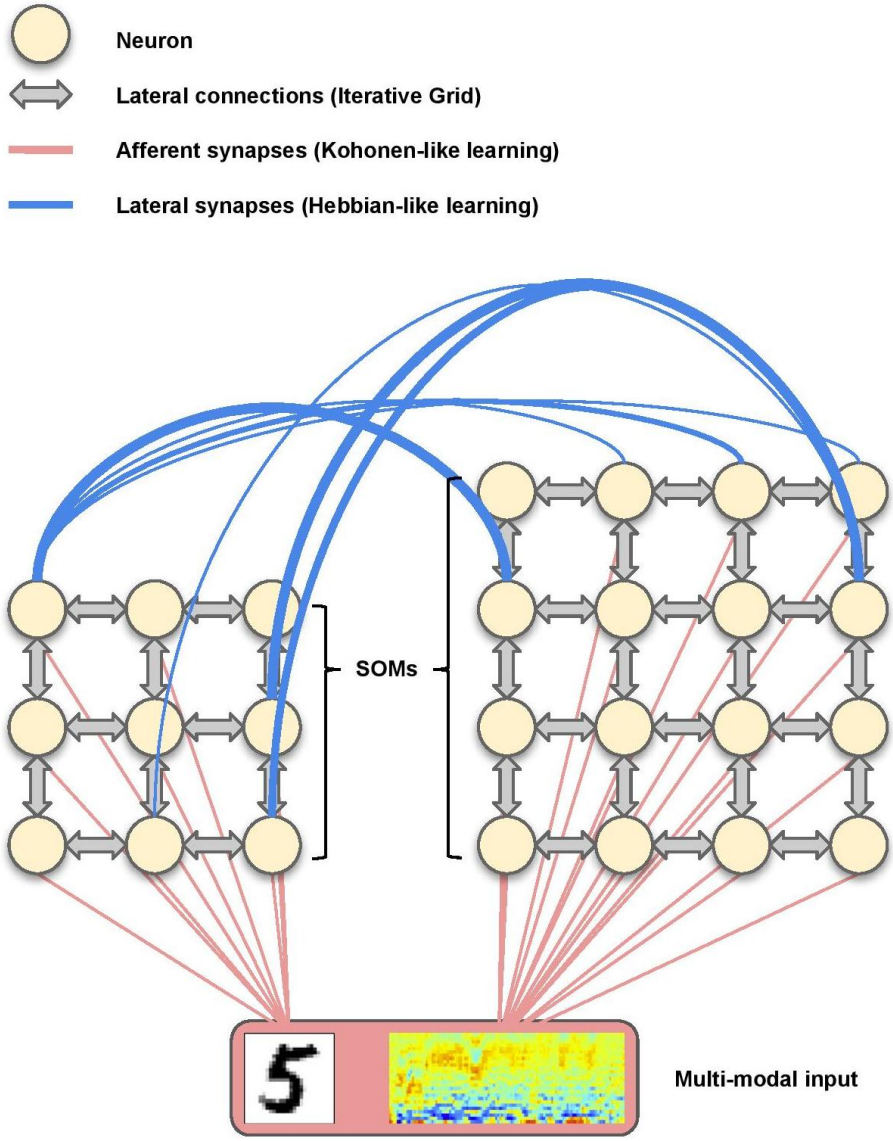

FIGURE 5.2: Schematic representation of the proposed ReSOM for multimodal association. For clarity, the lateral connections of only two neurons from each map are represented.

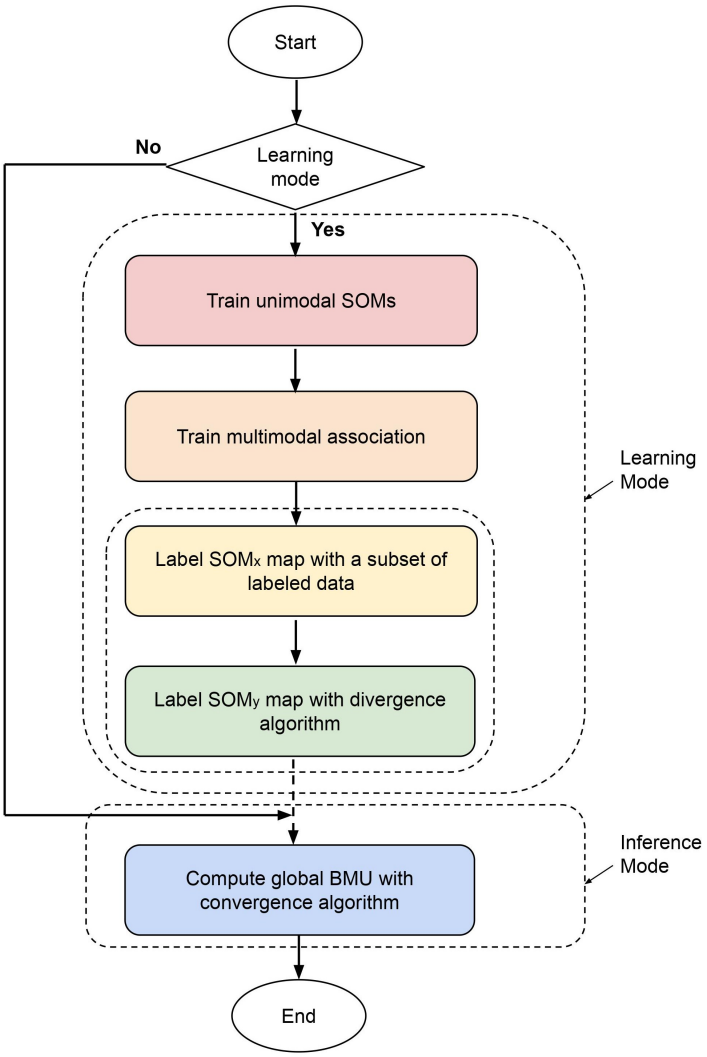

FIGURE 5.3: Flowchart: Multimodal unsupervised learning overview.

5.4.1 ReSOM multimodal association learning

Brain's plasticity can be divided into two distinct forms of plasticity: the (1) structural plasticity that changes the neurons connectivity by sprouting (creating) or pruning (deleting) synaptic connections, and (2) the synaptic plasticity that modifies (increasing or decreasing) the existing synapses strength (Fauth and Tetzlaff, 2016). We explore both mechanisms for multimodal association through Hebbian-like learning, as illustrated as a flowchart in figure 5.4.

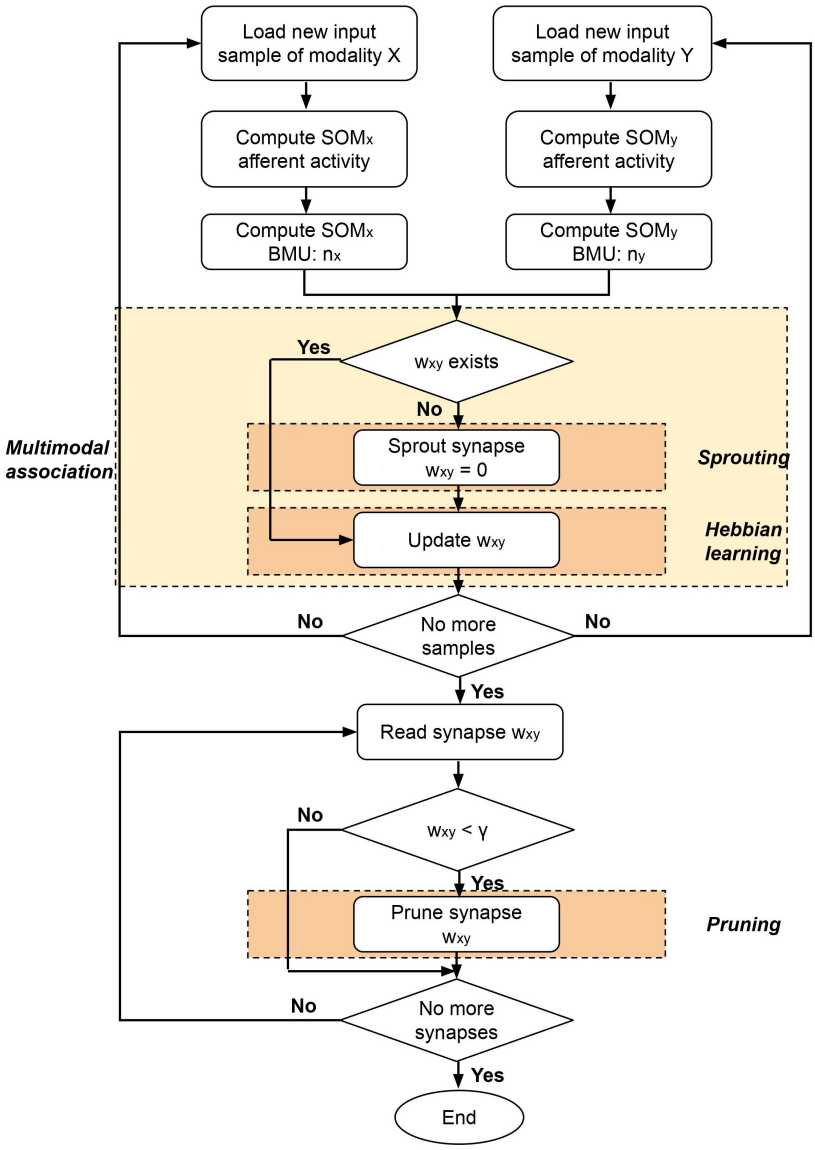

FIGURE 5.4: Flowchart: Multimodal association learning.

The original Hebbian learning principle (Hebb, 1949) proposed by Hebb in 1949 states that "when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." In other words, any two neurons that are repeatedly active at the same time will tend to become "associated" so that activity in one facilitates activity in the other. The learning rule is expressed by equation 5.1. The Hebbian learning exhibits

Algorithm 8: ReSOM multimodal association learning

Learn neurons afferent weights for SOM_x and SOM_y corresponding to modalities x and y respectively.
for every multimodal input vectors v_x and v_y **do**
  Compute the SOM_x and SOM_y neurons activities.
  Compute the unimodal BMUs n_x and n_y with activities a_x and a_y respectively.
  if Lateral connection w_{xy} between n_x and n_y does not exist **then**
    Sprout (create) the connection $w_{xy} = 0$.
  else
    Update lateral connection w_{xy}:
    if Hebb's learning **then**

$$w_{xy} = w_{xy} + \eta \times a_x \times a_y \tag{5.1}$$

    else if Oja's learning **then**

$$w_{xy} = w_{xy} + \eta \times (a_x \times a_y - w_{xy} \times a_y^2) \tag{5.2}$$

    end if
  end if
end for
for every neuron x in the SOM_x network **do**
  Sort the lateral synapses w_{xy} and deduce the pruning threshold γ.
  for every lateral synapse w_{xy} **do**
    if $w_{xy} < \gamma$ **then**
      Prune (delete) the connection w_{xy}.
    end if
  end for
end for

several interesting computational features like the storage of patterns, pattern completion, or temporal storage (Hopfield, 1982) (Bauer, 2013). Although Hebb could not verify his theory himself, some evidence for Hebb's rule was later found in the hippocampus in the form of LTP (Ranganathan and Kira, 2003).

However, Hebb's rule is limited in terms of stability for online learning, as synaptic weights tend to infinity with a positive learning rate. This could be resolved by normalizing each weight over the sum of all the corresponding neuron weights, which guarantees the sum of each neuron weights to be equal to 1. The effects of weights normalization are explained in (Goodhill and Barrow, 1994). However, this solution breaks up with the locality of the synaptic learning rule, and that is not biologically plausible. In 1982, Oja proposed a Hebbian-like rule (Oja, 1982) that adds a "forgetting" parameter, and solves the stability problem with a form of local multiplicative normalization for the neurons weights, as expressed in equation 5.2. In addition, It has been shown that Oja's learning performs an *on-line* Principal Component Analysis (PCA) of the data in the neural network (Fyfe, 1997), which is a very interesting property in the context of unsupervised learning since the PCA finds the best linear compression of data by finding the linear basis of the dataset that minimizes the Mean Squared Error (MSE) between the compressed and uncompressed data (Ranganathan and Kira, 2003).

Nevertheless, Hebb's and Oja's rules were both used in recent works with good results, respectively in (Escobar-Juárez et al., 2016) and (Zahra and Navarro-Alarcon, 2019). Hence, we applied and compared both rules. The proposed reSOM multimodal association model is detailed in algorithm 8, where η is a learning rate that we fix to 1 in our experiments, and γ is deduced according to the number or the percentage of synapses to prune, as discussed in section 6.2. The neurons activities in the line 3 of algorithm 8 are calculated following equations 2.1 and 2.2.

5.4.2 ReSOM divergence for labeling

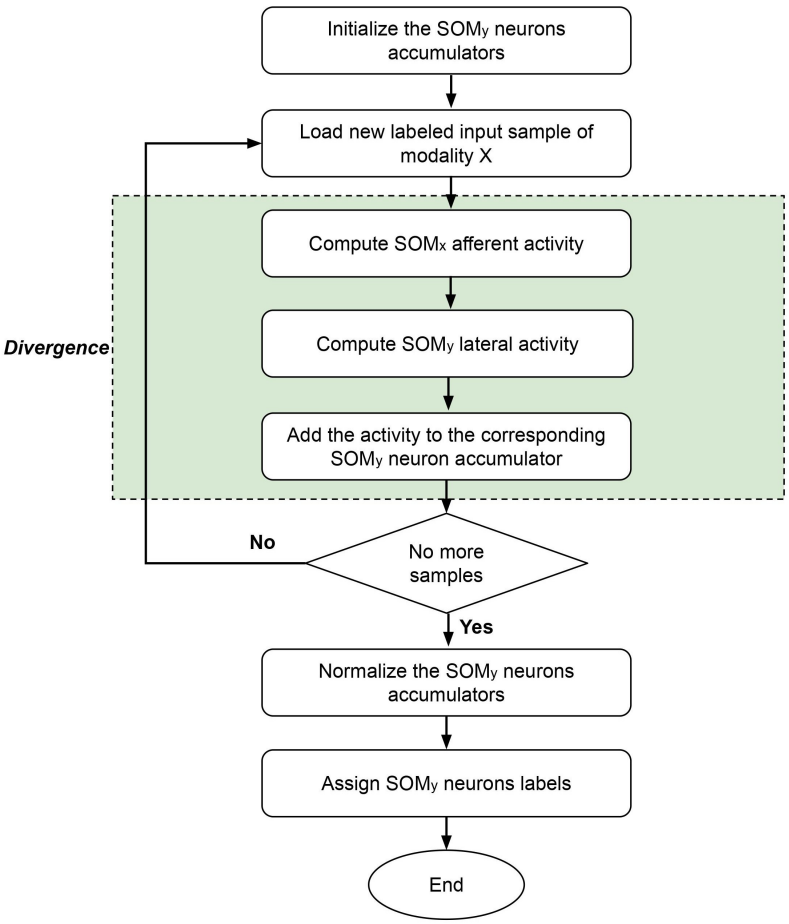

FIGURE 5.5: Flowchart: Divergence mechanism for labeling.

As explained in chapter 3, neurons labeling is based on a labeled subset from the training database. We tried use the fewest labeled samples while keeping the best accuracy, which was 1% of the training set for MNIST. However, we will see in section 6.2 that depending on the database, we sometimes need a considerable number of labeled samples, up to 10% of the training set. In this section, we propose an original method based on the divergence mechanism of the multimodal association, as illustrated as a flowchart in figure 5.5: for two modalities x and y, since we can activate one modality based on the other, we propose to label the SOM_y neurons from the activity and the labels induced from the SOM_x neurons, which are based on the labeling subset of modality x. Therefore, we only need one labeled subset of a single modality which needs the fewest labels to label both modalities,

Algorithm 9: ReSOM divergence for labeling

Initialize $class_{act}$ as a two-dimentionnal array of accumulators: the first dimension is the neurons and the second dimension is the classes.
for every input vector v_x of the x-modality labeling set with label l **do**
  for every neuron x in the SOM_x network **do**
    Compute the afferent activity a_x:

$$a_x = e^{-\frac{||v_x - w_x||}{\alpha}} \tag{5.3}$$

  end for
  for every neuron y in the SOM_y network **do**
    Compute the divergent activity a_y from the SOM_x:

$$a_y = \max_{x=0}^{n-1} \left(w_{xy} \times a_x \right) \tag{5.4}$$

    Add the normalized activity with respect to the max activity to the corresponding accumulator:

$$class_{act}[y][l] += a_y \tag{5.5}$$

  end for
end for
Normalize the accumulators $class_{act}$ with respect to the number of samples per class.
for every neuron y in the SOM_y network **do**
  Assign the neuron label $neuron_{lab}$:

$$neuron_{lab} = argmax(class_{act}[y]) \tag{5.6}$$

end for

taking profit of the bidirectional aspect of reentry. A good analogy to biological observations would be the retro-activation of the auditory cortical areas from the visual cortex, if we take the example of written/spoken digits presented in section 6.3. It is also similar to how infants respond to sound symbolism by associating shapes with sounds (Asano et al., 2015). The proposed ReSOM divergence method for labeling is detailed in algorithm 9.

5.4.3 ReSOM convergence for classification

Once the multimodal learning is done and all neurons from both SOMs are labeled, we need to converge the information of the two modalities to achieve a better representation of the multisensory input, as illustrated as a flowchart in figure 5.6. Since we use the reentry paradigm, there is no hierarchy in the processing, and the neurons computing is fully distributed based on the IG. We propose an original cellular convergence method in the ReSOM, as detailed in algorithm 10. We can summarize it in three main steps:

- First, there is an *independent* activity computation (Equation 5.7) where each neuron of the two SOMs computes its activity based on the afferent activity

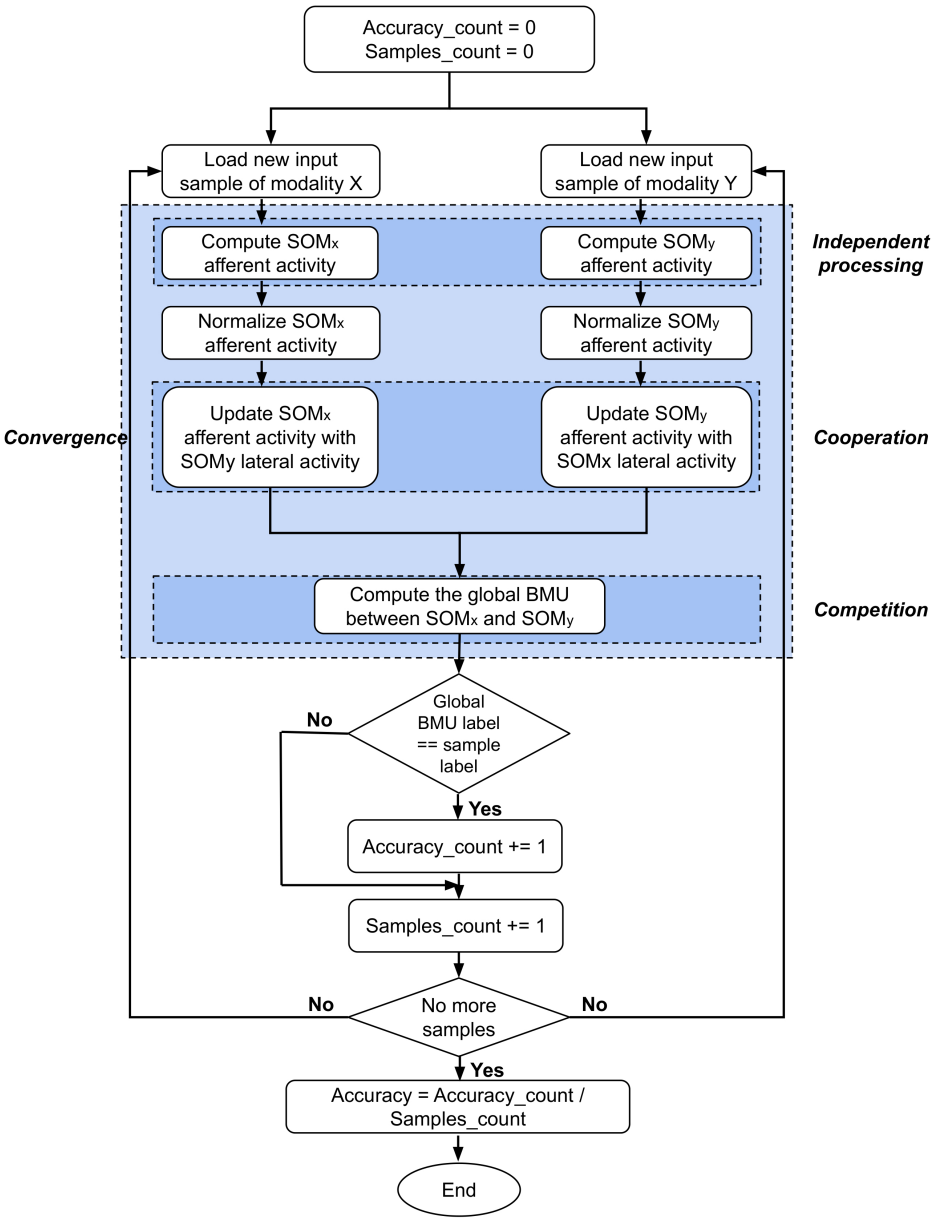

FIGURE 5.6: Flowchart: Convergence mechanism for classification.

from the input.

- Second, there is a *cooperation* amongst neurons from different modalities (Equations 5.8 and 5.9) where each neuron updates its afferent activity via a multiplication with the lateral activity from the neurons of the other modality.
- Third and finally, there is a global *competition* amongst all neurons (line 19 in Algorithm **??**) where they all compete to elect a winner, i.e. a global BMU with respect to the two SOMs.

We explore different variants of the proposed convergence method regarding two aspects. First, both afferent and lateral activities can be taken as raw values or normalized values. We used min-max normalization that is therefore done with respect to the BMU and the Worst Matching Unit (WMU) activities. Second, the

Algorithm 10: ReSOM convergence for classification

for every multimodal input vectors v_x and v_y **do**
  Do in parallel every following step inter-changing modality x with modality y and vice-versa:
  Compute the afferent activities a_x and a_y:
  for every neuron x in the SOM_x network **do**
    Compute the afferent activity a_x:

$$a_x = e^{-\frac{||v_x - w_x||}{\beta}} \tag{5.7}$$

  end for
  Normalize (min-max) the afferent activities a_x and a_y.
  Update the afferent activities a_x and a_y with the lateral activities based on the associative synapses weights w_{xy}:
  if Update with max_{update} **then**
    for every neuron x in the SOM_x network connected to n neurons from the SOM_y network **do**

$$a_x = a_x \times \max_{x=0}^{n-1} \left(w_{xy} \times a_y \right) \tag{5.8}$$

    end for
  else if Update with sum_{update} **then**

    for every neuron x in the SOM_x network connected to n neurons from the SOM_y network **do**

$$a_x = a_x \times \frac{\sum_{x=0}^{n-1} \left(w_{xy} \times a_y \right)}{n} \tag{5.9}$$

    end for
  end if
  Compute the global BMU with the maximum activity between the SOM_x and the SOM_y.
end for

afferent activities update could be done for all neurons or only the BMUs. In the second case, the global BMU cannot be another neuron but one of the two local BMUs, and if there is a normalization then it is only done for lateral activities (otherwise, the BMUs activities would be 1, and the lateral map activity would be the only relevant one). The results of our comparative study are presented and discussed in section 6.3.

5.5 Discussion: Hardware support for multimodal association

For the multimodal association learning in algorithm 8, the local BMU in each of the two SOMs needs both the activity and the position of the local BMU of the other SOM to perform the Hebbian-like learning in the corresponding lateral synapse. This communication problem has not been experimented in this work. However, this suppose a simple communication mechanism between the two maps that would

implemented in two FPGAs where only the BMUs of each map send a message to each other in a bidirectional way. The message could go through the routers of the IG thanks to an XY-protocol to reach an inter-map communication port in order to avoid the multiplication of communication wires. It is to note that in this work, we follow the same approach as (Escobar-Juárez et al., 2016) since we only reinforce the synaptic connections between the two unimodal BMUs for each sample. The other approach is the learning with all neurons, but it would create a bottleneck in the inter-map communication and thus drastically increase the learning time.

For the divergence and convergence methods in algorithm 9 and algorithm 10 respectively, the local BMU in each of the two SOMs needs the activity of all the connected neurons from the other SOM after pruning, i.e. around 20 connections per neuron in a 10×10 neurons map, as further detailed in chapter 6. Because the number of remaining synapses is statistically bounded to 20%, the number of communications remains low in front of the number of neurons. Here again, we did not experiment on this communication mechanism but the same communication support could be used. Each BMU can send a request that contains a list of connected neurons. This request can be transmitted to the other map through the IG routers to an inter-map communication channel. Once on the other map, the message could be broadcasted to each neuron using again the routers of the IG. Only the requested neurons send back their activity coupled to their position in the BMU request. This simple mechanism supposes a low amount of communications thanks to the pruning that has been done previously. This inter-map communication can be possible if the IG routers support XY or equivalent routing techniques and broadcast in addition to the one of the propagation wave.

At this point of our work, each neuron is implemented in a NPU based on a previous implementation (Fiack, Rodriguez, and Miramond, 2015) with all the computing and memory resources, amongst which a list of the lateral synaptic weights with indexes to identify existing connections from pruned (or not sprouted) connections. Therefore, each lateral weight is saved by two neurons in the case of two SOMs. This aspect has to be further studied, as discussed in chapter 7.

5.6 Conclusion

We proposed in this chapter a new brain-inspired computational model for multimodal unsupervised learning called the ReSOM. Based on the reentry paradigm proposed by Edelman, it is a generic model regardless of the number of maps (as further discussed in chapter 7) and the number of neurons per map. The ReSOM learns unimodal representations with Kohonen-based SOMs, then creates and reinforces the multimodal association via sprouting, Hebbian-like learning and then pruning. It enables both structural and synaptic plasticities that are the core of neural self-organization. We exploited both convergence and divergence that are highlighted by Damasio thanks to the bi-directional property of the multimodal representation in a classification task: the divergence mechanism is used to label one modality based on the other, and the convergence is used to introduce cooperation and competition between the modalities and reach a better accuracy than the best of the two unimodal accuracies. The experimental results on three datasets are presented in chapter 6.

Chapter 6

ReSOM performance in multimodal unsupervised learning

> The mind is embodied, in the full sense of the term, not just embrained.
>
> *Antonio Damasio.*

6.1 Introduction

In this chapter, we experiment the ReSOM model with three multimodal datasets. First, we use a constructed written/spoken digits database with raw data. Second, we use a DVS/EMG hand gestures database with extracted features. Third and finally, we use a constructed DVS hand gestures/spoken digits database in which we associate each hand gesture to a spoken digit that serves as a *label* as we can observe it in infants development. We compare the ReSOM computational methods proposed in chapter 5, then we quantify the gain of both convergence and divergence mechanisms in the three multimodal classification tasks. Finally, we discuss the gain of the so-called *hardware* plasticity induced by the ReSOM model, where the system's topology is not fixed by the user but learned along the system's experience through self-organization (Rodriguez, Fiack, and Miramond, 2013).

6.2 SOM unimodal classification results

6.2.1 Multimodal databases

The first database is the constructed written/spoken digits database (Khacef, Rodriguez, and Miramond, 2019) with the visual MNIST handwritten digits (LeCun and Cortes, 1998) and the spoken version that we call Spoken MNIST (S-MNIST) (Warden, 2018). Following the original MNIST structure, the dataset consists of $70,000$ samples ($60,000$ for training and $10,000$ for test). For this first experimentation, we do not rely on the feature extraction methods presented in chapter 4 for MNIST classification. Instead, the ReSOM learns on the raw data in order to quantify the impact on the classifier performance only. To validate our results, we experiment our model on a second database that was originally recorded with multiple sensors: the DVS/EMG hand gestures database (Ceolini et al., 2019). The dataset consists of $6,750$ samples ($5,400$ for training and $1,350$ for test) of muscle activities via EletroMyoGraphy (EMG) signals recorded by a Myo armband (Thalmic Labs Inc) from the forearm, and video recordings from a Dynamic Vision Sensor (DVS).

The details on both databases are presented in appendix B. The most important hypothesis that we want to confirm through this work is that the multimodal association of two modalities leads to a better accuracy than the best of the two modalities alone.

In this section, we present the results from our experiments with each modality alone, summarized in table 6.1. All unimodal trainings were performed with the KSOM over 10 epochs with the same hyper-parameters used in chapter 3: $\epsilon_i = 1.0$, $\epsilon_f = 0.01$, $\sigma_i = 5.0$ and $\sigma_f = 0.01$. All the results presented in this work have been averaged over a minimum of 10 runs, with shuffled datasets and randomly initialized neurons afferent weights.

TABLE 6.1: ReSOM classification accuracies and convergence/divergence gains for multimodal digits and hand gestures.

Database		Digits		Hand gestures	
		MNIST	**S-MNIST**	**DVS**	**EMG**
SOMs	Dimensions	784	507	972	192
	Neurons	100	256	256	256
	Labeled data (%)	1	10	10	10
	Accuracy (%) $_{\alpha}$	87.04 $_{1.0}$	75.14 $_{0.1}$	70.06 $_{2.0}$	66.89 $_{1.0}$
ReSOM divergence	Labeled data (%)	1	0	10	0
	Gain (%)	/	+0.76	/	-1.33
	Accuracy (%)	/	75.90	/	65.56
ReSOM convergence	Gain (%)	**+8.03**	+19.17	**+5.67**	+10.17
	Accuracy (%)	**95.07**		**75.73**	

6.2.2 Written/spoken digits

MNIST classification with the KSOM was already performed in chapter 3, achieving around 87% of classification accuracy using 1% of labeled images from the training dataset for the neurons labeling. The only difference is the computation of the α in equation 2.2 for the labeling process, for which an approximation method was proposed in algorithm 6. In this chapter, we consider it as a simple hyper-parameter. We therefore calculate the best value off-line with a grid search since we do not want to include any centralized computation, and because we can find a closer value to the optimum, as summarized in table 6.1. The same procedure with the same hyper-parameters defined above is applied for each of the remaining unimodal classifications. Finally, we obtain 87.04% $\pm$ 0.64 of accuracy. Figure 6.1-a shows the confusion matrix that highlights the most frequent mis-classifications between the digits whose representations are close: 23.12% of the digits 4 are classified as 9 and 12.69% of the digits 9 are classified as a 4. We find the same mistakes with a lower percentage between the digits 3, 5 and 8, because of their proximity in the 784-dimensional vector space. That's what we aim to compensate by adding the auditory modality.

Even though we achieved state of the art performance with the same number of neurons (100) and only 1% of labeled samples for neurons labeling, the obtained accuracy of 87.36% on MNIST is not comparable to supervised DNNs, and only two approaches have been used in the literature to bridge the gap: either use a huge number of neurons (6400 neurons in (Diehl and Cook, 2015)) with exponential increase in size for linear increase in accuracy (Rathi and Roy, 2018) which is not scalable for complex databases, or use unsupervised feature extraction followed by a supervised classifier such as a SVM in (Kheradpisheh et al., 2018) which relies on the complete

labeled dataset. We propose the multimodal association as a way to bridge the gap while keeping a small number of neurons and an unsupervised learning method from end to end. For this purpose, we associate MNIST to an auditory modality: Spoken-MNIST (S-MNIST).

We extracted S-MNIST from Google Speech Commands (GSC) (Warden, 2018), an audio dataset of spoken words that was proposed in 2018 to train and evaluate keyword spotting systems. It was therefore captured in real-world environments though phone or laptop microphones. The dataset consists of $105,829$ utterances of 35 words, amongst which $38,908$ utterances ($34,801$ for training and $4,107$ for test) of the 10 digits from 0 to 9. We constructed S-MNIST associating written and spoken digits of the same class, respecting the initial partitioning in (LeCun and Cortes, 1998) and (Warden, 2018) for the training and test databases. Since we have less samples in S-MNIST than in MNIST, we duplicated some random spoken digits to match the number of written digits and have a multimodal-MNIST database of $70,000$ samples.

A pre-processing was done via the extraction of the Mel Frequency Cepstral Coefficients (MFCC) with a framing window size of $50ms$ and frame shift size of $25ms$. Since the speech samples are approximately $1s$ long, we end up with 39 time slots. For each one, we extract 12 MFCC coefficients with an additional energy coefficient. Thus, we have a final vector of $39 \times 13 = 507$ dimensions. Standardization and normalization were applied on the MFCC features. The SOM classification accuracy is $75.14\% \pm 0.57$. The confusion matrix in Figure 6.1 shows that the confusion between the digits 4 and 9 is almost zero, which strengthens our hypothesis that the auditory modality can complement the visual modality for a better overall accuracy.

6.2.3 DVS/EMG hand gestures

The framework was applied for the hand gestures recognition task with five hand gestures: *Pinky* (P), *Elle* (E), *Yo* (Y), *Index* (I) and *Thumb* (T). In order to use the DVS events with the ReSOM, we converted the stream of events into frames (event-based computing is out of the scope of this work). The frames were generated by counting the events occurring in a fixed time window for each of the pixels separately, followed by a min-max normalization to get gray scale frames. The time window was fixed to $200ms$ so that the DVS frames can be synchronized with the EMG signal, as further detailed in Reference (Ceolini et al., 2019b). The event frames obtained from the DVS camera have a resolution of 128×128 pixels. Since the region with the hand gestures does not fill the full frame, we extract a 60×60 pixels patch that allows us to significantly decrease the amount of computation needed during learning and inference.

Even though unimodal classification accuracies are not the first goal in this chapter, we need to reach a *satisfactory* performance before going to the multimodal association. Since the dataset is small and the DVS frames are of high complexity with a lot of noise from the data acquisition, we either have to significantly increase the number of neurons for the SOM or use feature extraction. We decided to use the second method with a CNN-based feature extraction as described in chapter 4. We use supervised feature extraction to demonstrate that the ReSOM multimodal association is possible using features. Future works will focus on the transition to unsupervised feature extraction with more complex datasets than MNIST. Thus, we used a supervised CNN feature extractor with the LeNet-5 topology (Lecun et al., 1998) except for the last convolution layer which has only 12 filters instead of 120. The CNN topology is therefore $60 \times 60 \times 1 - 6c3 - p2 - 16c3 - p2 - 12c5 - 84d - 5d$.

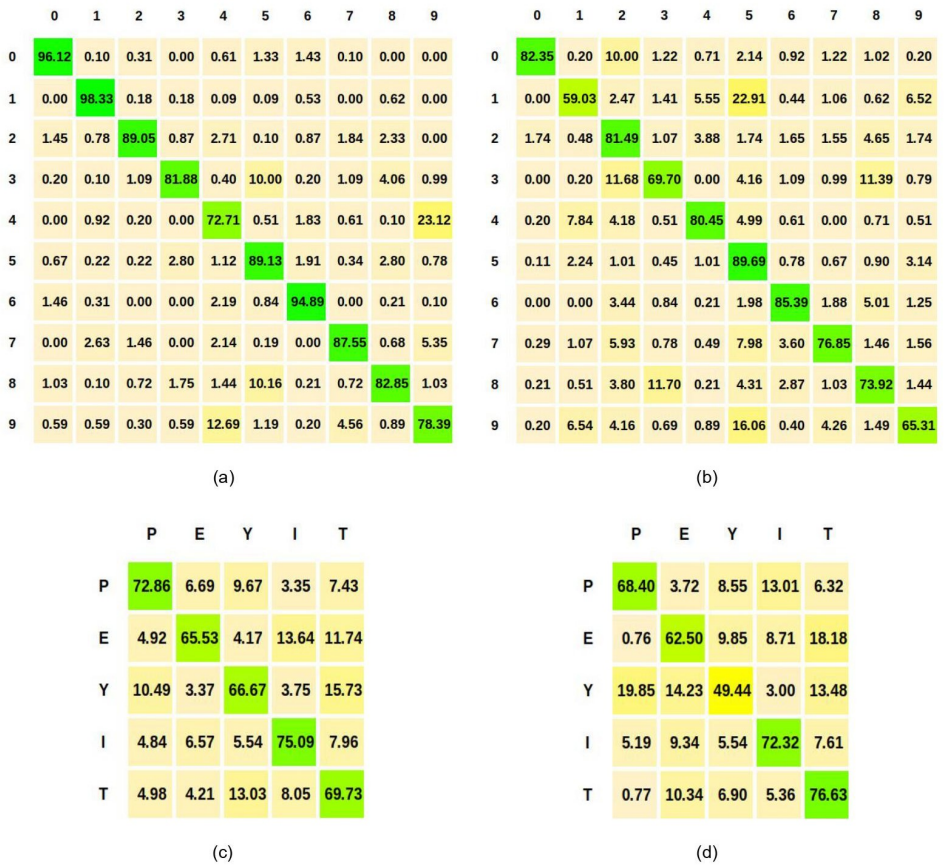

FIGURE 6.1: KSOM learning confusion matrix: (a) MNIST (b) S-MNIST divergence; (c) DVS hand gestures; (d) EMG hand gestures divergence.

Hence, we extract CNN-based features of 972 dimensions that we standardize and normalize. We obtain an accuracy of $70.06\% \pm 1.15$.

For the EMG signal, we selected two time domain features that are commonly used in the literature (Phinyomark, N Khushaba, and Scheme, 2018): the Mean Absolute Value (MAV) and the Root Mean Square (RMS) which are calculated over the same window of length $20ms$, as detailed in appendix B. With the same strategy as for DVS frames, we use a supervised CNN for feature extraction with the LeNet-5 topology (Lecun et al., 1998) without pooling. The CNN topology is thus $16 \times 1 - 6c3 - 16c3 - 120d - 84d - 5d$. Hence, we extract CNN-based features of 192 dimensions that we standardize and normalize. The SOM reaches a classification accuracy of $66.89\% \pm 0.84$.

6.3 ReSOM multimodal classification results

In this section, we present the results from our experiments with the multimodal association convergence and divergence mechanisms. After inter-SOM sprouting (figure 6.2), training and pruning (figure 6.3), we move to the inference for two different tasks: (1) labeling one SOM based on the activity of the other (*divergence*), and (2) classifying multimodal data with cooperation and competition between the two SOMs (*convergence*).

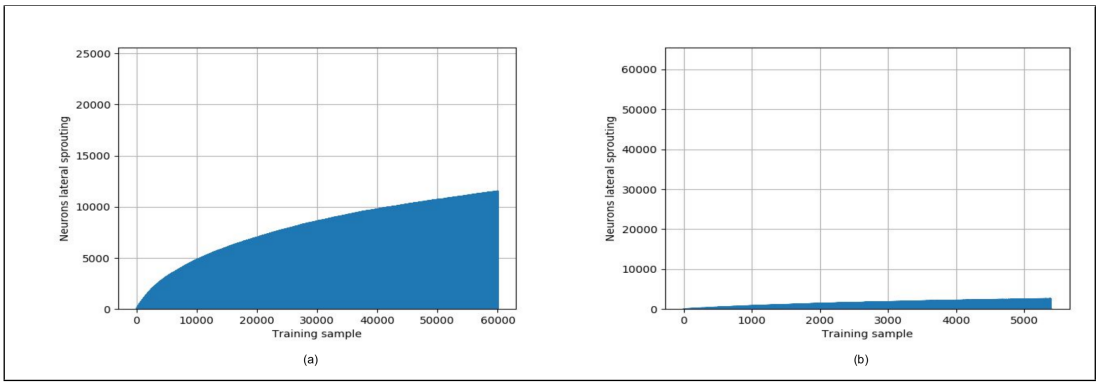

FIGURE 6.2: SOMs lateral sprouting in the multimodal association process: (a) Written/Spoken digits maps; (b) DVS/EMG hand gestures maps.

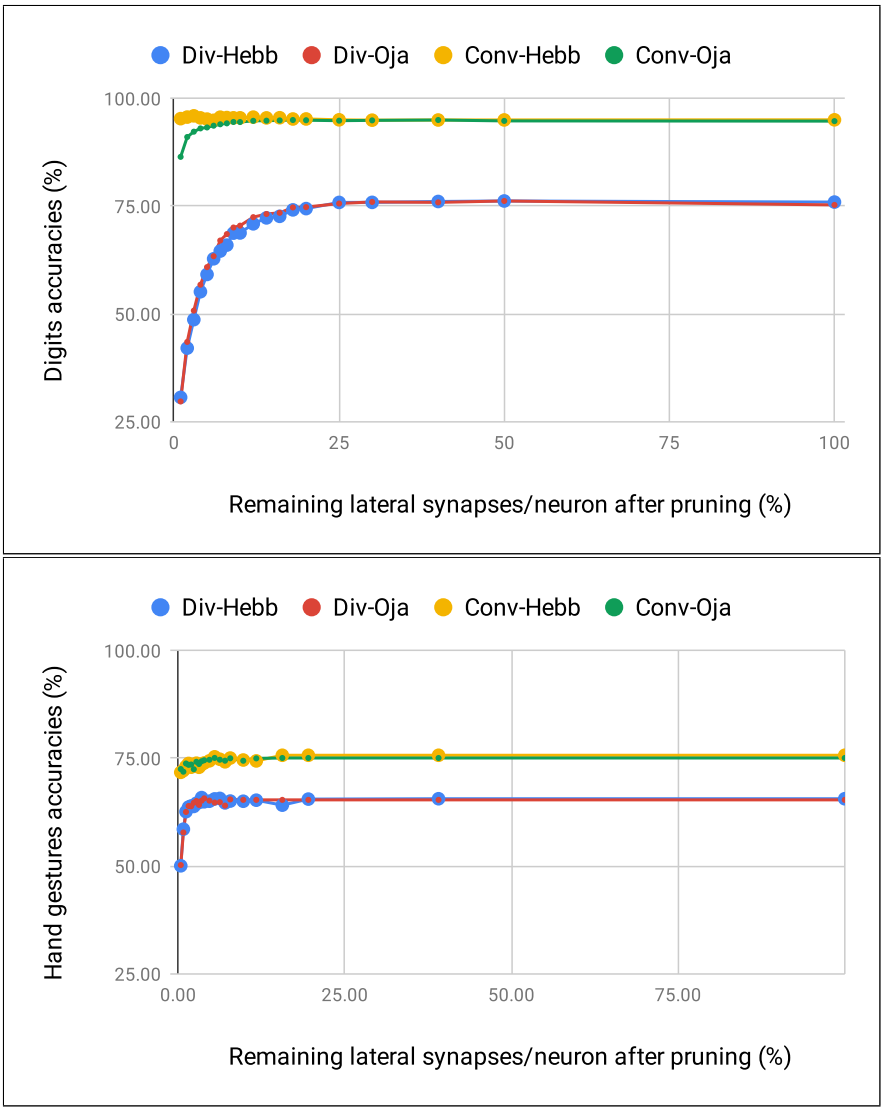

FIGURE 6.3: Divergence and convergence classification accuracies vs. the remaining percentage of lateral synapses after pruning: (top) Written/Spoken digits maps; (bottom) DVS/EMG hand gestures maps.

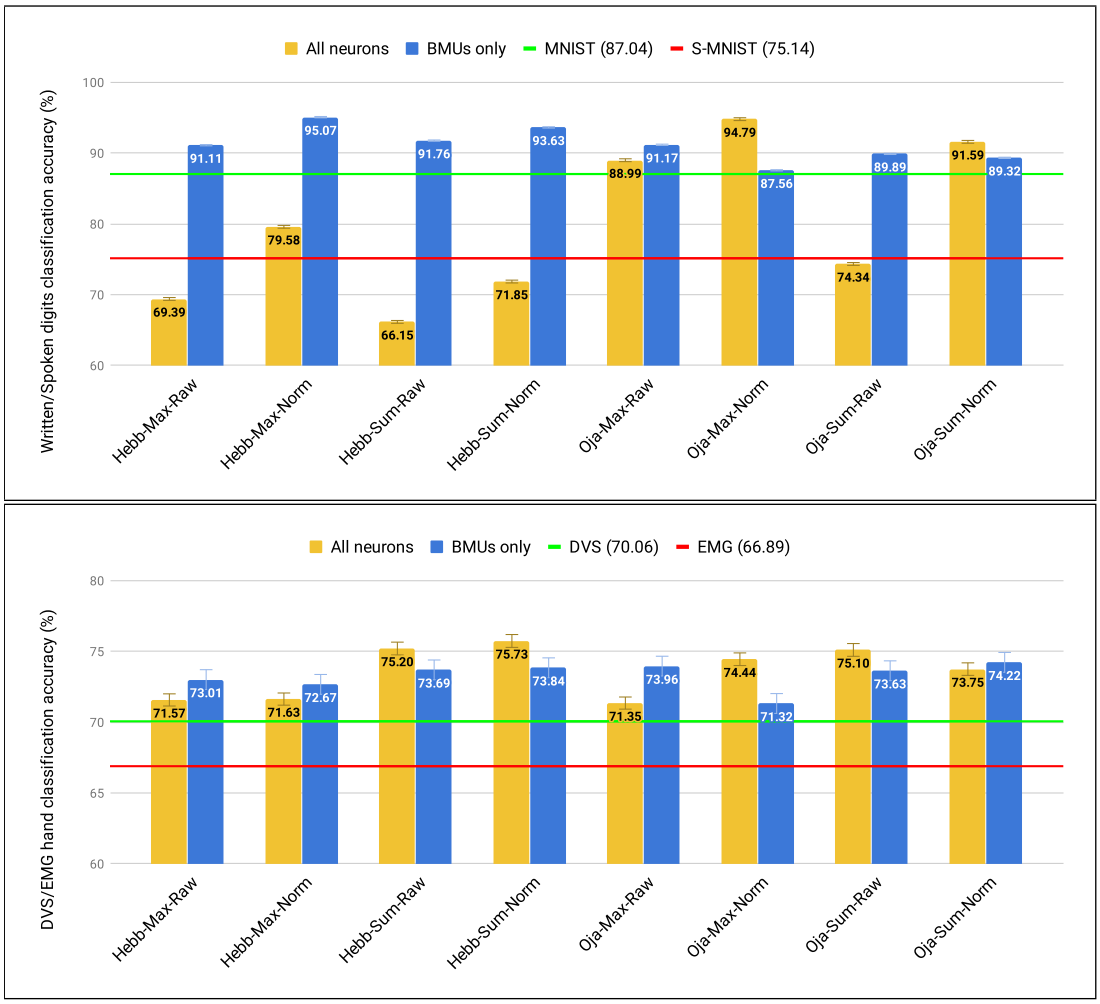

FIGURE 6.4: Multimodal convergence classification: (top) Written/Spoken digits; (bottom) DVS/EMG hand gestures.

6.3.1 ReSOM divergence results

Table 6.1 shows unimodal classification accuracies using the divergence mechanism for labeling, with 75.9% $\pm$ 0.2 for S-MNIST classification and 65.56% $\pm$ 0.25 for EMG classification. As shown in figure 6.3, we reach this performance using respectively 20% and 25% of the potential synapses for digits and hand gestures. We see in figure 6.3 that we need more connections per neuron for the divergence process, because the pruning is done by the neurons of one of the two maps, and a small number of connections results in some disconnected neurons in the other map. Since the pruning is performed by the neurons of the *source* SOMs, i.e. the MNIST-SOM and DVS-SOM, pruning too much synapses causes some neurons of the S-MNIST-SOM and EMG-SOM to be completely disconnected from the source map, and therefore do not get any activity for the labeling process. Hence, the labeling would be incorrect in that case, with the disconnected neurons stuck with the default label 0. In comparison to the standard labeling process with 10% of labeled samples, we have a loss of only -1.33% for EMG, and even a small gain of 0.76% for S-MNIST even though we only use 1% of labeled digits images. The choice of which modality to use to label the other is made according to two criteria: the source map must (1) achieve the best unimodal accuracy so that we maximize the separability of the transmitted activity to the other map, and it must (2) require the least number of labeled data

for its own labeling so that we minimize the number of samples to label during data acquisition. Overall, the divergence mechanism for labeling leads to approximately the same accuracy than the standard labeling. Therefore, we perform the unimodal classification of S-MNIST and EMG with no corresponding labels from end to end.

6.3.2 ReSOM convergence results

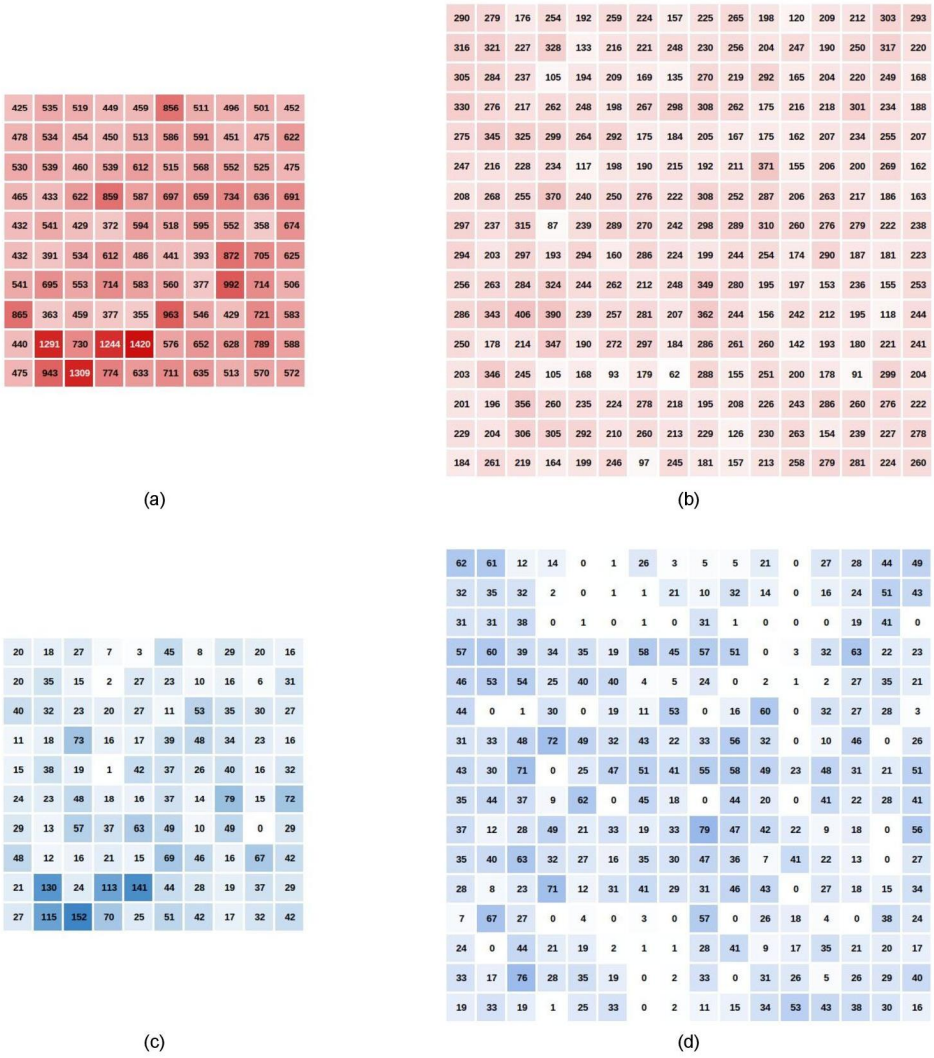

FIGURE 6.5: Written/Spoken digits neurons BMU counters during multimodal learning and inference using $Hebb - Max^{BMU}_{Norm}$ method: (a) MNIST SOM in learning; (b) S-MNIST SOM neurons during learning; (c) MNIST SOM neurons during inference; (d) S-MNIST SOM neurons during inference.

We proposed eight variants of the convergence algorithm for each the two learning methods. For the discussion, we denote them as follow: $Learning - Update^{Neurons}_{Normalization}$ such that *Learning* can be *Hebb* or *Oja*, *Update* can be *Max* or *Sum*, *Normalization* can be *Raw* (the activites are taken as initially computed by the SOM) or *Norm* (all activities are normalized with a min-max normalization thanks to the WMU and BMU activities of each SOM), and finally *Neurons* can be *BMU* (only the two BMUs update each other and all other neurons activities are reset to zero) or *All* (all neurons update their activities and therefore the global BMU can be different from the two

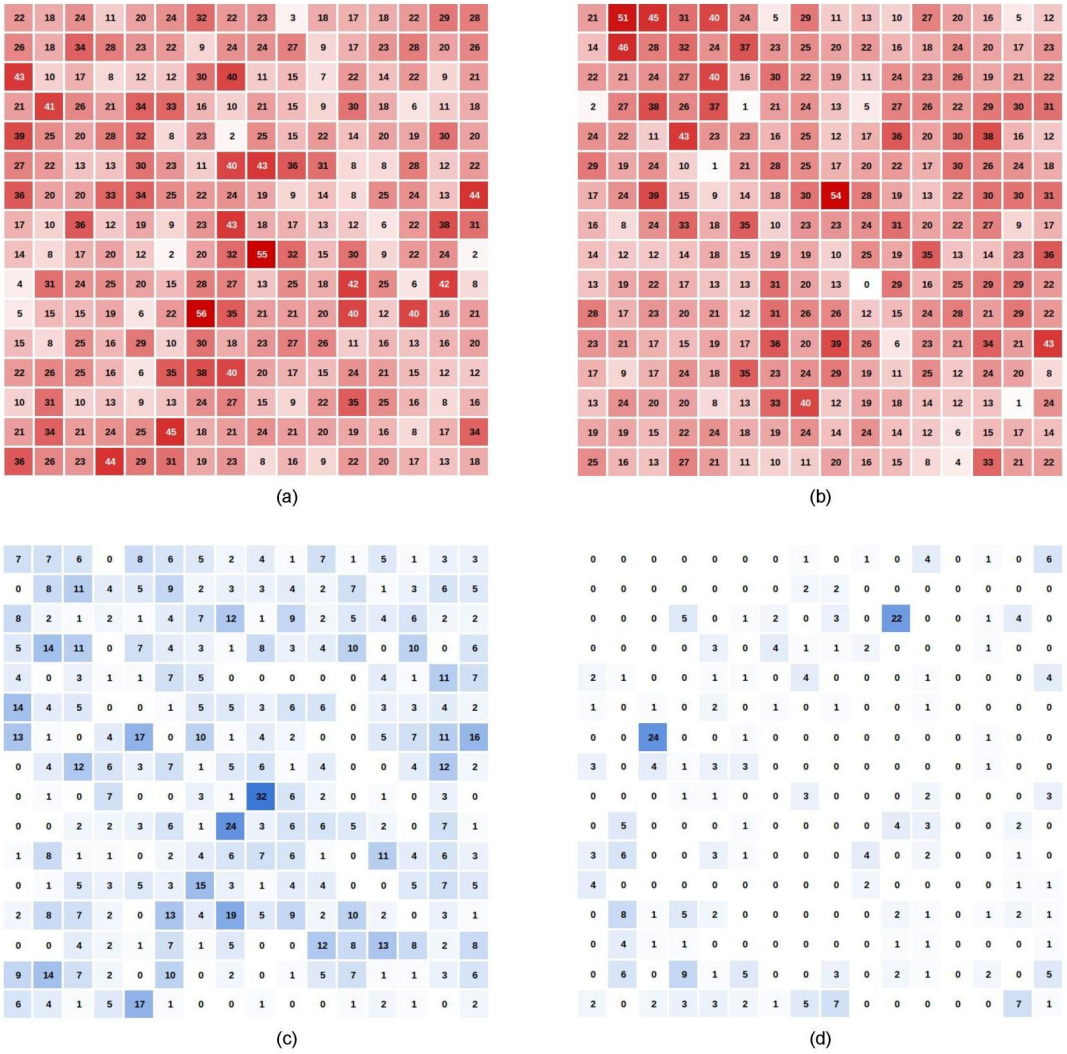

FIGURE 6.6: DVS/EMG hand gestures neurons BMU counters during multimodal learning and inference using $Hebb - Sum^{All}_{Norm}$ method: (a) DVS SOM in learning; (b) EMG SOM neurons during learning; (c) DVS SOM neurons during inference; (d) EMG SOM neurons during inference.

local BMUs). It is important to note that since we constructed the written/spoken digits dataset, we maximized the cases where the two local BMUs have different labels such as one of them is correct. This choice was made in order to better assess the accuracies of the methods based on BMUs update only, as both cases when the two BMUs are correct or incorrect at the same time lead to the same global results regardless of the update method. The convergence accuracies for each of the eight method applied on the two databases are summarized in table 6.2 and figure 6.4. The red and green lines are respectively the lowest and highest unimodal accuracies. Hence, there is an overall gain whenever the convergence accuracy is above the green line.

For the digits, we first notice that the Hebb's learning with all neurons update leads to a very poor performance, worse than the unimodal classification accuracies. To explain this behavior, we have to look at the neurons BMU counters during learning in figure 6.5. We notice that some neurons, labeled as 1 in Figure 3.5-c, are winners much more often that other neurons. Hence, their respective lateral

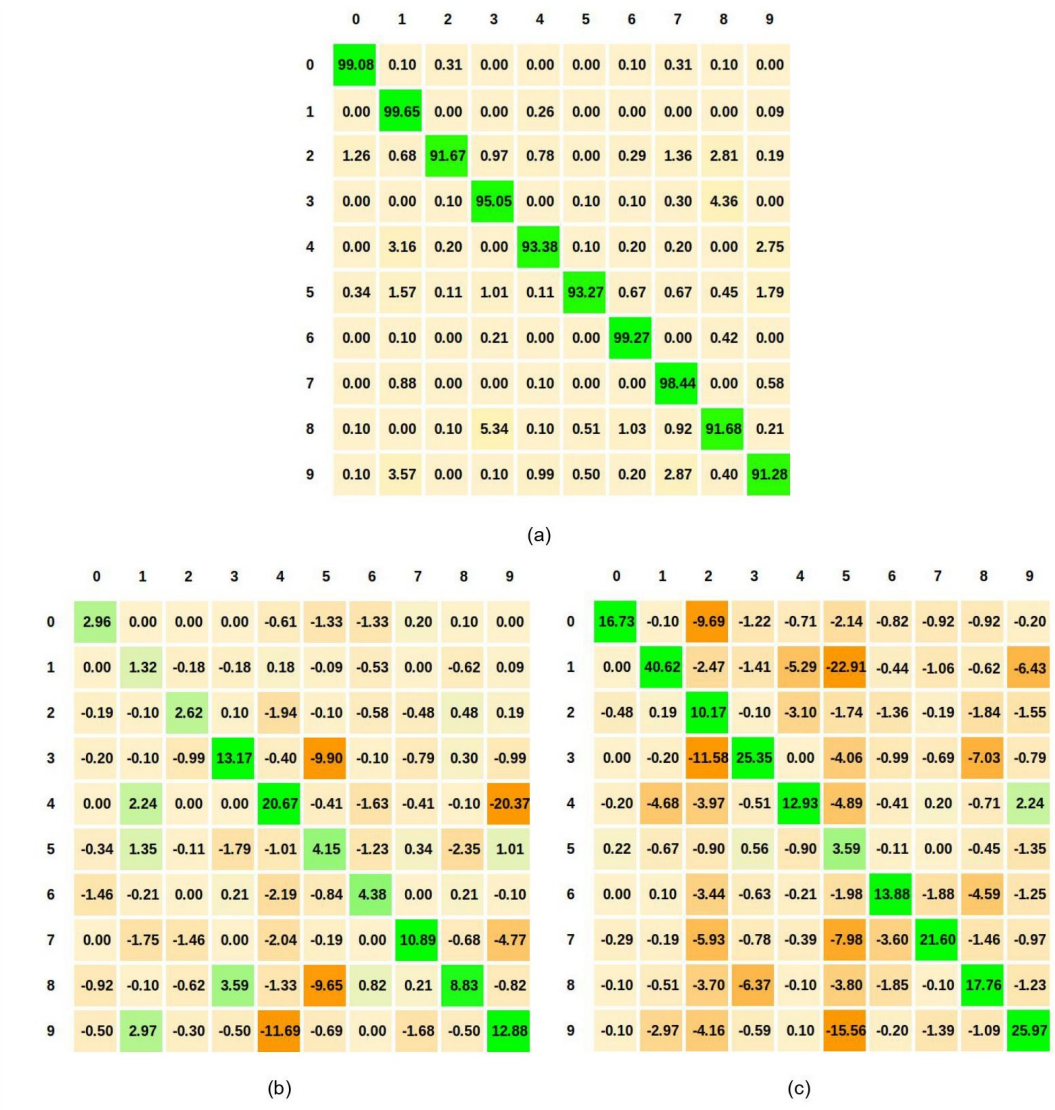

FIGURE 6.7: Written/Spoken digits confusion matrices using $Hebb - Max^{BMU}_{Norm}$ method: (a) convergence; (b) convergence gain with respect to MNIST; (c) convergence gain with respect to S-MNIST.

TABLE 6.2: ReSOM multimodal unsupervised classification accuracies.

Learning	ReSOM convergence method and accuracy (%) $_{\beta}$					
	Update algorithm	Neurons activities	Digits		Hand gestures	
			All neurons	BMUs only	All neurons	BMUs only
Hebb	Max	Raw	69.39 $_{1}$	91.11 $_{1}$	71.57 $_{5}$	73.01 $_{5}$
		Norm	79.58 $_{20}$	**95.07** $_{10}$	71.63 $_{3}$	72.67 $_{20}$
	Sum	Raw	66.15 $_{1}$	91.76 $_{10}$	**75.20** $_{4}$	73.69 $_{4}$
		Norm	71.85 $_{1}$	93.63 $_{20}$	**75.73** $_{4}$	73.84 $_{20}$
Oja	Max	Raw	88.99 $_{4}$	91.17 $_{1}$	71.35 $_{3}$	73.96 $_{10}$
		Norm	**94.79** $_{4}$	87.56 $_{3}$	74.44 $_{30}$	71.32 $_{10}$
	Sum	Raw	74.34 $_{2}$	89.89 $_{3}$	75.10 $_{4}$	73.63 $_{10}$
		Norm	91.59 $_{15}$	89.32 $_{30}$	73.75 $_{4}$	74.22 $_{30}$

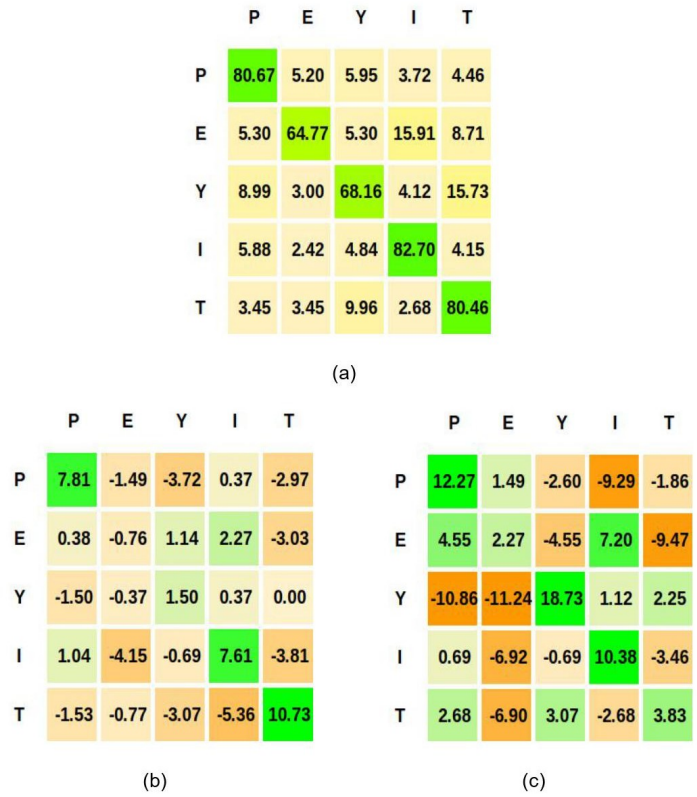

FIGURE 6.8: DVS/EMG hand gestures confusion matrices using $Hebb - Sum^{All}_{Norm}$ method: (d) convergence; (e) convergence gain with respect to DVS; (f) convergence gain with respect to EMG.

synapses weights increase disproportionately compared to other synapses, and lead those neurons to be winners most of the time after the update, as their activity is higher than other neurons very often during convergence. This behavior is due to two factors: first, the neurons that are active most of the time are those that are the fewest to represent a class. Indeed, we have less neurons labeled 1 compared to other classes, because the digit 1 have less *sub-classes*. In other words, the digit 1 has less variants and therefore can be represented by less prototype neurons. Consequently, those neurons are active more often because the number of samples for each class is approximately equal. Second, the Hebb's learning is unbounded, leading the lateral synapses weights to increase indefinitely. Thus, this problem occurs less when we use Oja's rule, as shown in figure 6.4. We notice that Oja's learning leads to more homogenous results, and normalization often leads to a better accuracy. The best method using Hebb's learning is $Hebb - Max^{BMU}_{Norm}$ with $95.07\% \pm 0.08$, while the best method using Oja's learning is $Oja - Max^{All}_{Norm}$ with $94.79\% \pm 0.11$.

For the hand gestures, all convergence methods lead to a gain in accuracy even though the best gain is smaller than for digits, as summarized in Table 6.1. It can be explained by the absence of neurons that would be BMUs much more often than other neurons, as shown in Figure 6.6. The best method using Hebb's learning is $Hebb - Sum^{All}_{Norm}$ with $75.73\% \pm 0.91$, while the best method using Oja's learning is $Oja - Sum^{All}_{Raw}$ with $75.10\% \pm 0.9$. In contrast with the digits database, here the most accurate methods are based on the *Sum* update. Thus, each neuron takes in account the activities of all the neurons that it is connected to. A plausible reason is the fact that the digits database was constructed whereas the hand gestures database was

initially recorded with multimodal sensors, which gives it a more natural correlation between the two modalities.

Overall, the best methods for both digits and hand gestures databases are based on Hebb's learning, even though the difference with the best methods based on Oja's learning is very small, and Oja's rule has the interesting property of bounding the synaptic weights. For hardware implementation, the synaptic weights of the Hebb's learning can be normalized after a certain threshold without affecting the model's behavior, since the strongest synapse stays the same when we divide all the synapses by the same value. However, the problem is more complex in the context of on-line learning as discussed in section 6.6. Quantitatively, we have a gain of $+8.03\%$ and $+5.67\%$ for the digits and the hand gestures databases respectively, compared to the best unimodal accuracies. The proposed convergence mechanism leads to the election of a global BMU between the two unimodal SOMs: it is one of the local BMUs for the $Hebb - Max_{Norm}^{BMU}$ method used for digits, whereas it can be a completely different neuron for the $Hebb - Sum_{Norm}^{All}$ used for hand gestures. In the first case, since the convergence process can only elect one of the two local BMUs, we can compute the *absolute accuracy* in the cases where the two BMUs are different with one of them being correct. We find that the correct choice between the two local BMUs is made in about 87% of the cases. However, in both cases, the convergence leads to the election a global BMU that is indeed spread in the two maps, as shown in figures 6.5 and 6.6. Nevertheless, the neurons of the hand gestures SOMs are less active in the inference process, because we only have 1350 samples in the test database.

The best accuracy for both methods is reached using a sub-part of the lateral synapses, as we prune a big percentage of the potential synapses as shown in figure 6.3. We say "potential" synapses, because the pruning is performed with respect to a percentage (or number) of synapses for each neuron, and the neuron does not have the information of other neurons due to the cellular architecture. Thus, the percentage is calculated with respect to the maximum number of potential lateral synapses, that is equal to the number of neurons in the other SOM, and not the actual number of synapses. In fact, at the end of the Hebbian-like learning, each neuron is only connected to the neurons where there is at least one co-occurrence of BMUs, as shown in figure 6.2. We notice that less than half of the possible lateral connections are created at the end of the Hebbian-like learning, because only meaningful connections between correlated neurons are created. Especially for the hand gestures database, the sprouting leads to a small total number of lateral synapses even before pruning, because of the small number of samples in the training dataset. Finally, we need at most 10% of the total lateral synapses to achieve the best performance in convergence as shown in Figure 6.3. However, if we want to maintain the unimodal classification with the divergence method for labeling, then we have to keep 20% and 25% of the potential synapses for digits and hand gestures, respectively.

One interesting aspect of the multimodal fusion is the explainability of the better accuracy results. To do so, we plot the confusion matrices with the best convergence methods for the digits and hand gestures datasets in Figures 6.7 and 6.8, respectively. The gain matrices mean an improvement over the unimodal performance when they have positive values in the diagonal and negative values elsewhere. If we look at the gain matrix of the convergence method compared to the image modality, we notice two main characteristics: first, all the values in the diagonal are positive, meaning that there is a total accuracy improvement for all the classes. Second and more interestingly, the biggest absolute values outside the diagonal lie where there is the biggest confusion for the images, i.e. between the digits 4 and 9, and between the digits 3, 5 and 8, as previously pointed out in section 6.2. It confirms our initial

hypothesis, which means that the auditory modality brings a complementary information that leads to a greater separability for the classes which have the most confusion in the visual modality. Indeed, the similarity between written 4 and 9 is compensated by the dissimilarity of spoken 4 and 9. The same phenomenon can be observed for the auditory modality, where there is an important gain for the digit 9 that is often mis-classified as 1 or 5 in the speech SOM, due to the similarity of their sounds. Similar remarks are applicable for the hand gestures database with more confusion in some cases, which leads to a smaller gain.

Our results confirm that multimodal association is interesting because the strengths and weaknesses of each modality can be complementary. Indeed, Rathi and Roy (Rathi and Roy, 2018) state that if the non-idealities in the unimodal datasets are independent, then the probability of mis-classification is the product of the mis-classification probability of each modality. Since the product of two probabilities is always lower than each probability, then each modality helps to overcome and compensate for the weaknesses of the other modality. Furthermore, multimodal association improves the robustness of the overall system to noise (Rathi and Roy, 2018), and in the extreme case of losing one modality, the system could rely on the other one which links back to the concept of degeneracy in neural structures (Edelman, 1987).

6.4 Comparative study

In this section, we compare our ReSOM model to three different approaches. First, we compare our results with STDP approaches to assess the classification accuracy with a comparable number of neurons. Next, we confront our results with early data fusion using one SOM. Third and finally, we use supervised perceptrons to learn the multimodal representations based on the two unimodal SOMs activities.

6.4.1 SOM early data fusion

We find in the literature two main different strategies for multimodal fusion (Baltrusaitis, Ahuja, and Morency, 2019) (Cholet, Paugam-Moisy, and Regis, 2019): (1) score-level fusion where data modalities are learned by distinct models then their predictions are fused with another model that provides a final decision, and (2) data-level fusion where modalities are concatenated then learned by a unique model. Our approach can be classified as a classifier-level fusion which is closer to score-level fusion and usually produces better results than feature-level or data-level fusion for classification tasks (Guo et al., 2014) (Peng et al., 2016) (Biagetti, Crippa, and Falaschetti, 2018). However, it is worth trying to learn the concatenated modalities with one SOM having as much neurons as the two uni-modal SOMs, for a fair comparison.

We use 361 and 529 neurons for digits and hand gestures respectively. We have few neurons more compared to the sum of the two uni-modal SOMs, as we want to keep the same square grid topology. We train the SOMs with the same hyper-parameters as for the uni-modal SOMs, and reach $90.68\% \pm 0.29$ and $75.6\% \pm 0.32$ accuracy for digits and hand gestures, respectively. We still have a gain compared to the uni-modal SOMs, but have an important loss of -4.39% for digits and a negligible loss of -0.13% for hand gestures compared to the proposed ReSOM multimodal association. The incremental aspect of the ReSOM from simple (unimodal) to more

complex (multimodal) representations improves the system's accuracy, which is coherent with the literature findings. Furthermore, the accuracy is not the only metric, as the memory footprint is an important factor to take in consideration when choosing a fusion strategy (Castanedo, 2013), especially for embedded systems. Today, implementing a large number of synapses in a neuromorphic device is a great challenge (Strukov et al., 2019). Indeed, since we target a hardware implementation on FPGA, the total number of afferent and lateral synaptic weights are parameters that require on-chip memory, which is very limited. With a simple calculation using the number of neurons and input dimensions, we find that we have a gain of 49.84% and 40.96% for digits and hand gestures respectively (before the lateral pruning) using the multimodal association compared to a data-level fusion strategy.

6.4.2 Confronting SOMs to SNNs for multimodal association

TABLE 6.3: Digits unsupervised classification comparison.

ANN	Model	Neurons	Labels (%) *	Modality	Dataset	Accuracy (%)
SNN	(Diehl and Cook, 2015)	400	100	Unimodal	MNIST	88.74
	(Hazan et al., 2018)	400	100	Unimodal	MNIST	92.56
	(Rathi and Roy, 2018)	400	100	Unimodal	MNIST	86.00
	(Rathi and Roy, 2018)	400	100	Multimodal	MNIST + TI46	89.00
SOM	(Khacef et al., 2020)	356	1	Multimodal	MNIST + SMNIST	**95.07**

* Labeled data are only used for the neurons labeling after unsupervised training.

Table 6.3 summarizes the digits classification accuracy achieved using brain-inspired unsupervised approaches, namely SOMs with self-organization (Hebb, Oja and Kohonen principles) and SNNs with STDP. We achieve the best accuracy with a gain of about 6% over Rathi and Roy (Rathi and Roy, 2018) with approximately the same number of neurons, which is to the best of our knowledge the only work that explores brain-inspired multimodal learning for written/spoken digits classification. It is to note that we do not use the TI46 spoken digits database (Liberman et al., 1991), but a subpart of GSC (Warden, 2018) because the TI46 is not freely available.

We notice from table 6.3 that all other works use the complete training dataset to label the neurons, which is incoherent with the first goal of not using labels, as explained in chapter 3. Moreover, the work of Rathi and Roy (Rathi and Roy, 2018) differs from our work in the following points:

- The cross-modal connections are formed randomly and initialized with random weights. The multimodal STDP learning is therefore limited to connections that have been randomly decided, which induces an important variation in the network performance.

- The cross-modal connections are not bi-directional, thus breaking with the biological foundations of reentry and CDZ. Half the connections carry spikes from image to audio neurons and the other half carry spikes from audio to image neurons, otherwise making the system unstable.

- The accuracy goes down beyond 26% connections. When the number of random cross-modal connections is increased, the neurons that have learned different label gets connected. We do not observe such a behavior in the ReSOM, as shown in figure 6.3.
- The decision of the multimodal network is computed by "observing" the spiking activity in both ensembles, thus requiring a central unit.

Nevertheless, the STDP-based multimodal learning is still a promising approach for the hardware efficiency of SNNs (Khacef, Abderrahmane, and Miramond, 2018), and because of the alternative they offer for using even-based sensors with asynchronous computation.

6.4.3 SOMs coupled to supervised fusion

In order to have an approximation of the best accuracy that we could obtain with multimodal association, we used a number of perceptrons equal to the number of classes on top of the two uni-modal SOMs of the two databases, and performed supervised learning for the same number of epochs (10) using gradient descent (Adadelta algorithm). We obtain $91.29\% \pm 0.82$ and $80.19\% \pm 0.63$ of accuracy for the digits and hand gestures respectively. Surprisingly, we have a loss of -3.78% for the digits. However, we have a gain of 4.43% for the hand gestures. We argue that the hand gestures dataset is too small to construct robust multimodal representations through unsupervised learning, and that could explain the smaller overall gain compared to the digits dataset.

6.5 Coupling DVS hand gestures with spoken digits

6.5.1 Motivation and goal

An important question in developmental robotics is how the conceptual system and language co-develop (Pointeau, Petit, and Dominey, 2014). In cognitive and developmental psychology (Waxman and Markow, 1995) (Waxman and Braun, 2005), words are often seen as an invitation to the infant to form categories. Infancy research demonstrates indeed a facilitation of visual category formation in the presence of verbal *labels*. The term *label* refers here to the spoken modality of the information when, for example, infants learn to associate objects with their names in a particular language (this *label* is written in *italic* in the rest of this section to distinguish from the label as the class of the sample). However, there is an open question about the role of these verbal *labels*: do they function as features that increase separability between objects (Gliozzi et al., 2009) or do they act as a referential, serving as "names" (“Early word-learning entails reference, not merely associations” 2009)?

Several works in experimental psychology tend to pick up the first answer. For example, Plunkett et al. (Plunkett, Hu, and Cohen, 2008) highlighted the constructive effects of *labels* on categorization. In their study, 10-month-old infants were provided with identical labels and formed a single category over a set of stimuli. However, they divided the same stimuli into two groups when familiarized in silence. It indicates that infants relied on the *label*'s identity to form categories. Similarly, Althaus and Westermann demonstrated in (Althaus and Westermann, 2016) that *labeling* caused infants to use more restrictive criteria for a classification of two items as similar, effectively producing two smaller categories. These experimental

results show that the so-called *label* serves not so much as an additional "name" but instead as a feature that modulates the way in which visual features are classified. It suggests that object categories and words develop interactively in infants (Althaus and Mareschal, 2013) to facilitate categorization (Althaus and Mareschal, 2014).

In the following, we experiment our ReSOM model on a third database that we construct using two previously seen unimodal databases: the DVS hand gestures for the visual modality and the spoken digits for the auditory modality, as further detailed in section 6.5.2.

6.5.2 Database construction

We have already shown in this chapter that spoken digits help to improve the representation of written digits, providing *labels* (as referred to in psychology) that improve the digits classification. The objective with this third database is to confirm that the auditory modality can improve the representation of the visual modality with a complex dataset such as the DVS hand gestures, even though the auditory dataset was associated offline to the DVS visual modality. In fact, we could associate any gesture to a spoken *label*, even though both do not represent the same "concept" in our common representation. The ReSOM model exploits the intrinsic complementarity of multiple modalities, we can therefore expect an improvement in accuracy when associating both modalities.

A necessary and sufficient condition when constructing the dataset is to associate every class in the DVS hand gestures dataset to a unique class in the spoken digits datatset. The goal is to retrieve the co-occurrence of the multimodal information which is necessary for the multimodal association learning. We have arbitrarily chosen to couple the classes as follows: *Pinky - one*, *Elle - two*, *Yo - three*, *Index - four* and *Thumb - five*. We therefore extracted only 5 classes from the 10 classes of S-MNIST, which made a database of 17204 samples for training and 2073 samples for test. We call this spoken digits database of 5 digits S-MNIST-5. Since we have less samples in the DVS hand gestures than in S-MNIST, we duplicated some random DVS hand gestures to match the number of spoken digits.

6.5.3 ReSOM divergence and convergence results

TABLE 6.4: ReSOM classification accuracy and convergence/divergence gain for DVS hand gestures with spoken *labels*.

Database		DVS hand gestures	S-MNIST-5
SOMs	Dimensions	972	507
	Neurons	256	256
	Labeled data (%)	10	10
	Accuracy (%) $_{\alpha}$	70.06 $_{2.0}$	84.62 $_{0.1}$
ReSOM divergence	Labeled data (%)	0	10
	Gain (%)	2.01	/
	Accuracy (%)	72.07	/
ReSOM convergence	Gain (%)	+18.3	**+5.75**
	Accuracy (%)	**90.37**	

In this section, we present the results from our experiments with the multimodal association convergence and divergence mechanisms for the third database of DVS

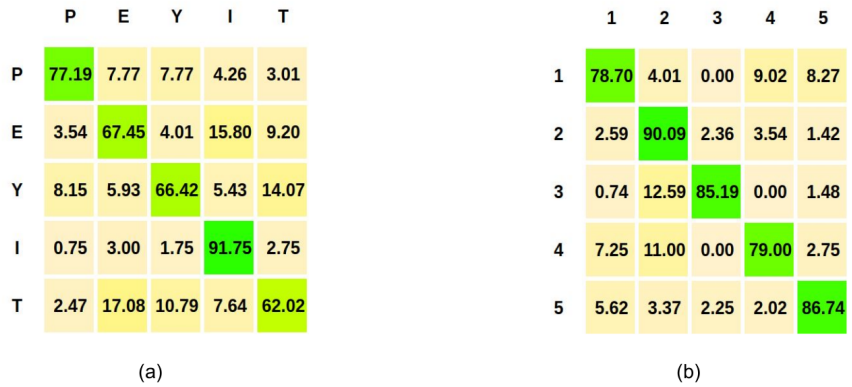

FIGURE 6.9: KSOM learning confusion matrix: (a) DVS hand gestures divergence; (b) S-MNIST-5.

hand gestures associated with S-MNIST-5. The first step was to train and label a SOM with the unimodal S-MNIST-5 database, using the same hyper-parameters as the original S-MNIST database as reported in table 6.4. We achieved an accuracy of 84.62% $\pm$ 0.4 on S-MNIST-5, which is better than the S-MNIST accuracy because S-MNIST-5 only contains 5 classes. For the DVS hand gestures, we used the same SOM that was previously trained for the DVS/EMG hand gesture multimodal association in section 6.2.3, with an accuracy of 70.06% $\pm$ 1.15. Figure 6.9 shows that the two SOMs have different confusion matrices, meaning that their respective misclassifications do not occur within the same classes with the same percentage. There is therefore a potential for the ReSOM to exploit this complementarity in order to improve the overall accuracy. Afterwards, we trained the multimodal association as explained in chapter 5 with sprouting, Hebbian-like learning and pruning to keep 20% of all lateral connections.

Then, we move to the ReSOM divergence mechanism, i.e. labeling the DVS hand gestures SOM with the lateral activity from the S-MNIST-5 SOM. As explained before, we use the SOM with the best accuracy as the reference map that we use to label all other maps, so that we maximize the separability of the lateral activity. The results in table 6.4 show that the DVS hand gestures classification reaches 72.07% $\pm$ 0.46 of accuracy using the divergence mechanism for labeling, which means a gain of 2.01% compared to the original afferent labeling with DVS hand gestures data. It shows that even though both modalities need approximately the same amount of labels (10%), we still have a gain in performance regarding the unimodal accuracy of the labeled SOM with the divergence mechanism.

Moreover, the ReSOM divergence performance is very interesting in the context of experimental studies with infants. For example, Gliga et al. (Gliga, Volein, and Csibra, 2010) measured the induced EEG gamma-band activation in the brain of 12-month-old infants. They showed that hearing previously *labeled* objects modulated visual processing of that object, even though the object was not shown. This is what the ReSOM simulates for *labeling* the DVS hand gestures visual modality with the S-MNIST-5 spoken *labels*. Furthermore, Mani and Plunkett (Mani and Plunkett, 2010) demonstrated that infants as young as 18 months implicitly generate phonological representations upon seeing a picture for which they know a word, which demonstrates the bi-directional aspect of the lateral connections and provides experimental evidence for the ReSOM biological plausibility.

Finally, we conducted a comparative study for the eight variants of the ReSOM

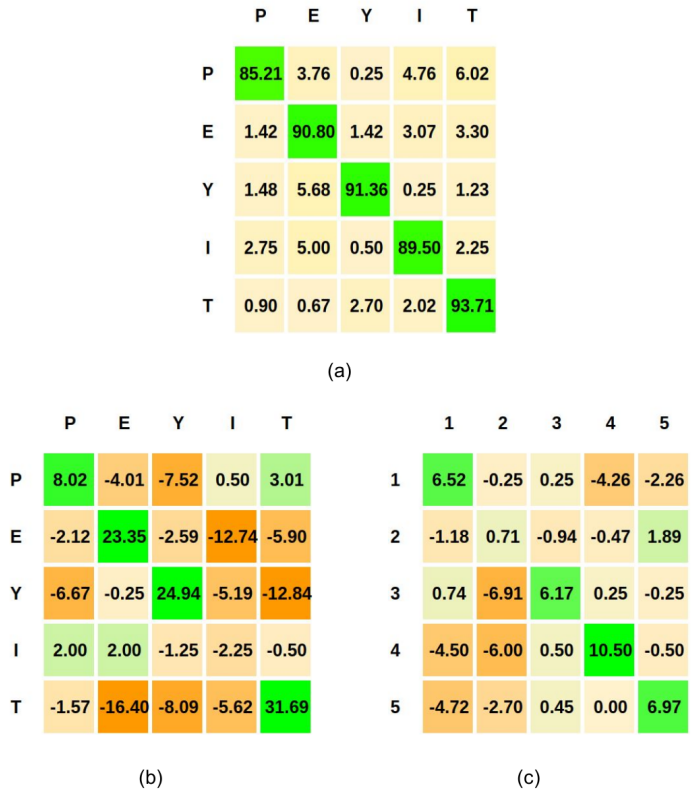

FIGURE 6.10: DVS hand gestures and spoken digits confusion matrices using $Hebb - Max^{BMU}_{Norm}$ method: (a) convergence; (b) convergence gain with respect to DVS hand gestures; (c) convergence gain with respect to S-MNIST-5.

convergence for each of the two learning methods, i.e. Hebb's and Oja's learning, as explained in section 6.3.2. Similarly to the written/spoken digits convergence, the best performance was obtained using $Hebb - Max^{BMU}_{Norm}$ ($\beta = 50$) with 90.37% $\pm$ 0.3, which means a gain of +5.75% compared to the best unimodal accuracy achieved with S-MNIST-5 as reported in table 6.4. Interestingly, figure 6.10 shows that both modalities had a positive *impact* on each other, leading to an increase in accuracy for every class except *Index* in the DVS hand gestures. It is mainly because it was already well classified in the unimodal SOM. Overall, the confusion matrices of gain in figure 6.10 reflect well the better accuracy of the ReSOM compared to the unimodal SOMs.

The ReSOM convergence results are coherent with respect to recent computational models in cognitive and developmental psychology, in which authors try to simulate the role of word *labels* for categorization. For example, Althaus and Mareschal (Althaus and Mareschal, 2013) demonstrated how early interactions between word learning and learning about objects led to improved category representations compared to isolated learning without spoken words acting as *labels*. Interestingly, the model of (Althaus and Mareschal, 2013) uses SOMs and Hebbian connections to propagate activation between the visual and auditory maps during learning. Even though no classification accuracy was reported, the results show that categorical perception emerges from these early audio–visual interactions in both domains. Another example is the work of Westermann and Mareschal (Westermann and Mareschal, 2014) who proposed a model that shows how spoken *labels* can affect the similarity relations between objects, by increasing the perceptual distance

between objects that have different *labels*. These results contrast with the notion that words are simply mapped onto previously existing categories. They reinforce the idea that spoken *labels* and the auditory modality in general adds a complementary information that affects the classification accuracy. These findings are supported by the ReSOM convergence results.

6.6 Discussion

6.6.1 A universal multimodal association model?

The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence (Fiebelkorn, Foxe, and Molholm, 2010) (Ursino, Cuppini, and Magosso, 2014). Similarly to (Vavrecka and Farkas, 2013), (Morse et al., 2015) and (Parisi et al., 2017) and based on our experimental results, we argue that the co-occurrence of sensory inputs is a sufficient source of information to create robust multimodal representations. This is achieved using associative links between unimodal representations that can be incrementally learned in an unsupervised fashion.

In terms of learning, the best methods are based on *Hebb*'s learning with a slightly better accuracy over *Oja*'s learning, but the overall results are more homogeneous using *Oja*'s learning that prevents the synaptic weights from growing indefinitely. The best results are obtained using $Hebb - Max^{BMU}_{Norm}$ with $95.07\% \pm 0.08$ and $Hebb - Sum^{All}_{Norm}$ with $75.73\% \pm 0.91$ for the digits and hand gestures datatabases, respectively. We notice that the BMU method is coupled with the Max update while the All neurons method is coupled with the Sum update, and the $Norm$ activities usually perform better than Raw activities. However, we cannot have a final conclusion on the best method, especially since it depends on the nature of the dataset.

Moreover, the experimental results depend on the β hyper-parameter, the Gaussian kernel width that has to be tuned for every database and every method. Thanks to the multiplicative update, the values of both SOMs are brought into the same scale which gives the possibility to elect the correct global BMU, and we get rid of a second hyper-parameter that would arise with a sum update method like in (Jayaratne et al., 2018). However, it is still time-taking in the exploration of the proposed methods for future works, even if it is a common limit when dealing with any ANN. Finding a more efficient approach for computing β will be part of our future works.

Finally, multimodal association bridges the gap between unsupervised and supervised learning, as we obtain approximately the same results compared to unimodal MNIST using a supervised Multi-Layer Perceptron (MLP) with 95.73% (Khacef, Abderrahmane, and Miramond, 2018) and S-MNIST using a supervised attention Recurent Neural Network (RNN) with 94.5% (Andrade et al., 2018) (even though this result was obtained on 20 commands rather than 10). Multimodal association can also be seen as a way to reach the same accuracy of about 95% as (Diehl and Cook, 2015) with much less neurons, going from 6400 neurons to 356 neurons, i.e. a gain of 94% in the total number of neurons. It therefore is a very promising approach to deeper explore, as we have in most cases the possibility to include multiple sensory modalities when dealing with the real-world environment.

6.6.2 Offline vs. online multimodal association learning

The ReSOM multimodal association learning methods explored in this work are performed sequentially in two times: first, we train the SOMs for unimodal classifications, and second we create and reinforce bidirectional connections between the two maps based upon their activities on the same training dataset. We refer to this learning approach as *asymmetric*. This is particularly interesting in the context of off-line learning when working on pre-established datasets. First, from a purely practical way, it gives a lot of flexibility since we could train the unimodal SOMs on their respective available data separately, then train the multimodal association based on a smaller synchronized multimodal dataset. The synchronicity here means that the multimodal samples that belong to the same class are presented at the same time. Second, from a developmental point of view, it has been shown that auditory learning begins before birth while visual learning only starts after birth (Althaus and Mareschal, 2013). Moreover, the ability to build associations between words and objects in infants appears to develop at about 14 months of age (Werker et al., 1998). The opportunity to process visual and auditory information sequentially may offer computational advantages in infants learning, as it could be a facilitating factor in the extraction of the complex structures needed for categorisation (Althaus and Plunkett, 2015). These observations support the actual learning approach of the ReSOM, where multimodal associations begin to develop after unimodal representations are learned sequentially.

Nevertheless, in the context of on-line learning in a dynamic and changing environment, another approach would be to perform both Kohonen-like and Hebbian-like learning at the same time, continuously. We refer to this learning apporach as *symmetric*. For example, this approach is followed with STDP learning in (Rathi and Roy, 2018). For this purpose, The KSOM would be replaced by the DSOM presented in chapter 2. The reason is that the KSOM has a decaying learning rate and neighborhood width, so that the learning stabilizes after a certain number of iterations. Therefore, the learning is stable but not dynamic. It can be considered as an off-line unsupervised learning algorithm. In contrast, the DSOM introduced by Rougier et al. (Rougier and Boniface, 2011) is a variation of the KSOM algorithm where the time dependency of the learning rate and neighborhood function has been replaced by the distance between the BMU and the input stimulus. Even if the DSOM is less accurate than the KSOM, as shown in chapter 3, it is more suitable for on-line learning. Moreover, Oja's learning would be the only alternative because we need the forgetting parameter that enables the synaptic weights decaying, which is not available in Hebb's learning. In addition, we would need a dynamic learning rate so that the multimodal association becomes stronger when the sample is well learned by the SOM, i.e. when the distance between the BMU and the sample is small. One way to do that is to add a Gaussian kernel to that distance, so that the multimodal binding becomes more relevant after the convergence of the SOMs without any manual tuning on the hyper-parameters of the SOM.

6.6.3 SOMA: Toward hardware plasticity

As discussed in chapter 1, this work is part of the SOMA project (Khacef et al., 2018), where the objective is to study neural-based self-organization in computing systems and to prove the feasibility of a self-organizing multi-FPGA hardware structure. The concept of the IG is further supported in (Heylighen and Gershenson, 2003) as it states the following: "Changes initially are local: components only interact with

their immediate *neighbors*. They are virtually independent of components farther away. But self-organization is often defined as global order emerging from local interactions". Moreover, it states that "a self-organizing system not only regulates or adapts its behavior, it creates its own organization. In that respect it differs fundamentally from our present systems, which are created by their designer". Indeed, the multimodal association through Hebbian-like learning is a self-organization that defines the inter-SOMs structure, where neurons are only connected to each other when there is a strong correlation between them. That's a form of *hardware plasticity*. The hardware gain of the ReSOM is therefore the gain in communication support, which is proportional to the percentage of remaining synapses for each neuron after learning and pruning. Indeed, the multimodal learning via structural and synaptic plasticity allows the pruning of the majority of the potential lateral synapses amongst the SOMs. It reduces the number of connections, thus the number of communications and therefore the overall energy consumption. Hence, the system is more energy-efficient as only relevant communications are performed without any control by an external expert.

6.7 Conclusion

We have experimented in this chapter our ReSOM model and demonstrated its good performance in divergence and convergence for three datasets. First, we have shown that the divergence labeling leads to approximately the same unimodal accuracy as when using labels, with even a consequent gain of $+2.01\%$ for the DVS hand gestures labeled using spoken digits. The divergence mechanism is hence extremely interesting for real-world applications: even though we usually have the same number of labels in all modalities, it happens that the divergent lateral labeling performs better than the afferent labeling with the same amount of labels, because the lateral activity is "pre-processed" by the SOM and has therefore a better separability.

Second, we showed that the ReSOM convergence leads to a gain in the multimodal accuracy of $+8.03\%$ for the written/spoken digits database, $+5.67\%$ for the DVS/EMG hand gestures database and $+5.75\%$ for the DVS hand gestures/spoken digits. The ReSOM exploits the natural complementarity between different modalities like sight and sound as shown by the confusion matrices, so that they complete each other and improve the multimodal classes separability and classification. In addition to the ReSOM biological plausibility with respect to Edelman (Edelman, 1982) and Damasio (Damasio, 1989) theories in cognitive neurosciences, our results find an echo amongst many works in cognitive and developmental psychology in infants (Althaus and Mareschal, 2013). There experimental studies support our multimodal learning approach and both the convergence and divergence mechanisms.

Implemented on the IG, the ReSOM's inter-map structure is therefore learned along the system's experience through self-organization and not fixed by the user. It leads to a gain in the communication time compared to the fully-connected topology without pruning. Indeed, this gain is proportional to the number of pruned lateral synapses for each neuron, which is about 80% of the possible connections. In addition to the convergence and divergence gains, the ReSOM self-organization induces a form of hardware plasticity which has an impact on the hardware efficiency of the system, reducing its overall energy consumption. That is a first result that opens very interesting perspectives for future designs and implementations of self-organizing architectures inspired from the brain's plasticity.

Chapter 7

Conclusion and further works

> Listen to the technology; find out what it's telling you.
>
> *Carver Mead.*

7.1 Conclusion

We have seen through this manuscript that brain-inspired computing is an interdisciplinary field in the intersection of many disciplines amongst which we find embedded electronics, developmental robotics, cellular computing, machine learning, computational and theoretical neurosciences, cognitive psychology, philosophy, etc. The objective of this thesis was to explore brains-inspired computing from these different perspectives and bring *the pieces of the puzzle all together*, in order to propose a computational model that satisfies some particular behavioral, algorithmic and hardware properties. For this purpose, this manuscript was built around one fundamental paradigm of brain's computation: the self-organization.

We have started in chapter 2 by defining *our* foundations for brain-inspired computing (in three levels): multimodal unsupervised learning with structual and synaptic plasticities (behavioral level), cellular computing (algorithmic level) and neuromorphic implementation (hardware level). These properties guided us toward the choice of the SOM model associated with the proposed IG cellular neuromorphic architecture as the main component for unimodal processing in our multimodal framework. Then, we introduced in chapter 3 the *post-labeled unsupervised learning* problem in which training is fully unsupervised, then very few labels are used to name the learned representations with a proposed labeling method. In fact, one major limit in the literature is the use of the whole labeled training set for labeling. From then, we confronted the Kohonen-based SOMs with STDP-based SNNs on MNIST unsupervised classification. The comparative results showed that the SOM achieves a better accuracy using only 1% of labels with the same number of neurons (100). Moreover, we have analytically demonstrated that the SOM with the IG cellular substrata is scalable in terms of time complexity and connectivity complexity, as opposed to centralized or fully-connected architectures. This was an important step in the construction of the proposed multimodal framework which needs to scale-up to multiple SOM networks.

The next step was to show that the SOM could achieve better results in MNIST unsupervised classification and reach a good performance with more complex datasets such as mini-ImageNet without increasing exponentially the number of neurons. Instead, we investigated feature extraction in chapter 4 for two case studies. First, in the context of unsupervised learning, we conducted a comparative study for unsupervised feature extraction, and concluded that the SCAE+SOM achieves a better

accuracy than the SNN+SOM thanks to the sparsity constraints that were applied to the SCAE. We improved the SOM classification by +6.09% and achieved state of the art performance on MNIST unsupervised classification, using only 256 neurons with post-labeled unsupervised learning. Second, in the context of transfer learning, we proposed a solution that combines transfer learning and the SOM. We reached a good performance on the mini-ImageNet few shot classification benchmark, which proves that the SOM can handle complex data provided that we have an efficient feature extraction strategy.

At this point of the work, the unimodal learning and inference using the SOM reached state f the art performance in terms of classification accuracy, with the advantage of hardware scalability. Therefore, we proposed in chapter 5 the ReSOM model inspired from the brain's self-organization for multimodal unsupervised learning. Based on the reentry theory of Edelman (Edelman, 1982), the ReSOM learns unimodal representations with multiple SOMs supported by the IG cellular substrata, then it creates and reinforces the multimodal association via sprouting, Hebbian-like learning and pruning. It relies on both structural and synaptic plasticities that enable self-organization. Thanks to the bi-directional property of the multimodal representation, we exploited both convergence and divergence mechanisms highlighted by Damasio (Damasio, 1989) in classification tasks: the divergence mechanism was used to label one modality based on the other, and the convergence mechanism was used to introduce cooperation and competition between the modalities and improve the overall accuracy of the system.

Finally, the ReSOM experiments described in chapter 6 have been conducted on three different datasets. The results show that, on the one hand, the ReSOM divergence labeling leads to approximately the same unimodal accuracy as when directly using labels. Interestingly, the divergent lateral labeling can perform better than the afferent labeling with the same amount of labels, because the lateral activity is "preprocessed" by the SOM and has therefore a better separability. On the other hand, the ReSOM convergence mechanism leads to a gain in the multimodal accuracy of +8.03% for the written/spoken digits database, +5.67% for the DVS/EMG hand gestures database and +5.75% for the DVS hand gestures/spoken digits. Interestingly, whether the database is originally recorded with multiple sensors or artificially constructed for experimental purposes, the ReSOM exploits the natural complementarity between different modalities to improve the classification accuracy.

Overall, this work was a step forward in our understanding of the important concepts and paradigms of brain-inspired computing in the behavioral, algorithmic and hardware levels. We used these new insights gathered from different and complementary disciplines to complete previous works in the literature with the proposed ReSOM model. It assembles all our efforts and provides the community with a quantitative analysis of the gap we bridged in the three levels with respect to our foundations listed in chapter 2 and discussed in the following:

- **Behavioral level**: The ReSOM learns from multiple modalities in a completely unsupervised fashion, then needs very few labels from one of the learned modalities for the neurons labeling process, as opposed to most works in the literature where all labels are used. In addition to the redundancy provided by multiple modalities which keeps the system working in the case of the loss of one modality, the results show a consequent increase in accuracy for the three experimented datasets, both using raw data and extracted features. The latter is further discussed in section 7.2.2. The ReSOM biological plausibility regarding the works in neuroscience (Edelman, 1982) (Damasio, 1989) is further

supported by experimental results in cognitive and developmental psychology in infants (Althaus and Mareschal, 2013). The proposed ReSOM model can therefore serve as a base to further explore the self-organizing multimodal association mechanisms for two objectives: first, a more detailed understanding of biological observations and, second, a better performance in classification tasks. Nevertheless, the question of the sensori-motor loop was not handled in our experiments, as further explained in section 7.2.2.

- **Algorithmic level**: The ReSOM neuromorphic architecture is based on the IG cellular substrata, which gives the model scalability properties in terms of time complexity and connectivity complexity. The ReSOM exploits the huge potential of cellular automata, where the global behavior emerges from local interactions amongst neurons without any centralized controller.

- **Hardware level**: The hardware implementation was not handled in this thesis. Nevertheless, the architectural design was studied with the IG which is the cellular substrata that supports the ReSOM computation, and in turn takes profit from the structural plasticity mechanism that reduces the lateral connections. This aspect is further discussed in section 7.2.1.

We believe that the studied self-organization concepts and paradigms at different levels and from different disciplines should be studied together as much as possible, because only then we can handle the remaining scientific and technical challenges of brain-inspired computing. Our results are promising and encourage this interdisciplinary approach.

7.2 Perspectives

In this final section, we discuss the limits and perspectives of our work, especially regarding the ReSOM model in two levels: the hardware level and the behavioral level with respect to the sensori-motor interaction in the environment.

7.2.1 From brain's plasticity to hardware plasticity

The proposed ReSOM model is based on the IG substrata for scalability purposes, which means that each neuron computes in a distributed fashion and exchanges information only with its local neighbor neurons. Therefore, the ReSOM self-organizing neuromorphic architecture is adapted to the multi-FPGA hardware structure targeted in the SOMA project. The idea for future works is to process each modality in a specific FPGA board, then implement the multimodal mechanisms through the boards inter-connections based on the SCALP plateform proposed by Vannel et al. (Vannel et al., 2018). These works have to focus particularly on the communication part between multiple FPGA boards.

Importantly, the hardware inter-map structure is not fixed by the user but self-organized along the system's experience. This self-organization of the ReSOM's lateral connections impacts the hardware that takes profit from the structural plasticity. This comes from the pruning mechanism that reduces the number of connections compared to a non-adaptive fully-connected topology. It reduces thus the number of communications and therefore the overall energy consumption of the hardware. The energy-efficiency gain of the hardware plasticity would be proportional to the number of pruned lateral connections which is about 80% of the possible connections,

but this claim needs to be further quantified in short-term future works through measurement on the FPGA hardware.

Mid-term future works will focus on transposing the ReSOM model into the spiking domain for direct integration with event-based sensors. The idea is to replace the SOMs with SNNs and the Hebbian-like learning with symmetric STDP. The main difference would be an additional hyper-parameter which is the time window during which every pixel is converted into spikes in the case of original frame-based data, or the time window during which the modalities should be presented in the case of event-based data. A first interesting attempt to that was done by Rathi et al. (Rathi and Roy, 2018) with some limitations as explained in chapters 5 and 6. Therefore, it would be an extension of the comparative study that we started in this thesis between SOMs and SNNs to multimodal association in terms of accuracy, scalability and energy-efficiency.

7.2.2 Toward intelligent artificial systems

The ReSOM model exhibits very interesting properties of self-organization, and provides a base-line model to improve toward the design of *intelligent artificial systems* that can autonomously learn and adapt in their environment. A technical limit of the ReSOM multimodal association learning and convergence is the hyper-parameter β in equation 5.7. Even though it is common to ANNs in general, it should be mentioned because it is a true limit when put in the context of autonomous systems that should not need any human *supervision*. In our work, this hyper-parameter was approximated with a simple grid-search approach. Future works may tackle this question to figure the optimal function for tuning this particular hyper-parameter.

Thereafter comes the question of feature extraction, which is necessary when working on real-world data with complex structures. We have seen that the ReSOM is capable of handling extracted features, and that unsupervised feature extraction provides good performance on MNIST either using an gradient-based SCAE or an STDP-bases SNN associated with the SOM. However, the task is much more difficult for complex datasets, and that is why we experimented the DVS/EMG hand gestures with supervised features. Mid-term works should focus on more generic strategies for unsupervised feature extraction, by providing a new metric for the quality of the features. We have seen that sparsity is part of that metric, but our experimental results have shown that it cannot be considered as a reliable quantification alone.

Finally, the ReSOM algorithm is generic in terms of the number of modalities to be used. As a short-term objective, we can add a third modality to the framework by investigating a DVS/EMG hand gestures/spoken digits. Since the overall accuracy is an *emergent* result of the self-organization, it is difficult to provide an hypothesis on the overall accuracy gain we can expect from three modalities. Furthermore, the third modality can be a motor modality, following the work of Lallee and Dominey (Lallee and Dominey, 2013). Their proposed model encoded the sensori-motor experience of a robot based on the visual, auditory and motor modalities, then used the learned representations to control the robot behaviour. However, the model's computation relies on a centralized controller. As a long-term objective, the ReSOM model must be extended to the motor modality because it is a necessary condition for a complete biologically plausible model at the behavioral level, and because self-organization coupled with the sensori-motor interaction in the environment is the key to enable intelligence in biological as well as artificial systems.

Appendix A

GPU-based software implementation for fast simulation

A.1 TensorFlow-based SOM

The SOM was implemented using TensorFlow (TF) (Abadi et al., 2016) 2.1, an end-to-end open source platform for machine learning that uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across multiple computational devices including multi-core CPUs, general-purpose GPUs and custom-designed ASICs known as Tensor Processing Units (TPUs) (Abadi et al., 2016). TF facilitates the design of many machine learning models providing built-in functionalities such as convolution, pooling and fully-connected layers. However, TF does not provide computational neuroscience models, and to the best of our knowledge, there is no efficient implementation for SOMs using TF. The complete GPU-based source code for the SOM training, labeling and test is available in `https://github.com/lyes-khacef/GPU-SOM`.

A.2 CPU and GPU speedups

The SOMs of different sizes were trained for 10 epochs on MNIST database, i.e. 600,000 samples of 784 dimensions. The CPU mono-core implementation is based on NumPy (van der Walt, Colbert, and Varoquaux, 2011) and run on an Intel Core i9-9880H CPU (2.3 GHz $\times$ 16), while the GPU implementation is based on TF 2.1 (Abadi et al., 2016) and run on two different GPUs: Nvidia GeForce RTX 2080 with Max-Q, and Nvidia Tesla K80 GPU (2.3 GHz) freely available from Google Colab cloud service (Carneiro et al., 2018). Interestingly, the TF-based SOM can also run on the multiple cores of the CPU, providing a speed-up even without access to GPU. Figure A.1 shows that the time complexities of the CPU, TF-CPU and TF-GPU implementations are all linear. It is to note that the time complexity slope of the TF-CPU, TF-GPU GeForce and TF-GPU Tesla implementations changes at 1600 neurons, 400 neurons and 1024 neurons respectively, which is due to their different degrees of parallelism.

As shown in figure A.2 and reported in table A.1, we achieved a minimum speedup of 12$\times$ (22$\times$) and a maximum speedup of 161$\times$ (138$\times$) with the TF-GPU Tesla (TF-GPU GeForce) implementation, with an increasing speedup with respect to the number of neurons. Our GPU implementation is therefore scalable in simulation time with respect to the SOM size, which is an important aspect to accelerate the simulations and hyper-parameters exploration. For example, with a 32 $\times$ 32 SOM

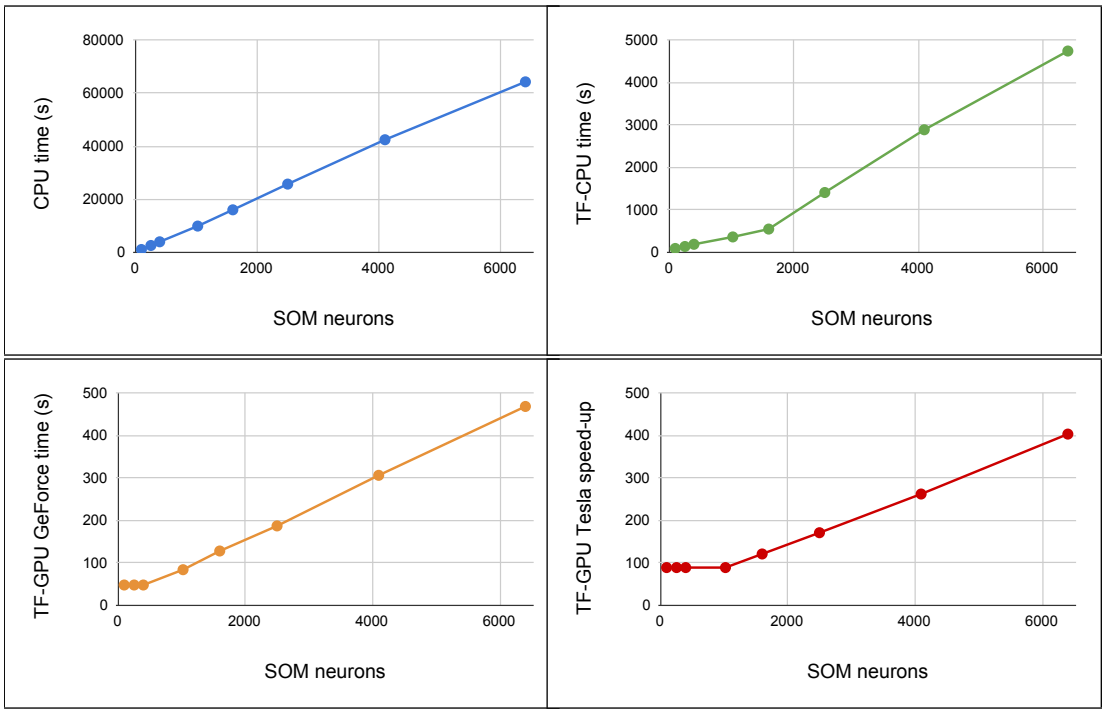

FIGURE A.1: SOM training speed on MNIST database for 10 epochs (i.e. 600,000 samples of 784 dimensions) vs. number of SOM neurons: (top-left) CPU (mono-core) implementation; (top-right) TF-CPU implementation; (bottom-left) TF-GPU GeForce implementation; (bottom-right) TF-GPU Tesla implementation.

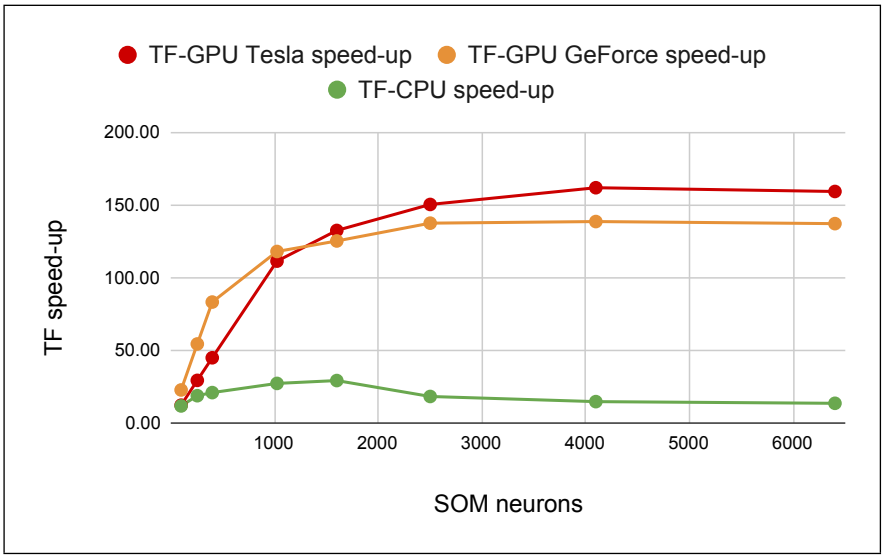

FIGURE A.2: TF-CPU and TF-GPU speed-ups compared to CPU.

(1024 neurons) trained on MNIST, we achieve a speedup of 118× going from 60 images/s on the Intel Core i9-9880H mono-core CPU to 7142 images/s on the Nvidia GeForce RTX 2080 GPU.

Between the two GPU devices, we find that the Nvidia GeForce is faster for small SOMs (< 1024 neurons), while the Nvidia Tesla is faster for bigger networks (> 1600 neurons). In addition, we achieved a minimum speedup of 11× times and a maximum speedup of 29× times with the TF-CPU implementation, which runs the 16

cores of the CPU. Nevertheless, the gap between the GPU and CPU speed-ups increases with the number of neurons, which is expected due to the highly parallel computation of the GPU hardware.

TABLE A.1: TF-CPU and TF-GPU minimum, maximum and average speed-ups compared to CPU.

Hardware	**Min speedup**	**Max speedup**	**Average speedup**
CPU (Intel Core i9-9880H x16)	11×	29×	19×
GPU (Nvidia GeForce RTX 2080)	22×	138×	**102**×
GPU (Nvidia Tesla K80)	12×	161×	**100**×

Recent works have tried an other approach using CUDA acceleration on Nvidia GPUs. They showed relative gains to CPU of 44× (Moraes et al., 2012), 47× (Gaval et al., 2019) and 67× (McConnell et al., 2012). Our implementation reaches an average gain of 19× in a multi-core Intel Core i9 CPU, 100× in a Nvidia Tesla GPU and 102× in a Nvidia GeForce GPU. A fair comparison is difficult since we do not target the same hardware, but the order of magnitude is comparable and our results are in the state of the art. Moreover, another advantage of our TF-based approach is the easy integration of the SOM layer into Keras (Chollet et al., 2015), a high-level neural networks API capable of running on top of TF with a focus on enabling fast experimentation. The TF SOM implementation was therefore a big step forward to overcome the technical limitation of the classical CPU implementation.

Appendix B

Multimodal databases details

B.1 Written/spoken digits database

The written and spoken digits database available in (Khacef, Rodriguez, and Miramond, 2019) is not a new database but a constructed database from existing ones. The objective is to provide a ready-to-use database for multimodal fusion.

B.1.1 Written digits

The Mixed National Institute of Standards and Technology (MNIST) database (LeCun and Cortes, 1998) is a database of 70000 handwritten digits (60000 for training and 10000 for test) proposed in 1998. Even if the database is quite old, it is still commonly used as a reference for training, testing and comparing various ML systems for classification tasks.

B.1.2 Spoken digits

Speech recognition is more and more present in human-computer interfaces like personal assistants (Google Assistant, Microsoft Cortana, Amazon Alexa, Apple Siri, etc.). The most commonly used acoustic feature in speech recognition is the Mel Frequency Cepstral Coefficients (MFCC) (Luque et al., 2018) (Darabkh et al., 2018) (Pan et al., 2018). MFCC was first proposed in (Mermelstein, 1976), which has since become the standard algorithm for representing speech features. It is a representation of the short-term power spectrum of a speech signal, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. We first extracted the MFCC features from the S-MNIST data, using the hyper-parameters from (Pan et al., 2018): framing window size = $50ms$ and frame shift size = $25ms$. Since the S-MNIST samples are approximately $1s$ long, we end up with 39 dimensions. However, it's not clear how many coefficients one has to take. Thus, we compared three methods: (Chapaneri, 2012) proposed to use 13 weighted MFCC coefficients, (Sainath and Parada, 2015) proposed to use 40 log-mel filterbank features, and (Pan et al., 2018) proposed to use 12 MFCC coefficients with an additional energy coefficient, making it 13 coefficients in total. The classification accuracy is respectively $61.79\% \pm 1.19$, $50.33\% \pm 0.59$ and $75.14\% \pm 0.57$. We therefore chose to work with a 39×13 dimensional features that are standardized (each feature is transformed by subtracting the mean value and dividing by the standard deviation of the training dataset, also called Z-score normalization) then min-max normalized (each feature is re-scaled to $0 - 1$ based on the minimum and maximum values of the training dataset).

B.2 DVS/EMG hand gestures database

The discrimination of human gestures using wearable solutions is extremely important as a supporting technique for assisted living, healthcare of the elderly and neuro-rehabilitation. More generally, the gestural interaction is a well-known technique that can be utilized in a vast array of applications (Yasen and Jusoh, 2019) such as sign language translation (Cheok, Omar, and Jaward, 2019), sports (Loss et al., 2012), Human-Robot Interaction (HRI) (Cicirelli et al., 2015) (Liu and Wang, 2018) and other applications related to Human-Machine Interaction (HMI) (Haria et al., 2017). Hand-gesture recognition systems also target medical applications, where they are detected via bioelectrical signals instead of vision. In particular, among the biomedical signals, ElectroMyoGraphy (EMG) is the most used for hand-gesture identification and for the design of prosthetic hand controllers (Benatti et al., 2015) (Chen et al., 2020) (Donati et al., 2019).

EMG measures the electrical signal resulting from muscle activation. The source of the signal is the motor neuron action potentials generated during the muscle contraction. Generally, EMG can be detected either directly with electrodes inserted in the muscle tissue, or indirectly with surface electrodes positioned above the skin (sEMG). For simplicity, we will refer to it as EMG. The EMG is more popular for its accessibility and non-invasive nature. However, the use of EMG to discriminate hand-gestures is a non-trivial task due to several physiological processes in the skeletal muscles underlying their generation. One way to overcome these limitations is to use a multimodal approach, combining EMG with recordings from other sensors. Therefore, we consider the complementary system comprising of a vision sensor and EMG measurements. Using EMG or camera systems separately presents some limitations, but their fusion has several advantages, in particular EMG-based classification can help in case of camera occlusion, whereas the vision classification provides an absolute measurement of hand state.

For this purpose, we proposed in (Ceolini et al., 2019b) a framework that allows the integration of EMG and vision data to perform sensor fusion based on supervised learning. The software version of the system was run on a mobile phone (Ceolini et al., 2019a) with a good performance on real-time multimodal classification. Afterwards, we proposed a neuromorphic hardware implementation (Ceolini et al., 2020) for energy-efficient processing on two target devices: Intel Loihi (Davies et al., 2018) and ODIN (Frenkel et al., 2019) + MorphIC (Frenkel, Legat, and Bol, 2019).

B.2.1 DVS sensor and pre-processing

The DVS is an event-based camera inspired by the mammalian retina (Lichtsteiner, Posch, and Delbruck, 2006), such that each pixel responds asynchronously to changes in brightness with the generation of events. Whenever the incoming illumination increases or decreases above a certain threshold, it generates a polarity spike event. The polarity corresponds to the sign of the change, "ON" polarity for increasing in light and "OFF" polarity for decreasing in light, as shown in figure B.1. Hence, only the active pixels transfer information and the static background is directly removed on hardware at the front-end. The event-based asynchronous nature of the DVS makes the sensor low power, low latency (down to 10 μs) and low-bandwidth, as the amount of data transmitted is very small. Each event (also called spike) is encoded using the Address Event Representation (AER) communication protocol (Deiss, Douglas, Whatley, et al., 1999) and is represented by the address of the pixel

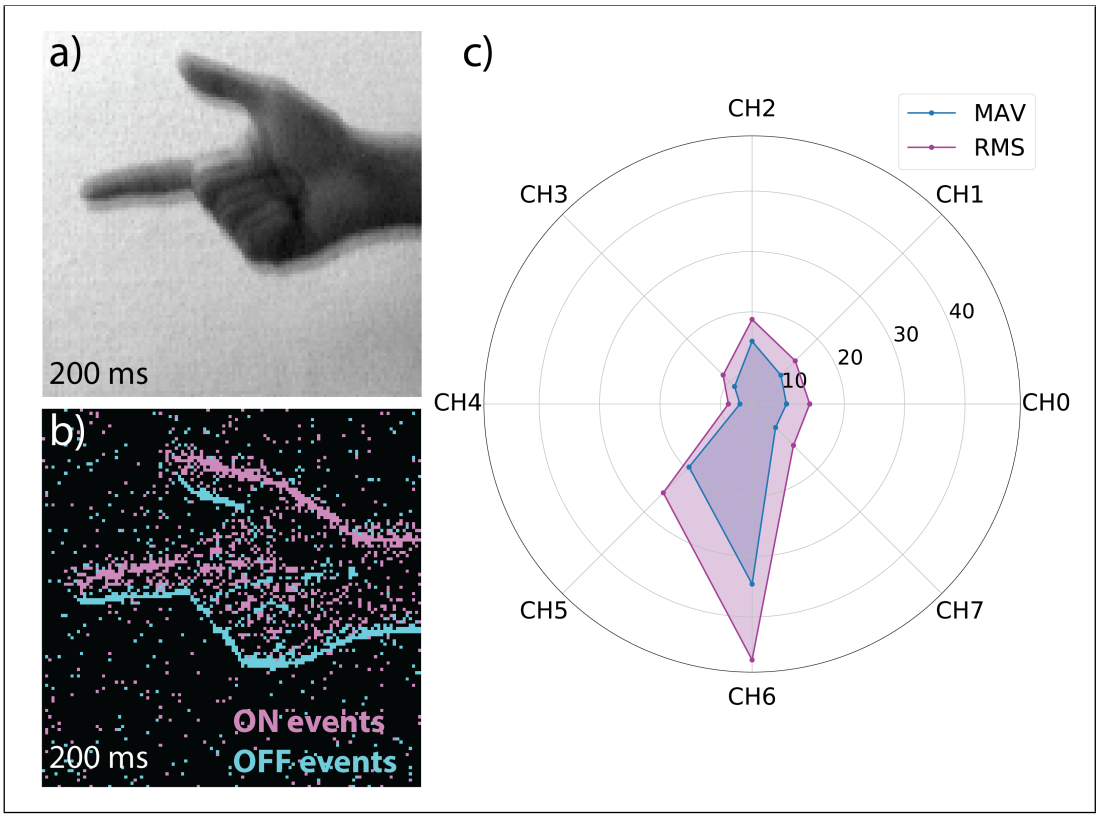

FIGURE B.1: Example of data from the DVS/EMG hand gestures dataset: (a) original frame; (b) DVS frame generated by the accumulation of events during 200*ms*; (c) EMG features for the 8 channels of the Myo.

(in x-y coordinates), the polarity (1 bit for the sign), and the timestamp (in microsecond resolution).

In order to use the DVS events for gesture classification with conventional algorithms, we need to turn the stream of events into frames, which we refer to as event frames. These frames are generated by accumulating the events occurring in a fixed time window of length *Tms*. DVS frames can so be synchronized with the EMG signal. In particular, we consider all the events within the time window (ignoring their polarity) and count how many events occur for each of the pixels separately. We then transform the event count frame into gray scale by min-max normalization. The event frames obtained from the DVS sensor have a resolution of 128x128 pixels. Since the region with the hand gestures does not fill the full frame, we extract a 60×60 pixels patch that allows us to significantly decrease the amount of computation needed during the visual feature extraction. This patch is extracted by detecting the hand in the frame with the zeroth order moment. This approach is reliable for event frames and has very low computational complexity.

B.2.2 EMG sensor and pre-processing

We collected the EMG corresponding to the hand gestures by using the Myo armband made by Thalmic Labs Inc. The Myo armband is a wearable device provided with eight equally spaced non-invasive EMG electrodes and a Bluetooth transmission module. The EMG 8 electrodes detect the signals from the forearm muscles activity and afterwards the acquired data is sent to an external electronic device, as

shown in figure B.1. The sampling rates for Myo data are fixed at $200Hz$ and the data is returned as a unitless $8-bit$ unsigned integer for each sensor representing "activation" and does not translate to mV.

We extracted two time domain features generally used in the literature (Phinyomark, N Khushaba, and Scheme, 2018), namely the Mean Absolute Value (MAV) and the Root Mean Square (RMS) shown in equation B.1. The MAV is the average of the muscles activation value and it is calculated by a stride-moving window. The RMS is represented as amplitude relating to a gestural force and muscular contraction. The two features are calculated across a window of 40 samples, corresponding to $200ms$.

$$MAV(x_c) = \frac{1}{T}\sum_{t=0}^{T}|x_c(t)| \qquad RMS(x_c) = \sqrt{\frac{1}{T}\sum_{t=0}^{T}x_c^2(t)} \tag{B.1}$$

It is to note that $x_c(t)$ is the signal in the time domain for the EMG channel with index c and T is the number of samples in the considered window, which was set to be $T = 40$ ($N = 200ms$) across this work. The features were calculated for each channel separately and the resulting values were concatenated in a vector $\mathbf{F}(n)$ described in equation B.2.

$$\mathbf{F}(n) = [F(x_1), ..., F(x_C)]^T \tag{B.2}$$

It is to note that $\mathbf{F}$ is MAV or RMS, n is the index of the window and C is the number of EMG channels. The final feature vector $\mathbf{E}(n)$ for window n is shown in equation B.3, it is used for the classification and is obtained by concatenating the two single feature vectors:

$$\mathbf{E}(n) = \left[\mathbf{MAV}(n)^T, \mathbf{RMS}(n)^T\right]^T \tag{B.3}$$

B.2.3 DVS/EMG dataset

The DVS/EMG hand gestures database available in (Ceolini et al., 2019) contains recordings from 21 subjects: 12 males and 9 females of age from 25 to 35. This first version contains therefore 6750 samples (5400 for training and 1350 for test). The structure is the following: each subject repeats 3 sessions, in each session the subject performs 5 hand gestures as shown in figure B.2: *Pinky*, *Elle*, *Yo*, *Index* and *Thumb*, repeated 5 times. Each single gesture recording lasts $2s$. The gestures are separated by a relaxing time of $1s$, in order to remove any residual activity from the previous gesture. Every recording is cut in 10 chunks of $200ms$ each, this duration was selected to match the requirements of a real-case scenario of low latency prosthesis control where there is a need for the classification and creation of the motor command within $250ms$ (Smith et al., 2011). The Myo records the superficial muscle activity at the middle forearm from 8 electrodes with a sampling rate of $200Hz$. During the recordings, the DVS was mounted on a random moving system to generate relative movement between the sensor and the subject hand. As shown in figure B.2, the hand stands static during the recording to avoid noise in the Myo sensor and the gestures are performed in front of a static white background.

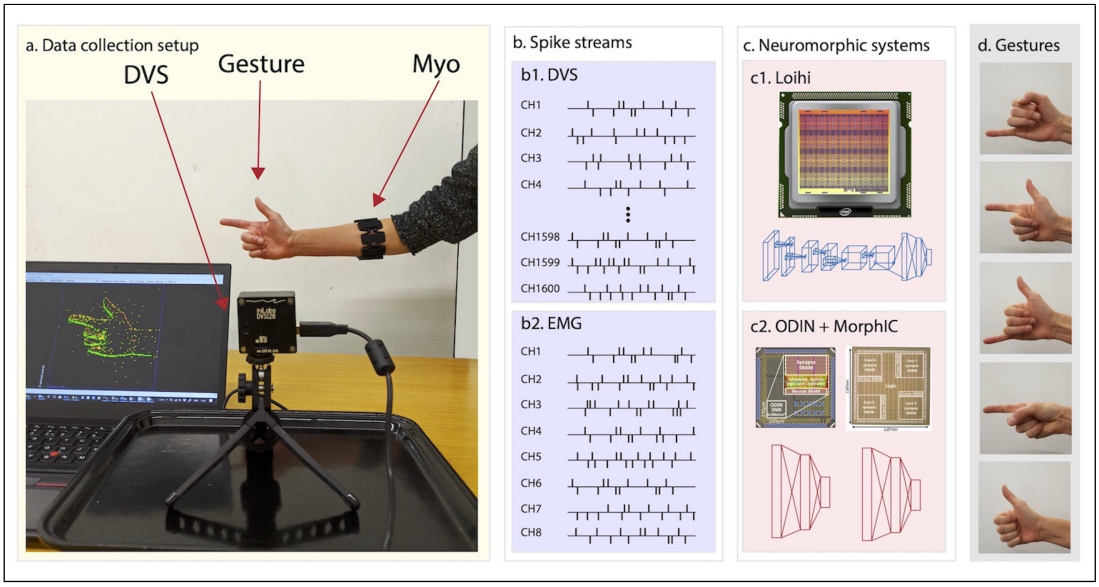

FIGURE B.2: System overview: (a) data collection setup featuring the DVS, the traditional camera and the subject wearing the EMG armband sensor; data streams of (b1) DVS and (b2) EMG transformed into spikes via the Sigma Delta modulation approach; the two neuromorphic systems namely (c1) Loihi and (c2) ODIN + MorphIC; (d) the hand gestures that the system is able to recognize in real time.

Bibliography

Abadi, Martín et al. (2016). "TensorFlow: A System for Large-Scale Machine Learning". In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. OSDI'16. Savannah, GA, USA: USENIX Association, 265–283. ISBN: 9781931971331.

Abderrahmane, Nassim, Edgar Lemaire, and Benoît Miramond (2020). "Design Space Exploration of Hardware Spiking Neurons for Embedded Artificial Intelligence". In: *Neural Networks* 121, pp. 366 –386. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.09.024. URL: http://www.sciencedirect.com/science/article/pii/S0893608019303041.

Alahakoon, D., S. K. Halgamuge, and B. Srinivasan (2000). "Dynamic self-organizing maps with controlled growth for knowledge discovery". In: *IEEE Transactions on Neural Networks* 11.3, pp. 601–614. DOI: 10.1109/72.846732.

Allman, Brian L, Leslie P. Keniston, and M. Alex Meredith (2009). "Not Just for Bimodal Neurons Anymore: The Contribution of Unimodal Neurons to Cortical Multisensory Processing". In: *Brain Topography* 21, pp. 157–167.

Althaus, N. and D. Mareschal (2013). "Modeling Cross-Modal Interactions in Early Word Learning". In: *IEEE Transactions on Autonomous Mental Development* 5.4, pp. 288–297.

Althaus, Nadja and Denis Mareschal (July 2014). "Labels Direct Infants' Attention to Commonalities during Novel Category Learning". In: *PLOS ONE* 9.7, pp. 1–10. DOI: 10.1371/journal.pone.0099670. URL: https://doi.org/10.1371/journal.pone.0099670.

Althaus, Nadja and Kim Plunkett (2015). "Timing matters: The impact of label synchrony on infant categorisation". In: *Cognition* 139, pp. 1 –9. ISSN: 0010-0277. DOI: https://doi.org/10.1016/j.cognition.2015.02.004. URL: http://www.sciencedirect.com/science/article/pii/S0010027715000281.

Althaus, Nadja and Gert Westermann (2016). "Labels constructively shape object categories in 10-month-old infants". In: *Journal of Experimental Child Psychology* 151. Interrelations Between Non-Linguistic and Linguistic Representations of Cognition and Action in Development, pp. 5 –17. ISSN: 0022-0965. DOI: https://doi.org/10.1016/j.jecp.2015.11.013. URL: http://www.sciencedirect.com/science/article/pii/S0022096515002891.

Andrade, Douglas Coimbra de et al. (2018). "A neural attention model for speech command recognition". In: *ArXiv* abs/1808.08929.

Asano, Michiko et al. (2015). "Sound symbolism scaffolds language development in preverbal infants". In: *Cortex* 63, pp. 196 –205. ISSN: 0010-9452. DOI: https://doi.org/10.1016/j.cortex.2014.08.025. URL: http://www.sciencedirect.com/science/article/pii/S0010945214002883.

Azcarraga, A. and A. Giacometti (1991). "A prototype-based incremental network model for classification tasks". In: *Proceedings of Neuro-Nmes*, 121–134.

Baldi, Pierre (2012). "Autoencoders, Unsupervised Learning, and Deep Architectures". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. Ed. by Isabelle Guyon et al. Vol. 27. Proceedings of Machine Learning Research.

Bellevue, Washington, USA: PMLR, pp. 37–49. URL: http://proceedings.mlr.press/v27/baldi12a.html.

Baltrusaitis, Tadas, Chaitanya Ahuja, and Louis-Philippe Morency (Feb. 2019). "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.2, 423–443. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607. URL: https://doi.org/10.1109/TPAMI.2018.2798607.

Barth, Daniel S. et al. (1995). "The spatiotemporal organization of auditory, visual, and auditory-visual evoked potentials in rat cortex". In: *Brain Research* 678, pp. 177–190.

Bauer, Roman (2013). "Self-construction and -configuration of functional neuronal networks". en. Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 21387, 2013. PhD thesis. Zürich: ETH Zurich. DOI: 10.3929/ethz-a-009988668.

Benatti, Simone et al. (2015). "A versatile embedded platform for EMG acquisition and gesture recognition". In: *IEEE transactions on biomedical circuits and systems* 9.5, pp. 620–630.

Bengio, Yoshua et al. (2006). "Greedy Layer-Wise Training of Deep Networks". In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Canada: MIT Press, 153–160.

Biagetti, Giorgio, Paolo Crippa, and Laura Falaschetti (Aug. 2018). "Classifier Level Fusion of Accelerometer and sEMG Signals for Automatic Fitness Activity Diarization". In: *Sensors* 18, p. 2850. DOI: 10.3390/s18092850.

Bichler, O. et al. (2012). "Visual Pattern Extraction Using Energy-Efficient "2-PCM Synapse" Neuromorphic Architecture". In: *IEEE Transactions on Electron Devices* 59.8, pp. 2206–2214.

Bizley, Jennifer K and Andrew J King (2008). "Visual–auditory spatial processing in auditory cortical neurons". In: *Brain Research* 1242, pp. 24–36.

Blakemore, Colin and GRAHAME COOPER (Oct. 1970). "Development of the Brain Depends on the Visual Environment". In: *Nature* 228, pp. 477–8. DOI: 10.1038/228477a0.

Blazewicz, Jacek et al. (Jan. 2000). *Handbook on Parallel and Distributed Processing*. Springer. ISBN: 978-3-642-08571-0. DOI: 10.1007/978-3-662-04303-5.

Braun, S. et al. (2019). "Attention-driven Multi-sensor Selection". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2019.8852396.

Bremermann, Hans J. (1994). "Self-Organization in Evolution, Immune Systems, Economics, Neural Nets, and Brains". In: *On Self-Organization: An Interdisciplinary Search for a Unifying Principle*. Ed. by R. K. Mishra, D. Maaß, and E. Zwierlein. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 5–34. ISBN: 978-3-642-45726-5. DOI: 10.1007/978-3-642-45726-5_2. URL: https://doi.org/10.1007/978-3-642-45726-5_2.

Buckner, Cameron and James Garson (2019). "Connectionism". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2019. Metaphysics Research Lab, Stanford University.

Budayan, Cenk, Irem Dikmen, and M. Talat Birgonul (2009). "Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping". In: *Expert Systems with Applications* 36.9, pp. 11772–11781. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2009.04.022.

Calvert, G. A., C. Spence, and B. E. Stein (2004). *The Handbook of Multisensory Processing*. English. USA United States: MIT Press.

Calvert, Gemma A. (Dec. 2001). "Crossmodal Processing in the Human Brain: Insights from Functional Neuroimaging Studies". In: *Cerebral Cortex* 11.12, pp. 1110–

1123. ISSN: 1047-3211. DOI: 10.1093/cercor/11.12.1110. eprint: https://academic.oup.com/cercor/article-pdf/11/12/1110/9751117/1101110.pdf. URL: https://doi.org/10.1093/cercor/11.12.1110.

Cappe, C., E. M. Rouiller, and P. Barone (May 2009). "Multisensory anatomical pathways." In: *Hearing Research* 258.1-2, pp. 28–36. DOI: 10.1016/j.heares.2009.04.017. URL: https://hal.archives-ouvertes.fr/hal-00435518.

Carneiro, Tiago et al. (2018). "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications". In: *IEEE Access* 6, pp. 61677–61685.

Castanedo, Federico (Oct. 2013). "A Review of Data Fusion Techniques". In: *TheScientificWorldJournal* 2013, p. 704504. DOI: 10.1155/2013/704504.

Ceolini, E. et al. (2019a). "Live Demostration: Sensor fusion using EMG and vision for hand gesture classification in mobile applications". In: *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–1. DOI: 10.1109/BIOCAS.2019.8919163.

— (2019b). "Sensor fusion using EMG and vision for hand gesture classification in mobile applications". In: *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4. DOI: 10.1109/BIOCAS.2019.8919210.

Ceolini, Enea et al. (May 2019). *EMG and Video Dataset for sensor fusion based hand gestures recognition*. DOI: 10.5281/zenodo.3228846. URL: https://doi.org/10.5281/zenodo.3228846.

Ceolini, Enea et al. (2020). "Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing". In: *Frontiers in Neuroscience* 14, p. 637. ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00637. URL: https://www.frontiersin.org/article/10.3389/fnins.2020.00637.

Chapaneri, Santosh (Feb. 2012). "Spoken Digits Recognition using Weighted MFCC and Improved Features for Dynamic Time Warping". In: *International Journal of Computer Applications* 40, pp. 6–12. DOI: 10.5120/5022-7167.

Charte, David et al. (2018). "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines". In: *Information Fusion* 44, pp. 78 –96. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.12.007. URL: http://www.sciencedirect.com/science/article/pii/S1566253517307844.

Chen, Chen et al. (2020). "Hand gesture recognition based on motor unit spike trains decoded from high-density electromyography". In: *Biomedical Signal Processing and Control* 55, p. 101637.

Chen, Wei-Yu et al. (2019). *A Closer Look at Few-shot Classification*. arXiv: 1904.04232 [cs.CV].

Cheok, Ming Jin, Zaid Omar, and Mohamed Hisham Jaward (2019). "A review of hand gesture and sign language recognition techniques". In: *International Journal of Machine Learning and Cybernetics* 10.1, pp. 131–153.

Chicca, E. et al. (2014). "Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems". In: *Proceedings of the IEEE* 102.9, pp. 1367–1388.

Cholet, S., H. Paugam-Moisy, and S. Regis (2019). "Bidirectional Associative Memory for Multimodal Fusion : a Depression Evaluation Case Study". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. DOI: 10.1109/IJCNN.2019.8852089.

Chollet, François et al. (2015). *Keras*. https://github.com/fchollet/keras.

Chum, Lovish et al. (2019). "Beyond Supervised Learning: A Computer Vision Perspective". In: *Journal of the Indian Institute of Science* 99.2, pp. 177–199. ISSN: 0019-4964. DOI: 10.1007/s41745-019-0099-3. URL: https://doi.org/10.1007/s41745-019-0099-3.

Cicirelli, Grazia et al. (2015). "A kinect-based gesture recognition approach for a natural human robot interface". In: *International Journal of Advanced Robotic Systems* 12.3, p. 22.

Clark, Andy (2001). "Reasons, Robots and the Extended Mind". In: *Mind & Language* 16.2, pp. 121–145. DOI: 10.1111/1468-0017.00162. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0017.00162. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0017.00162.

Cook, Matthew (Jan. 2004). "Universality in Elementary Cellular Automata". In: *Complex Systems* 15.

Damasio, Antonio R. (1989). "Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition". In: *Cognition* 33.1. Special Issue Neurobiology of Cognition, pp. 25 –62. ISSN: 0010-0277. DOI: https://doi.org/10.1016/0010-0277(89)90005-X. URL: http://www.sciencedirect.com/science/article/pii/001002778990005X.

— (1994). *Descartes' error : emotion, reason, and the human brain*. G.P. Putnam New York, xix, 312 p. : ISBN: 0399138943.

Damasio, Antonio R. and Hannah Damasio (1994). "Cortical Systems for Retrieval of Concrete Knowledge: The Convergence Zone Framework". In: *Large-Scale Neuronal Theories of the Brain*. Ed. by Christof Koch and J. Davis. MIT Press, pp. 61–74.

Darabkh, Khalid A. et al. (2018). "An efficient speech recognition system for arm-disabled students based on isolated words". In: *Comp. Applic. in Engineering Education* 26, pp. 285–301.

Davies, M. et al. (2018). "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning". In: *IEEE Micro* 38.1, pp. 82–99.

Dayan, Peter (1999). *Unsupervised Learning*. The MIT Encyclopedia of the Cognitive Sciences, edited by Robert A. Wilson and Frank C. Keil.

Debes, Christian et al. (May 2014). "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7. DOI: 10.1109/JSTARS.2014.2305441.

Dehner, Lisa R. et al. (Apr. 2004). "Cross-modal Circuitry Between Auditory and Somatosensory Areas of the Cat Anterior Ectosylvian Sulcal Cortex: A 'New' Inhibitory Form of Multisensory Convergence". In: *Cerebral Cortex* 14.4, pp. 387–403. ISSN: 1047-3211. DOI: 10.1093/cercor/bhg135. eprint: http://oup.prod.sis.lan/cercor/article-pdf/14/4/387/806490/bhg135.pdf. URL: https://doi.org/10.1093/cercor/bhg135.

Deiss, Stephen R, Rodney J Douglas, Adrian M Whatley, et al. (1999). "A pulse-coded communications infrastructure for neuromorphic systems". In: *Pulsed neural networks*, pp. 157–178.

Diehl, Peter and Matthew Cook (2015). "Unsupervised learning of digit recognition using spike-timing-dependent plasticity". In: *Frontiers in Computational Neuroscience* 9, p. 99. ISSN: 1662-5188. DOI: 10.3389/fncom.2015.00099.

Donati, Elisa et al. (2019). "Discrimination of EMG Signals Using a Neuromorphic Implementation of a Spiking Neural Network". In: *IEEE transactions on biomedical circuits and systems*.

Douglas, Rodney J. and Kevan A.C. Martin (2004). "NEURONAL CIRCUITS OF THE NEOCORTEX". In: *Annual Review of Neuroscience* 27.1. PMID: 15217339,

pp. 419–451. DOI: 10.1146/annurev.neuro.27.070203.144152. eprint: https://doi.org/10.1146/annurev.neuro.27.070203.144152. URL: https://doi.org/10.1146/annurev.neuro.27.070203.144152.

Doya, K. (1999). "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" In: *Neural Networks* 12.7, pp. 961 –974. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(99)00046-5. URL: http://www.sciencedirect.com/science/article/pii/S0893608099000465.

Droniou, Alain, Serena Ivaldi, and Olivier Sigaud (2015). "Deep unsupervised network for multimodal perception, representation and classification". In: *Robotics and Autonomous Systems* 71. Emerging Spatial Competences: From Machine Perception to Sensorimotor Intelligence, pp. 83 –98. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2014.11.005. URL: http://www.sciencedirect.com/science/article/pii/S0921889014002474.

"Early word-learning entails reference, not merely associations" (2009). In: *Trends in Cognitive Sciences* 13.6, pp. 258 –263. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2009.03.006. URL: http://www.sciencedirect.com/science/article/pii/S136466130900093X.

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

Edelman, Gerald and Joseph Gally (Dec. 2001). "Edelman GM, Gally JA. Degeneracy and complexity in biological systems. Proc Natl Acad Sci USA 98: 13763-13768". In: *Proceedings of the National Academy of Sciences of the United States of America* 98, pp. 13763–8. DOI: 10.1073/pnas.231499798.

— (Aug. 2013). "Reentry: A Key Mechanism for Integration of Brain Function". In: *Frontiers in integrative neuroscience* 7, p. 63. DOI: 10.3389/fnint.2013.00063.

Edelman, Gerald M. (1982). "Group selection and phasic reentrant signaling a theory of higher brain function". In: *The 4th Intensive Study Program Of The Neurosciences Research Program*.

— (1993). "Neural Darwinism: Selection and reentrant signaling in higher brain function". In: *Neuron* 10.2, pp. 115 –125. ISSN: 0896-6273. DOI: https://doi.org/10.1016/0896-6273(93)90304-A. URL: http://www.sciencedirect.com/science/article/pii/089662739390304A.

Escobar-Juárez, Esau et al. (2016). "A Self-Organized Internal Models Architecture for Coding Sensory–Motor Schemes". In: *Frontiers in Robotics and AI* 3, p. 22. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00022. URL: https://www.frontiersin.org/article/10.3389/frobt.2016.00022.

Falez, Pierre et al. (2019). "Unsupervised visual feature learning with spike-timing-dependent plasticity: How far are we from traditional feature learning approaches?" In: *Pattern Recognition* 93, pp. 418 –429. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.04.016. URL: http://www.sciencedirect.com/science/article/pii/S0031320319301621.

Fauth, Michael and Christian Tetzlaff (2016). "Opposing Effects of Neuronal Activity on Structural Plasticity". In: *Frontiers in Neuroanatomy* 10, p. 75. ISSN: 1662-5129. DOI: 10.3389/fnana.2016.00075. URL: https://www.frontiersin.org/article/10.3389/fnana.2016.00075.

Fiack, L., L. Rodriguez, and B. Miramond (2015). "Hardware design of a neural processing unit for bio-inspired computing". In: *2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS)*, pp. 1–4. DOI: 10.1109/NEWCAS.2015.7181997.

Fiack, Laurent, Nicolas Cuperlier, and Benoundefinedt Miramond (Dec. 2015). "Embedded and Real-Time Architecture for Bio-Inspired Vision-Based Robot Navigation". In: *J. Real-Time Image Process.* 10.4, 699–722. ISSN: 1861-8200. DOI: 10.1007/s11554-013-0391-9. URL: https://doi.org/10.1007/s11554-013-0391-9.

Fiebelkorn, Ian C., John J. Foxe, and Sophie Molholm (2010). "Dual mechanisms for the cross-sensory spread of attention: how much do learned associations matter?" In: *Cerebral cortex* 20 1, pp. 109–20.

Frenkel, Charlotte, Jean-Didier Legat, and David Bol (2019). "MorphIC: A 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning". In: *IEEE Transactions on Biomedical Circuits and Systems* 13.5, pp. 999–1010.

Frenkel, Charlotte et al. (2019). "A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS". In: *IEEE Transactions on Biomedical Circuits and Systems* 13.1, pp. 145–158.

Fritzke, Bernd (1994). "A Growing Neural Gas Network Learns Topologies". In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. NIPS'94. Denver, Colorado: MIT Press, 625–632.

Fukushima, Kunihiko (1975). "Cognitron: A self-organizing multilayered neural network". In: *Biological Cybernetics* 20, pp. 121–136.

— (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36, pp. 193–202.

Furber, S. B. et al. (2014). "The SpiNNaker Project". In: *Proceedings of the IEEE* 102.5, pp. 652–665.

Fyfe, Colin (Aug. 1997). "A Neural Network for PCA and Beyond". In: *Neural Process. Lett.* 6.1–2, 33–41. ISSN: 1370-4621. DOI: 10.1023/A:1009606706736. URL: https://doi.org/10.1023/A:1009606706736.

Gavval, Rohit et al. (2019). "CUDA-Self-Organizing feature map based visual sentiment analysis of bank customer complaints for Analytical CRM". In: *ArXiv* abs/1905.09598.

Gidaris, Spyros and Nikos Komodakis (2019). *Generating Classification Weights with GNN Denoising Autoencoders for Few-Shot Learning*. arXiv: 1905.01102 [cs.CV].

Gliga, T., A. Volein, and G. Csibra (2010). "Verbal labels modulate perceptual object processing in one-year-old children". In: *Journal of Cognitive Neuroscience* 22, pp. 2781–2789. DOI: 10.1162/jocn.2010.21427. URL: http://dx.doi.org/10.1162/jocn.2010.21427.

Gliozzi, Valentina et al. (2009). "Labels as Features (Not Names) for Infant Categorization: A Neurocomputational Approach". In: *Cognitive Science* 33.4, pp. 709–738. DOI: 10.1111/j.1551-6709.2009.01026.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1551-6709.2009.01026.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1551-6709.2009.01026.x.

González, Julio et al. (Sept. 2006). "Reading cinnamon activates olfactory brain regions". In: *NeuroImage* 32, pp. 906–12. DOI: 10.1016/j.neuroimage.2006.03.037.

Goodhill, Geoffrey J. and Harry G. Barrow (1994). "The Role of Weight Normalization in Competitive Learning". In: *Neural Computation* 6.2, pp. 255–269. DOI: 10.1162/neco.1994.6.2.255. eprint: https://doi.org/10.1162/neco.1994.6.2.255. URL: https://doi.org/10.1162/neco.1994.6.2.255.

Gu, L. and H. Li (2013). "Memory or Time: Performance Evaluation for Iterative Operation on Hadoop and Spark". In: *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on*

Embedded and Ubiquitous Computing, pp. 721–727. DOI: 10.1109/HPCC.and.EUC.2013.106.

Guo, Haodong et al. (Sept. 2014). "Activity recognition exploiting classifier level fusion of acceleration and physiological signals". In: *UbiComp 2014 - Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 63–66. DOI: 10.1145/2638728.2638777.

Halbach, M. and R. Hoffmann (2004). "Implementing cellular automata in FPGA logic". In: *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* Pp. 258–. DOI: 10.1109/IPDPS.2004.1303324.

Haria, Aashni et al. (2017). "Hand gesture recognition for human computer interaction". In: *Procedia computer science* 115, pp. 367–374.

Hawkins, Jeff and Sandra Blakeslee (2004). *On Intelligence*. USA: Times Books. ISBN: 0805074562.

Hazan, H. et al. (2018). "Unsupervised Learning with Self-Organizing Spiking Neural Networks". In: *2018 International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2018.8489673.

Hebb, Donald O. (June 1949). *The organization of behavior: A neuropsychological theory*. New York: Wiley. ISBN: 0-8058-4300-0.

Herculano-Houzel, Suzana (2009). "The human brain in numbers: a linearly scaled-up primate brain". In: *Frontiers in Human Neuroscience* 3, p. 31. ISSN: 1662-5161. DOI: 10.3389/neuro.09.031.2009. URL: https://www.frontiersin.org/article/10.3389/neuro.09.031.2009.

Heylighen, Francis and Carlos Gershenson (2003). "The Meaning of Self-Organization in Computing". In: *IEEE Intelligent Systems*, 72–75. URL: http://pcp.vub.ac.be/Papers/IEEE.Self-organization.pdf.

Hinton, Geoffrey E. (1989). "Connectionist learning procedures". In: *Artificial Intelligence* 40.1, pp. 185 –234. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(89)90049-0. URL: http://www.sciencedirect.com/science/article/pii/0004370289900490.

Hirsch, HV and DN Spinelli (1970). "Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats". In: *Science (New York, N.Y.)* 168.3933, 869—871. ISSN: 0036-8075. DOI: 10.1126/science.168.3933.869. URL: https://doi.org/10.1126/science.168.3933.869.

Hodgkin, A. L. and A. F. Huxley (1952). "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of Physiology* 117.4, pp. 500–544. DOI: 10.1113/jphysiol.1952.sp004764. eprint: https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1952.sp004764. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1952.sp004764.

Hoeks, Caroline et al. (Oct. 2011). "Prostate Cancer: Multiparametric MR Imaging for Detection, Localization, and Staging". In: *Radiology* 261, pp. 46–66. DOI: 10.1148/radiol.11091822.

Hopfield, J J (1982). "Neural networks and physical systems with emergent collective computational abilities". In: *Proceedings of the National Academy of Sciences* 79.8, pp. 2554–2558. ISSN: 0027-8424. DOI: 10.1073/pnas.79.8.2554. eprint: https://www.pnas.org/content/79/8/2554.full.pdf. URL: https://www.pnas.org/content/79/8/2554.

Horwitz, Barry and David Poeppel (Sept. 2002). "How can EEG/MEG and fMRI/PET data be combined?" In: *Human brain mapping* 17, pp. 1–3. DOI: 10.1002/hbm.10057.

Hoyer, Patrik O. (Dec. 2004). "Non-Negative Matrix Factorization with Sparseness Constraints". In: *J. Mach. Learn. Res.* 5, 1457–1469. ISSN: 1532-4435.

Hu, Yuqing, Vincent Gripon, and Stéphane Pateux (2020). *Exploiting Unsupervised Inputs for Accurate Few-Shot Classification*. arXiv: 2001.09849 [cs.LG].

Hubel, D. H. and T. N. Wiesel (1959). "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of Physiology* 148.3, pp. 574–591. DOI: 10.1113/jphysiol.1959.sp006308. eprint: https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1959.sp006308. URL: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1959.sp006308.

Huffman, Carl (2017). "Alcmaeon". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2017. Metaphysics Research Lab, Stanford University.

Indiveri, Giacomo et al. (2011). "Neuromorphic Silicon Neuron Circuits". In: *Frontiers in Neuroscience* 5, p. 73. ISSN: 1662-453X. DOI: 10.3389/fnins.2011.00073. URL: https://www.frontiersin.org/article/10.3389/fnins.2011.00073.

Izhikevich, E. M. (2003). "Simple model of spiking neurons". In: *IEEE Transactions on Neural Networks* 14.6, pp. 1569–1572.

Jayaratne, Madhura et al. (2018). "Bio-Inspired Multisensory Fusion for Autonomous Robots". In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3090–3095.

Ji, Xu, Andrea Vedaldi, and João F. Henriques (2018). "Invariant Information Clustering for Unsupervised Image Classification and Segmentation". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9864–9873.

Kemeny, John G. (1967). "Theory of Self-Reproducing Automata. John von Neumann. Edited by Arthur W. Burks. University of Illinois Press, Urbana, 1966. 408 pp., illus. 10". In: *Science* 157.3785, pp. 180–180. DOI: 10.1126/science.157.3785.180.

Khacef, L., N. Abderrahmane, and B. Miramond (2018). "Confronting machine-learning with neuroscience for neuromorphic architectures design". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. DOI: 10.1109/IJCNN.2018.8489241.

Khacef, L. et al. (2019). "Self-organizing neurons: toward brain-inspired unsupervised learning". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. DOI: 10.1109/IJCNN.2019.8852098.

Khacef, Lyes, Vincent Gripon, and Benoît Miramond (2020). "GPU-Based Self-Organizing Maps for Post-labeled Few-Shot Unsupervised Learning". In: *Neural Information Processing*. Ed. by Haiqin Yang et al. Cham: Springer International Publishing, pp. 404–416. ISBN: 978-3-030-63833-7.

Khacef, Lyes, Laurent Rodriguez, and Benoit Miramond (Oct. 2019). *Written and spoken digits database for multimodal learning*. Version 1.0. DOI: 10.5281/zenodo.3515935. URL: https://doi.org/10.5281/zenodo.3515935.

Khacef, Lyes, Laurent Rodriguez, and Benoît Miramond (2020a). "Improving Self-Organizing Maps with Unsupervised Feature Extraction". In: *Neural Information Processing*. Ed. by Haiqin Yang et al. Cham: Springer International Publishing, pp. 474–486. ISBN: 978-3-030-63833-7.

Khacef, Lyes, Laurent Rodriguez, and Benoit Miramond (2020b). "Method and System for multimodal classification based on brain-inspired unsupervised learning". In: *International Patent (Submitted)*.

Khacef, Lyes, Laurent Rodriguez, and Benoît Miramond (2020c). "Brain-Inspired Self-Organization with Cellular Neuromorphic Computing for Multimodal Unsupervised Learning". In: *Electronics* 9.10. ISSN: 2079-9292. DOI: 10.3390/electronics9101605. URL: https://www.mdpi.com/2079-9292/9/10/1605.

Khacef, Lyes et al. (2018). "Neuromorphic hardware as a self-organizing computing system". In: *2018 IJCNN Neuromorphic Hardware In Practice and Use workshop*. URL: https://arxiv.org/abs/1810.12640.

Kheradpisheh, Saeed Reza et al. (2018). "STDP-based spiking deep convolutional neural networks for object recognition". In: *Neural Networks* 99, pp. 56 –67. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2017.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0893608017302903.

Kiefer, Markus et al. (Dec. 2008). "The Sound of Concepts: Four Markers for a Link between Auditory and Conceptual Brain Systems". In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 28, pp. 12224–30. DOI: 10.1523/JNEUROSCI.3579-08.2008.

Kingma, Diederik P and Max Welling (2013). *Auto-Encoding Variational Bayes*. arXiv: 1312.6114 [stat.ML].

Kohlbrenner, Maximilian (2017). "Pre-Training CNNs Using Convolutional Autoencoders". In:

Kohonen, T. (1982). "Self-organized formation of topologically correct feature maps". In: *Biological Cybernetics*. DOI: 10.1007/BF00337288.

— (1990). "The self-organizing map". In: *Proceedings of the IEEE* 78.9, pp. 1464–1480. ISSN: 0018-9219. DOI: 10.1109/5.58325.

Kohonen, T., M. R. Schroeder, and T. S. Huang, eds. (2001). *Self-Organizing Maps*. 3rd. Berlin, Heidelberg: Springer-Verlag. ISBN: 3540679219.

Kohonen, T. et al. (1996). "Engineering applications of the self-organizing map". In: *Proceedings of the IEEE* 84.10, pp. 1358–1384. ISSN: 0018-9219. DOI: 10.1109/5.537105.

Kosko, B. (1988). "Bidirectional associative memories". In: *IEEE Transactions on Systems, Man, and Cybernetics* 18.1, pp. 49–60. DOI: 10.1109/21.87054.

Kriegstein, Katharina and Anne-Lise Giraud (Nov. 2006). "Implicit Multisensory Associations Influence Voice Recognition". In: *PLoS biology* 4, e326. DOI: 10.1371/journal.pbio.0040326.

Kromes, R. et al. (2019). "Energy consumption minimization on LoRaWAN sensor network by using an Artificial Neural Network based application". In: *2019 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6. DOI: 10.1109/SAS.2019.8705992.

Kyparissas, N. and A. Dollas (2019). "An FPGA-Based Architecture to Simulate Cellular Automata with Large Neighborhoods in Real Time". In: *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 95–99. DOI: 10.1109/FPL.2019.00024.

Lahat, D., T. Adali, and C. Jutten (2015). "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects". In: *Proceedings of the IEEE* 103.9, pp. 1449–1477. DOI: 10.1109/JPROC.2015.2460697.

Lallee, Stephane and Peter Ford Dominey (2013). "Multi-modal convergence maps: from body schema and self-representation to mental imagery". In: *Adaptive Behavior* 21.4, pp. 274–285. DOI: 10.1177/1059712313488423. eprint: https://doi.org/10.1177/1059712313488423. URL: https://doi.org/10.1177/1059712313488423.

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". English (US). In: *Nature* 521.7553, pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539.

LeCun, Yann and Corinna Cortes (1998). *MNIST handwritten digit database*. http://yann.lecun.com/exdb/m URL: http://yann.lecun.com/exdb/mnist/.

Lefort, Mathieu, Yann Boniface, and Bernard Girau (Aug. 2013). "SOMMA: Cortically Inspired Paradigms for Multimodal Processing". In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8. ISBN: 978-1-4673-6128-6. DOI: 10.1109/IJCNN.2013.6706959.

Liberman, Mark et al. (1991). *TI 46-Word LDC93S9*. Philadelphia. URL: https://catalog.ldc.upenn.edu/docs/LDC93S9/ti46.readme.html.

Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbruck (2006). "A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change". In: *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, pp. 2060–2069.

Liu, Hongyi and Lihui Wang (2018). "Gesture recognition for human-robot collaboration: A review". In: *International Journal of Industrial Ergonomics* 68, pp. 355–367.

Liu, Jinlu, Liang Song, and Yongqiang Qin (2019). *Prototype Rectification for Few-Shot Learning*. arXiv: 1911.10713 [cs.CV].

Liu, N., J. Wang, and Y. Gong (2015). "Deep Self-Organizing Map for visual classification". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. DOI: 10.1109/IJCNN.2015.7280357.

Loss, Jefferson Fagundes et al. (2012). "Evaluating the Electromyographical Signal during Symmetrical Load Lifting". In: *Applications of EMG in Clinical and Sports Medicine*, p. 1.

Luque, Amalia et al. (2018). "Non-sequential automatic classification of anuran sounds for the estimation of climate-change indicators". In: *Expert Systems with Applications* 95, pp. 248 –260. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.11.016. URL: http://www.sciencedirect.com/science/article/pii/S0957417417307662.

Maass, Wolfgang et al. (2019). "Brain Computation: A Computer Science Perspective". In: *Computing and Software Science: State of the Art and Perspectives*. Ed. by Bernhard Steffen and Gerhard Woeginger. Cham: Springer International Publishing, pp. 184–199. ISBN: 978-3-319-91908-9. DOI: 10.1007/978-3-319-91908-9_11. URL: https://doi.org/10.1007/978-3-319-91908-9_11.

Makhzani, Alireza et al. (2015). *Adversarial Autoencoders*. arXiv: 1511.05644 [cs.LG].

Man, Kingson et al. (2015). "Convergent and invariant object representations for sight, sound, and touch". In: *Human Brain Mapping* 36.9, pp. 3629–3640. DOI: 10.1002/hbm.22867. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.22867.

Mani, Nivedita and Kim Plunkett (2010). "In the Infant's Mind's Ear: Evidence for Implicit Naming in 18-Month-Olds". In: *Psychological Science* 21.7. PMID: 20519485, pp. 908–913. DOI: 10.1177/0956797610373371. eprint: https://doi.org/10.1177/0956797610373371. URL: https://doi.org/10.1177/0956797610373371.

Mansouri-Benssassi, Esma and Juan Ye (Apr. 2020). "Synch-Graph: Multisensory Emotion Recognition Through Neural Synchrony via Graph Convolutional Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34, pp. 1351–1358. DOI: 10.1609/aaai.v34i02.5491.

Markram, Henry et al. (1997). "Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs". In: *Science* 275.5297, pp. 213–215. ISSN: 0036-8075. DOI: 10.1126/science.275.5297.213. eprint: https://science.sciencemag.org/content/275/5297/213.full.pdf. URL: https://science.sciencemag.org/content/275/5297/213.

Marr, D. and T. Poggio (1976). *From Understanding Computation to Understanding Neural Circuitry*. Tech. rep. USA.

Marr, David (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. USA: Henry Holt and Co., Inc. ISBN: 0716715678.

Marsland, Stephen, Jonathan Shapiro, and Ulrich Nehmzow (Oct. 2002). "A Self-organising Network That Grows when Required". In: *Neural Netw.* 15.8-9, pp. 1041–1058. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(02)00078-3. URL: http://dx.doi.org/10.1016/S0893-6080(02)00078-3.

Martinetz, Thomas and Klaus Schulten (1991). "A Neural Gas Network learns Topologies". In:

Masci, Jonathan et al. (2011). "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". In: *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*. ICANN'11. Espoo, Finland: Springer-Verlag, 52–59. ISBN: 9783642217340.

McConnell, Sabine et al. (2012). "Scalability of Self-organizing Maps on a GPU cluster using OpenCL and CUDA". In: *Journal of Physics: Conference Series* 341, p. 012018. DOI: 10.1088/1742-6596/341/1/012018. URL: https://doi.org/10.1088.

Mcculloch, Warren and Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5, pp. 127–147.

Mead, C. (1990). "Neuromorphic electronic systems". In: *Proceedings of the IEEE* 78.10, pp. 1629–1636.

Mead, Carver (1989). *Analog VLSI and Neural Systems*. USA: Addison-Wesley Longman Publishing Co., Inc. ISBN: 0201059924.

Memisevic, Roland et al. (2010). "Gated Softmax Classification". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., pp. 1603–1611. URL: http://papers.nips.cc/paper/3895-gated-softmax-classification.pdf.

Mermelstein, P. (1976). "Distance measures for speech recognition, psychological and instrumental". In: *Pattern Recognition and Artificial Intelligence*, pp. 374–388. URL: https://ci.nii.ac.jp/naid/10026808024/en/.

Merolla, Paul A. et al. (2014). "A million spiking-neuron integrated circuit with a scalable communication network and interface". In: *Science* 345.6197, pp. 668–673. ISSN: 0036-8075. DOI: 10.1126/science.1254642. eprint: https://science.sciencemag.org/content/345/6197/668.full.pdf. URL: https://science.sciencemag.org/content/345/6197/668.

Metta, Giorgio et al. (2008). "The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition". In: *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*. PerMIS '08. Gaithersburg, Maryland: ACM, pp. 50–56. ISBN: 978-1-60558-293-1. DOI: 10.1145/1774674.1774683. URL: http://doi.acm.org/10.1145/1774674.1774683.

Meyer, Kaspar and Antonio Damasio (2009). "Convergence and divergence in a neural architecture for recognition and memory". In: *Trends in Neurosciences* 32.7, pp. 376 –382. ISSN: 0166-2236. DOI: https://doi.org/10.1016/j.tins.2009.04.002. URL: http://www.sciencedirect.com/science/article/pii/S0166223609000903.

Miikkulainen, Risto and Mark Moll (Sept. 1997). "Convergence-Zone Episodic Memory: Analysis and Simulations". In: *Neural networks : the official journal of the International Neural Network Society* 10, pp. 1017–1036. DOI: 10.1016/S0893-6080(97)00016-6.

Miikkulainen, Risto et al. (1997). *Self-Organization, Plasticity, and Low-level Visual Phenomena in a Laterally Connected Map Model of the Primary Visual Cortex*.

Moore, S. W. et al. (2012). "Bluehive - A field-programable custom computing machine for extreme-scale real-time neural network simulation". In: *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, pp. 133–140. DOI: 10.1109/FCCM.2012.32.

Moradi, S. et al. (2018). "A Scalable Multicore Architecture With Heterogeneous Memory Structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)". In: *IEEE Transactions on Biomedical Circuits and Systems* 12.1, pp. 106–122.

Moraes, F. C. et al. (2012). "Parallel High Dimensional Self Organizing Maps Using CUDA". In: *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 302–306. DOI: 10.1109/SBR-LARS.2012.56.

Moraes, Fernando et al. (Oct. 2004). "HERMES: An Infrastructure for Low Area Overhead Packet-Switching Networks on Chip". In: *Integr. VLSI J.* 38.1, 69–93. ISSN: 0167-9260. DOI: 10.1016/j.vlsi.2004.03.003. URL: https://doi.org/10.1016/j.vlsi.2004.03.003.

Morse, Anthony F. et al. (2015). "Posture Affects How Robots and Infants Map Words to Objects". In: *PloS one*.

Movellan, Javier R. (1995). "Visual Speech Recognition with Stochastic Networks". In: *Advances in Neural Information Processing Systems 7*. Ed. by G. Tesauro, D. S. Touretzky, and T. K. Leen. MIT Press, pp. 851–858. URL: http://papers.nips.cc/paper/993-visual-speech-recognition-with-stochastic-networks.pdf.

Mozafari, Milad et al. (2019). "SpykeTorch: Efficient Simulation of Convolutional Spiking Neural Networks With at Most One Spike per Neuron". In: *Frontiers in Neuroscience* 13, p. 625. ISSN: 1662-453X. DOI: 10.3389/fnins.2019.00625. URL: https://www.frontiersin.org/article/10.3389/fnins.2019.00625.

Nallaperuma, Dinithi et al. (2018). "Intelligent Detection of Driver Behavior Changes for Effective Coordination Between Autonomous and Human Driven Vehicles". In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3120–3125.

Nan Jiang et al. (2015). "An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning". In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2015.7280568.

"Neurons and glial cells". In: *Biology*. OpenStax College. URL: https://openstax.org/books/biology-2e/pages/35-1-neurons-and-glial-cells.

Ng, Andrew (2011). "Sparse autoencoder". In: *Lecture notes CS294A. Stanford University. Stanford, CA*. URL: https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf.

Oja, Erkki (1982). "Simplified neuron model as a principal component analyzer". In: *Journal of Mathematical Biology* 15, pp. 267–273.

Olshausen, Bruno A. and David J. Field (1997). "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision Research* 37.23, pp. 3311 –3325. ISSN: 0042-6989. DOI: https://doi.org/10.1016/S0042-6989(97)00169-7. URL: http://www.sciencedirect.com/science/article/pii/S0042698997001697.

O'Reilly, Randall C. and Yuko Munakata (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. 1st. Cambridge, MA, USA: MIT Press. ISBN: 0262650541.

Pan, S. J. and Q. Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359.

Pan, Z. et al. (2018). "An Event-Based Cochlear Filter Temporal Encoding Scheme for Speech Signals". In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489434.

Parisi, German I. et al. (2017). "Emergence of multimodal action representations from neural network self-organization". In: *Cognitive Systems Research* 43, pp. 208 –221. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2016.08.002. URL: http://www.sciencedirect.com/science/article/pii/S138904171630050X.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Peng, Liangying et al. (2016). "Hierarchical complex activity representation and recognition using topic model and classifier level fusion". In: *IEEE Transactions on Biomedical Engineering* 64.6, pp. 1369–1379.

Pfister, Sabina S. et al. (2018). "A Gene Regulatory Model of Cortical Neurogenesis". In: *bioRxiv*. DOI: 10.1101/394734. eprint: https://www.biorxiv.org/content/early/2018/08/17/394734.full.pdf. URL: https://www.biorxiv.org/content/early/2018/08/17/394734.

Phinyomark, Angkoon, Rami N Khushaba, and Erik Scheme (2018). "Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors". In: *Sensors* 18.5, p. 1615.

Pitti, A. et al. (2012). "Gain-field modulation mechanism in multimodal networks for spatial perception". In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 297–302. DOI: 10.1109/HUMANOIDS.2012.6651535.

Plunkett, Kim, Jon-Fan Hu, and Leslie B. Cohen (2008). "Labels can override perceptual categories in early infancy". In: *Cognition* 106.2, pp. 665 –681. ISSN: 0010-0277. DOI: https://doi.org/10.1016/j.cognition.2007.04.003. URL: http://www.sciencedirect.com/science/article/pii/S0010027707001084.

Pointeau, G., M. Petit, and P. F. Dominey (2014). "Successive Developmental Levels of Autobiographical Memory for Learning Through Social Interaction". In: *IEEE Transactions on Autonomous Mental Development* 6.3, pp. 200–212.

Querlioz, D. et al. (2013). "Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices". In: *IEEE Transactions on Nanotechnology* 12.3, pp. 288–295.

Ranganathan, Ananth and Zsolt Kira (2003). "Self-Organization in Artificial Intelligence and the Brain". In:

Rasamuel, M. et al. (2019). "Specialized visual sensor coupled to a dynamic neural field for embedded attentional process". In: *2019 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6.

Rathi, N. and K. Roy (2018). "STDP-Based Unsupervised Multimodal Learning With Cross-Modal Processing in Spiking Neural Network". In: *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11. ISSN: 2471-285X. DOI: 10.1109/TETCI.2018.2872014.

Ravi, Sachin and Hugo Larochelle (2017). "Optimization as a Model for Few-Shot Learning". In: *ICLR*.

Richter, Michael M. (1994). "Self-Organization, Artificial Intelligence and Connectionism". In: *On Self-Organization: An Interdisciplinary Search for a Unifying Principle*. Ed. by R. K. Mishra, D. Maaß, and E. Zwierlein. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 80–91. ISBN: 978-3-642-45726-5. DOI: 10.1007/978-3-642-45726-5_6. URL: https://doi.org/10.1007/978-3-642-45726-5_6.

rita, P. Bach y (1972). "Brain mechanisms in sensory substitution". In:

Rita, Paul Bach y and Stephen W. Kercel (2003). "Sensory substitution and the human–machine interface". In: *Trends in Cognitive Sciences* 7.12, pp. 541 –546. ISSN:

1364-6613. DOI: https://doi.org/10.1016/j.tics.2003.10.013. URL: http://www.sciencedirect.com/science/article/pii/S1364661303002900.

Rivet, B. et al. (2014). "Audiovisual Speech Source Separation: An overview of key methodologies". In: *IEEE Signal Processing Magazine* 31.3, pp. 125–134. DOI: 10.1109/MSP.2013.2296173.

Rodriguez, L., L. Khacef, and B. Miramond (2018). "A distributed cellular approach of large scale SOM models for hardware implementation". In: *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 250–255.

Rodriguez, Laurent (2015). "Définition d'un substrat computationnel bio-inspiré : déclinaison de propriétés de plasticité cérébrale dans les architectures de traitement auto-adaptatif". Thèse de doctorat dirigée par Granado, Bertrand STIC (sciences et technologies de l'information et de la communication) - Cergy Cergy-Pontoise 2015. PhD thesis. URL: http://www.theses.fr/2015CERG0765.

Rodriguez, Laurent, Laurent Fiack, and Benoît Miramond (2013). "A neural model for hardware plasticity in artificial vision systems". In: *Proceedings of the Conference on Design and Architectures for Signal and Image Processing*.

Rodriguez, Laurent, Lyes Khacef, and Benoit Miramond (2020). "Distributed Cellular Computing System and Method for neural-based Self-Organizing Maps". In: *International Patent (Submitted)*.

Roqui, Julian et al. (2020). "Estimation of Small Antenna Performance Using a Machine Learning Approach". In: *2020 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*.

Rougier, Nicolas and Yann Boniface (May 2011). "Dynamic Self-organising Map". In: *Neurocomputing, Elsevier* 74.11, pp. 1840–1847. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2010.06.034.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1988). "Learning Internal Representations by Error Propagation". In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 673–695. ISBN: 0262010976.

Russakovsky, Olga et al. (Dec. 2015). "ImageNet Large Scale Visual Recognition Challenge". In: *Int. J. Comput. Vision* 115.3, 211–252. ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y. URL: https://doi.org/10.1007/s11263-015-0816-y.

Rutishauser, Ueli and Rodney J. Douglas (2009). "State-Dependent Computation Using Coupled Recurrent Networks". In: *Neural Computation* 21.2, 478–509. ISSN: 1530-888X. DOI: 10.1162/neco.2008.03-08-734. URL: http://dx.doi.org/10.1162/neco.2008.03-08-734.

Sainath, Tara and Carolina Parada (2015). "Convolutional Neural Networks for Small-Footprint Keyword Spotting". In: *Interspeech*.

Sathian, K and A Zangaladze (2002). "Feeling with the mind's eye: contribution of visual cortex to tactile perception". In: *Behavioural Brain Research* 135.1, pp. 127–132. ISSN: 0166-4328. DOI: https://doi.org/10.1016/S0166-4328(02)00141-9. URL: http://www.sciencedirect.com/science/article/pii/S0166432802001419.

Schemmel, J. et al. (2010). "A wafer-scale neuromorphic hardware system for large-scale neural modeling". In: *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1947–1950.

Schroeder, Charles E. and John Foxe (Aug. 2005). "Multisensory contributions to low-level, 'unisensory' processing". English (US). In: *Current Opinion in Neurobiology* 15.4, pp. 454–458. ISSN: 0959-4388. DOI: 10.1016/j.conb.2005.06.008.

Schuman, C. D. et al. (2017). "A Survey of Neuromorphic Computing and Neural Networks in Hardware". In: *ArXiv e-prints*. arXiv: 1705.06963.

Shatz, Carla J. (1992). "How are specific connections formed between thalamus and cortex?" In: *Current Opinion in Neurobiology* 2.1, pp. 78 –82. ISSN: 0959-4388. DOI: https://doi.org/10.1016/0959-4388(92)90166-I. URL: http://www.sciencedirect.com/science/article/pii/095943889290166I.

Shivappa, S. T., M. M. Trivedi, and B. D. Rao (2010). "Audiovisual Information Fusion in Human–Computer Interfaces and Intelligent Environments: A Survey". In: *Proceedings of the IEEE* 98.10, pp. 1692–1715. DOI: 10.1109/JPROC.2010.2057231.

Silva, Daswin De et al. (2018). "Machine learning to support social media empowered patients in cancer care and cancer treatment decisions". In: *PloS one*.

Singer, W. (1990). "The formation of cooperative cell assemblies in the visual cortex". In: *Journal of Experimental Biology* 153.1, pp. 177–197. ISSN: 0022-0949. eprint: https://jeb.biologists.org/content/153/1/177.full.pdf. URL: https://jeb.biologists.org/content/153/1/177.

Smith, L. H. et al. (2011). "Determining the Optimal Window Length for Pattern Recognition-Based Myoelectric Control: Balancing the Competing Effects of Classification Error and Controller Delay". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.2, pp. 186–192. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2010.2100828.

Smith, Linda and Michael Gasser (2005). "The Development of Embodied Cognition: Six Lessons from Babies". In: *Artificial Life* 11.1-2, pp. 13–29. DOI: 10.1162/1064546053278973. eprint: https://doi.org/10.1162/1064546053278973. URL: https://doi.org/10.1162/1064546053278973.

Snell, Jake, Kevin Swersky, and Richard Zemel (2017). "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 4077–4087. URL: http://papers.nips.cc/paper/6996-prototypical-networks-for-few-shot-learning.pdf.

Song, Sen, Kenneth D. Miller, and L. F. Abbott (2000). *Competitive Hebbian Learning through Spike-Timing-Dependent Synaptic Plasticity*. DOI: 10.1038/78829.

Sousa, M. A. de Abreu de and E. Del-Moral-Hernandez (2017). "An FPGA distributed implementation model for embedded SOM with on-line learning". In: *2017 International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2017.7966351.

Sternberg, Robert J. (2000). *Handbook of Intelligence*. Cambridge University Press. DOI: 10.1017/CBO9780511807947.

Strukov, Dmitri et al. (2019). "Building brain-inspired computing". In: *Nature Communications*. URL: https://doi.org/10.1038/s41467-019-12521-x.

Tan, Ah-Hwee et al. (2019). "Self-organizing neural networks for universal learning and multimodal memory encoding". In: *Neural Networks*. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.08.020. URL: http://www.sciencedirect.com/science/article/pii/S0893608019302370.

Thrun, Sebastian and Lorien Pratt, eds. (2012). *Learning to Learn*. Springer Science & Business Media.

Turing, Alan (1948). "Intelligent machinery: National Physical Laboratory Report". In:

Turk, Matthew (2014). "Multimodal interaction: A review". In: *Pattern Recognition Letters* 36, pp. 189 –195. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2013.07.003. URL: http://www.sciencedirect.com/science/article/pii/S0167865513002584.

Upegui, Andres et al. (2018). "Pruning Self-Organizing Maps for Cellular Hardware Architectures". In: *12th NASA/ESA Conference on Adaptive Hardware and Systems*.

Ursino, Mauro, Cristiano Cuppini, and Elisa Magosso (2014). "Neurocomputational approaches to modelling multisensory integration in the brain: A review". In: *Neural networks : the official journal of the International Neural Network Society* 60, pp. 141–65.

Valiant, L. G. (Nov. 1984). "A Theory of the Learnable". In: *Commun. ACM* 27.11, 1134–1142. ISSN: 0001-0782. DOI: 10.1145/1968.1972. URL: https://doi.org/10.1145/1968.1972.

van der Walt, S., S. C. Colbert, and G. Varoquaux (2011). "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science Engineering* 13.2, pp. 22–30. ISSN: 1558-366X. DOI: 10.1109/MCSE.2011.37.

Vannel, F. et al. (2018). "SCALP: Self-configurable 3-D Cellular Adaptive Platform". In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1307–1312. DOI: 10.1109/SSCI.2018.8628794.

Varela, Francisco J., Eleanor. Rosch, and Evan. Thompson (1991). *The embodied mind: cognitive science and human experience*. MIT Press Cambridge, Mass, xx, 308 p. ; ISBN: 0262220423 0262720213.

Vavrecka, Michal and Igor Farkas (2013). "A Multimodal Connectionist Architecture for Unsupervised Grounding of Spatial Language". In: *Cognitive Computation* 6, pp. 101–112.

Vigneron, A. and J. Martinet (2020). "A critical survey of STDP in Spiking Neural Networks for Pattern Recognition". In: *2020 International Joint Conference on Neural Networks (IJCNN)*.

Vincent, Pascal et al. (Dec. 2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *J. Mach. Learn. Res.* 11, 3371–3408. ISSN: 1532-4435.

Vinyals, Oriol et al. (2016). "Matching Networks for One Shot Learning". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 3630–3638. URL: http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf.

von Neumann, J. (1993). "First draft of a report on the EDVAC". In: *IEEE Annals of the History of Computing* 15.4, pp. 27–75.

Walsh, D. and P. Dudek (2012). "A compact FPGA implementation of a bit-serial SIMD cellular processor array". In: *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1–6. DOI: 10.1109/CNNA.2012.6331450.

Wang, Runchun and André van Schaik (2018). "Breaking Liebig's Law: An Advanced Multipurpose Neuromorphic Engine". In: *Front. Neurosci.*

Wang, Runchun, Chetan Singh Thakur, and André van Schaik (2018). "An FPGA-Based Massively Parallel Neuromorphic Cortex Simulator". In: *Front. Neurosci.*

Wang, Yaqing et al. (2019). *Generalizing from a Few Examples: A Survey on Few-Shot Learning*. arXiv: 1904.05046 [cs.LG].

Warden, Pete (2018). "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". In: *ArXiv* abs/1804.03209.

Waxman, Sandra and Irena Braun (May 2005). "Consistent (but not variable) names as invitations to form object categories: New evidence from 12-month-old infants". In: *Cognition* 95, B59–68. DOI: 10.1016/j.cognition.2004.09.003.

Waxman, Sandra R. and Dana B. Markow (1995). "Words as Invitations to Form Categories: Evidence from 12- to 13-Month-Old Infants". In: *Cognitive Psychology* 29.3, pp. 257 –302. ISSN: 0010-0285. DOI: https://doi.org/10.1006/cogp.

1995.1016. URL: http://www.sciencedirect.com/science/article/pii/S001002858571016X.

Werker, Janet et al. (Dec. 1998). "Acquisition of Word–Object Associations by 14-Month-Old Infants". In: *Developmental psychology* 34, pp. 1289–309. DOI: 10.1037/0012-1649.34.6.1289.

Westermann, Gert and Denis Mareschal (2014). "From perceptual to language-mediated categorization". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 369.1634, p. 20120391. DOI: 10.1098/rstb.2012.0391. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rstb.2012.0391. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2012.0391.

Wickramasinghe, C. S., K. Amarasinghe, and M. Manic (2017). "Parallalizable deep self-organizing maps for image classification". In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. DOI: 10.1109/SSCI.2017.8285443.

Wolfram, Stephen (1984a). "Cellular automata as models of complexity". In: *Nature* 311, pp. 419–424.

— (1984b). "Universality and complexity in cellular automata". In: *Physica D: Nonlinear Phenomena* 10.1, pp. 1 –35. ISSN: 0167-2789. DOI: https://doi.org/10.1016/0167-2789(84)90245-8. URL: http://www.sciencedirect.com/science/article/pii/0167278984902458.

Wolpert, D.M. and M. Kawato (1998). "Multiple paired forward and inverse models for motor control". In: *Neural Networks* 11.7, pp. 1317 –1329. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(98)00066-5. URL: http://www.sciencedirect.com/science/article/pii/S0893608098000665.

Yasen, Mais and Shaidah Jusoh (2019). "A systematic review on hand gesture recognition techniques, challenges and applications". In: *PeerJ Computer Science* 5, e218.

Zador, Antony (2019). "A critique of pure learning and what artificial neural networks can learn from animal brains". In: *Nature Communications*. URL: https://doi.org/10.1038/s41467-019-11786-6.

Zagoruyko, Sergey and Nikos Komodakis (2016). *Wide Residual Networks*. arXiv: 1605.07146 [cs.CV].

Zaharia, Matei et al. (Oct. 2016). "Apache Spark: A Unified Engine for Big Data Processing". In: *Commun. ACM* 59.11, pp. 56–65. ISSN: 0001-0782. DOI: 10.1145/2934664. URL: http://doi.acm.org/10.1145/2934664.

Zahra, Omar and David Navarro-Alarcon (2019). "A Self-organizing Network with Varying Density Structure for Characterizing Sensorimotor Transformations in Robotic Systems". In: *Towards Autonomous Robotic Systems*. Ed. by Kaspar Althoefer, Jelizaveta Konstantinova, and Ketao Zhang. Cham: Springer International Publishing, pp. 167–178. ISBN: 978-3-030-25332-5.

Zeiler, Matthew D. (2012). "ADADELTA: An Adaptive Learning Rate Method". In: *CoRR* abs/1212.5701.

Zhang, Y., Z. Wang, and J. Du (2019). "Deep Fusion: An Attention Guided Factorized Bilinear Pooling for Audio-video Emotion Recognition". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2019.8851942.

Zhao, D. and Y. Zeng (2019). "Dynamic Fusion of Convolutional Features based on Spatial and Temporal Attention for Visual Tracking". In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2019.8852301.

Zhao, Junbo et al. (2015). *Stacked What-Where Auto-encoders*. arXiv: 1506.02351 [stat.ML].