


Universidade de São Paulo
Instituto de Física

Análise das representações internas em redes neurais de spike

João Henrique de Sant'Ana



Orientador: Prof. Dr. Nestor Caticha

Dissertação de mestrado apresentada ao Instituto de Física da Universidade de São Paulo, como requisito parcial para a obtenção do título de Mestre em Ciências.

Banca Examinadora:

Prof. Dr. Nestor Caticha - Orientador (IF-USP)

Prof. Dr. Antonio Carlos Roque da Silva Filho - (FFCLRP/USP)

Prof. Dr. Mauro Copelli Lopes da Silva - (UFPE)

São Paulo
2024

FICHA CATALOGRÁFICA
Preparada pelo Serviço de Biblioteca e Informação
do Instituto de Física da Universidade de São Paulo

Sant'Ana, João Henrique de

Análise das representações internas em redes neurais de spike /
Internal representations analysis in spiking neural networks. São Paulo,
2024.

Dissertação (Mestrado) - Universidade de São Paulo, Instituto de
Física. Depto. de Física Geral.

Orientador: Prof. Dr. Nestor Felipe Caticha Alfonso

Área de Concentração: Física Geral

Unitermos: 1. Redes neurais; 2. Representações internas; 3. Cérebro
crítico; 4. Índices de complexidade.

USP/IF/SBI-020/2024

University of São Paulo
Physics Institute

Internal representations analysis in spiking neural networks

João Henrique de Sant'Ana

Supervisor: Prof. Dr. Nestor Caticha

Dissertation submitted to the Physics Institute of the
University of São Paulo in partial fulfillment of the
requirements for the degree of Master of Science.

Examining Committee:

Prof. Dr. Nestor Felipe Caticha Alfonso - Supervisor (IF-USP)

Prof. Dr. Antonio Carlos Roque da Silva Filho - (FFCLRP/USP)

Prof. Dr. Mauro Copelli Lopes da Silva - (UFPE)

São Paulo
2024

Acknowledgements

I would like to express my deep gratitude to my family, especially to my parents Ivan and Aparecida, and to my siblings Giovanna and Bruno, for the support and assistance that has been with me since the moment I decided to enter a public university. Additionally, I would like to thank my partner Ingrid for being by my side, supporting me, and facing with me the greatest adversities to keep our relationship strong during such a delicate time. Also, I would like to express profound gratitude to my deceased grandparents, Nelson, Diva, and Manoela, who have always taken care of me and who would undoubtedly be very proud. The support of all of you was crucial for the completion of this work.

I would like to thank my friends and research colleagues who, in some way, contributed to every piece of this work: Lucas Hideki, João Victor, Matheus Marsiglio, Murillo de Godoy, Guilherme Ferrari, Rodrigo Veiga, Evanildo Junior, Fernando Silva, Willy Kroschinsky, Gustavo Menesse, and especially Otávio Citton with whom I had brilliant conversations.

I would like to express gratitude to my advisor Nestor Caticha for the opportunity to be his student. For showing me that Physics is not just about calculations and that certainly the answer to a phenomenon lies in the effort to formulate the right question.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Agradecimentos

Eu gostaria de expressar minha profunda gratidão à minha família, especialmente aos meus pais Ivan e Aparecida, e aos meus irmãos Giovanna e Bruno, pelo apoio e suporte que vem desde o momento em que eu decidi entrar em uma universidade pública. Além disso, gostaria de agradecer à minha companheira Ingrid por estar ao meu lado, apoiando-me e por ter enfrentado comigo as maiores adversidades para manter nosso relacionamento em pé durante um momento tão delicado. Também, eu gostaria de agradecer profundamente aos meus avós falecidos, Nelson, Diva e Manoela, que sempre estiveram cuidando de mim e que, sem dúvida, estariam muito orgulhosos. O apoio de todos vocês foi crucial para conclusão desse trabalho.

Eu gostaria de agradecer aos meus amigos e amigos de pesquisa, que de alguma maneira, contribuíram para cada pedaço deste trabalho: Lucas Hideki, João Victor, Matheus Marsiglio, Murillo de Godoy, Guilherme Ferrari, Rodrigo Veiga, Evanildo Junior, Fernando Silva, Willy Kroschinsky, Gustavo Menesse, e em especial ao Otávio Citton com quem eu tive conversas brilhantes.

Eu gostaria de agradecer ao meu orientador Nestor Caticha pela oportunidade de ser seu aluno. Por ter me mostrado que a Física não se trata apenas de contas e que certamente a resposta para um fenômeno está no esforço de formular a pergunta correta.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

The aim of science is not things themselves, as the dogmatists in their simplicity imagine, but the relations among things; outside these relations there is no reality knowable.

HENRI POINCARÉ, *Science and Hypothesis* (1905)

Abstract

Optimal information processing in peripheral sensory systems has been associated in several examples to the signature of a critical or near critical state. Furthermore, cortical systems have also been described to be in a critical state in both wake and anesthetized experimental models, both in vitro and in vivo. Using a methodology inspired in the biological setup, we investigate whether a similar signature characterizes the internal representations (IR) of a multilayer (deep) spiking artificial neural network performing a recognition task. The increase of the characteristic time of the decay of the correlation of fluctuations of the IR, found when the network input changes are indications of a broad-tailed distribution of IR fluctuations. The broad tails are present even when the network is not yet capable of performing the classification tasks, either due to partial training or to the effect of a low dose of anesthesia in a simple model. This is a signature only for significant changes involving labeled inputs. However, we don't find enough evidence of power law distributions of avalanche size and duration. Finally, the perturbational complexity index (PCI) of IR was measured. The PCI distinguished levels of training time and also effect of anesthesia leading to a characterization of the spatio-temporal pattern of activity in the internal layers.

Keywords: Neural Networks; Internal Representations; Critical Brain, Complex index.

Resumo

O processamento de informações ótimo nos sistemas sensoriais periféricos tem sido associado em diversos exemplos à assinatura de um estado crítico ou próximo ao crítico. Além disso, sistemas corticais também foram descritos como estando em um estado crítico tanto em modelos experimentais em estados de vigília ou anestesiados, tanto *in vitro* quanto *in vivo*. Utilizando uma metodologia inspirada nos sistemas biológicos, nós investigamos se assinaturas similares caracterizam as representações internas (IR) de uma rede neural artificial de spike multi-camada (profunda) realizando uma tarefa de reconhecimento. O aumento do tempo característico do decaimento da correlação das flutuações da IR, encontrado quando o input da rede muda, são indicações de distribuições de caudas longas nas flutuações da IR. As flutuações estão presentes mesmo quando a rede não é ainda capaz de realizar uma tarefa de classificação, seja devido ao treinamento parcial ou ao efeito de uma baixa dose de anestesia em um modelo simples. Esta é uma assinatura apenas para mudanças significativas envolvendo inputs rotulados. Porém, nós não encontramos evidências suficientes de distribuições de lei de potência do tamanho e duração das avalanches. Por último, o índice de complexidade perturbacional (PCI) da IR foi medido. O PCI distinguiu níveis de tempo de treinamento e também efeito de anestesia resultando em uma caracterização do padrão espaço-temporal da atividade nas camadas internas.

Palavras-chave: Redes Neurais; Representações Internas; Cérebro Crítico; Índices de Complexidade.

Contents

1	Introduction	7
1.1	Artificial Intelligence	7
1.2	Critical Phenomena	11
1.2.1	Critical Phenomena and the Brain	13
1.3	Complexity Index	16
1.4	Materials and Methods	16
2	Spiking Neural Networks	18
2.1	A Brief Introduction	18
2.2	Training a Spiking Neural Network	20
2.2.1	Learning Problem	20
2.2.2	Neural Modeling	23
2.2.3	Learning Algorithm	27
2.2.4	Implementing	32
3	Internal Representations Analysis	38
3.1	Label Transition	40
3.2	Learning Process	47
3.3	Anesthetized Perception	48
3.4	Random Inputs	50
3.5	Avalanches	53
3.6	Perturbational Complexity Index	55
3.6.1	Complexity of Learning Process	58
3.6.2	Complexity of Perception	59
4	Conclusion	61
	Reference	64
	Appendices	71
A	Article	72

Chapter 1

Introduction

The behavior of large and complex aggregates of elementary particles, it turns out, is not to be understood in terms of a simple extrapolation of a few particles. Instead, at each level of complexity entirely new properties appear, and the understanding of the new behaviors requires research which I think is as fundamental in its nature as any other.

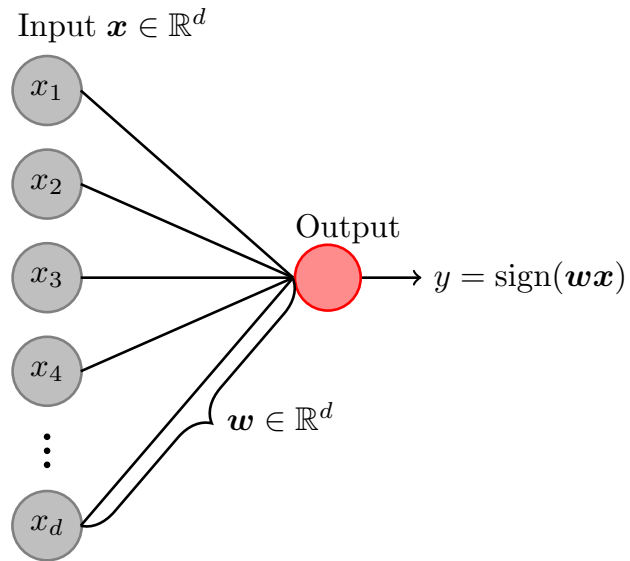
PHILIP WARREN ANDERSON, *More is Different* (1972)

1.1 Artificial Intelligence

Although there is no precise definition with regard to what *Artificial Intelligence* (AI) means, not to mention *intelligence*, at least we can partially describe what AI tries to do and how. For our purposes, it consists of the art of building biologically inspired machines to solve problems that in principle are intuitive tasks for living beings, but extremely complicated to put in simple mathematical rules. For instance, despite the computational power of programmable digital computers have, even a resourceful software implementation of procedural rules, fails significantly in pattern recognition, including image and speech recognition, where the brain handles it straightforwardly. It is in this sense that AI intrinsically involves and takes inspiration from the theory of cognitive science, such as learning and knowledge acquisition as its central approach. While this idea may seem new, due to its modern technological applications, the notion of a *thinking machine* has been in the imaginary since the Greek mythology.

The pioneers of *cybernetics* Warren McCulloch and Walter Pitts [1] introduced the first artificial neural modeling of the *all-or-none* behavior of nervous cells units, with the aim of computing logic operations by making circuits composed by them. From then on, a great effort appeared for the construction of a learning theory. Psychologist Donald Hebb in the seminal 1949 work [2] suggested that the learning process is expressed by an adaptation of synaptic connections. Later, the earliest artificial neural networks (ANN) that learn from examples appeared, such as the first feedforward network, called the Perceptron, invented in 1957 by Frank Rosenblatt [3]. The (multi-layer) perceptron is composed of layers of interacting McCulloch-Pitts neurons or single layer perceptrons, shown in figure 1.1. Each unit is defined by a weight vector $\mathbf{w} \in \mathbb{R}^d$, where d is the dimension of the input and by one output neuron described by a sign function. Among Rosenblatt's

Figure 1.1: Rosenblatt’s Perceptron. The perceptron is defined by a weight vector $\mathbf{w} \in \mathbb{R}^d$, where d is the dimension of input and by an output neuron described by a signal function, also referred by a McCulloch-Pitts neuron.



important contributions is the introduction of the learning algorithm, characterized by the reorganization of learnable parameters \mathbf{w} from the information in labeled examples of a training set, with the purpose of accomplishing the successful binary classification of arrays of feature inputs. He proved the theorem that the algorithm is able to find a solution \mathbf{w} that correctly classifies the examples of the training set, should a solution exist. This was undoubtedly an intellectual milestone in the development of AI machines. It is no wonder that a human created system that learns from experience without having previously established any specific rule gave substantial impulse for the AI future. Nevertheless, it was at this moment of optimism that Marvin Minsk and Seymour Papert, in their book entitled *Perceptrons* [4], demonstrated that there are some cases where the solution does not exist. In particular, the perceptron can not be applied to the XOR function. They also extended this result incorrectly to the multilayer perceptron, arguing as a proof, that they did not know how to analyze this type of model. Unexpectedly, it culminated in discouraging of financial support for this type of paradigm in AI. The next important stimulus for the study of ANN came from the discovery [5] and the rediscovery [6] of algorithms to train multilayer networks that can solve general classification problems.

In the present day, variations on the perceptron are the building blocks of Deep Learning Neural Networks, renowned for a remarkable success in applications across multiplies fields. Such networks encompass, in the midst of many, Convolution Neural Networks for Vision [7] and Transformers for Language [8]. Therefore, the perceptron is a fundamental piece for the construction of powerful machines whose outstanding performance exceeds human beings in many tasks. For example, one of the many breakthrough examples can be found in the domain of Board Gamer, when the first AI based on deep reinforcement learning beat a human world champion at Go in 2016 [9]. This seemingly unimportant task led to the important development of AlphaFold that largely improved the prediction of the structure of proteins from the DNA sequence information [10].

Indeed, the undergoing revolution brought by the many applications of AI in general and Deep Learning in particular, currently underway, will change many aspects of society in the near future. Despite the technological advances, the understanding of the reasons behind the performance of a deep neural network are far from satisfactory. Due to their importance, this is a relevant issue. The study of the Statistical Mechanics of

Neural Networks has a history that spans several decades and topics [11]. The formal concepts of Statistical Learning theory as the VC (Vapnik–Chervonenkis) dimension and the Rademacher complexity have been the canonical measure for understanding of the generalization error¹ in neural networks [12, 13]. Yet, there is a huge gap in having a complete description of the learning processes in large neural networks.

Before delving deeper into this intriguing question, let’s define concretely what an ANN is and what its relation is to the brain. Quoting the words of Teuvo Kohonen from reference [14],

“Artificial neural networks” are massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous systems do.

By means of this description, it is possible to identify characteristics partially resembling the brain. Some of them are: parallel information processing, composition by many interconnected excitable units, interaction with an environment and, crucially, execution of a meaningful task. As a whole, these constitute one of the fundamental aspects of the information processing dynamics in the brain computational architecture. It is important to note the following point, on one hand, regardless of its inspiration and similarities in biological neural networks, ANNs are far from being accurate models for brain functioning. On the other hand, this can be a useful tool for the understanding of the neurophysiological processes (perception, memory, language), since in the controlled setup of ANN models, questions can be related to the analysis of experimental data from biological systems in a wide range of empirical conditions. This approach is loosely known by *connectionism*, and it was developed in the 70-80’s by many authors, systematically presented in the book *Parallel Distributed Processing* [6].

The difficulty in building a fully developed theory for capabilities of deep learning systems stems mainly from two factors (1) neural networks are universal approximators [15, 16] and (2) the nature of acquiring artificial intelligence is a NP-complete optimization problem [17]. In essence, neural networks are devices capable of approximating *any* arbitrary continuous function. As a result, having a learning algorithm, neural networks have the ability to correctly map the features of a relevant task, where initially it is described by an *unknown* function, at least for the examples of the training set. It is at this point that the training time takes its important place. The main obstacle to attain a good performance in the learning from data process is in the computational complexity. Even for shallow neural networks in the worst-case scenario, the optimization of all neural network’s parameters is definitely a non-efficient algorithm. And to make things worse, the training techniques become even more challenging in front of the heavily parameterized models, since that large numbers of parameters increase complexity. It was only at the beginning of this millennium that the training of deep nets became practicable. Some reasons were the low cost and the general purpose implementation of GPU (Graphics processing unit) devices, unlike the CPU (Central Processing Unit), is a parallel computation circuit.

Thus, in short, deep networks are hard to interpret. Each breakthrough within the scope of applications makes the learning problem in such models additionally fascinating. As Leo Breiman pointed out in [18],

¹The accuracy in samples outside of training set.

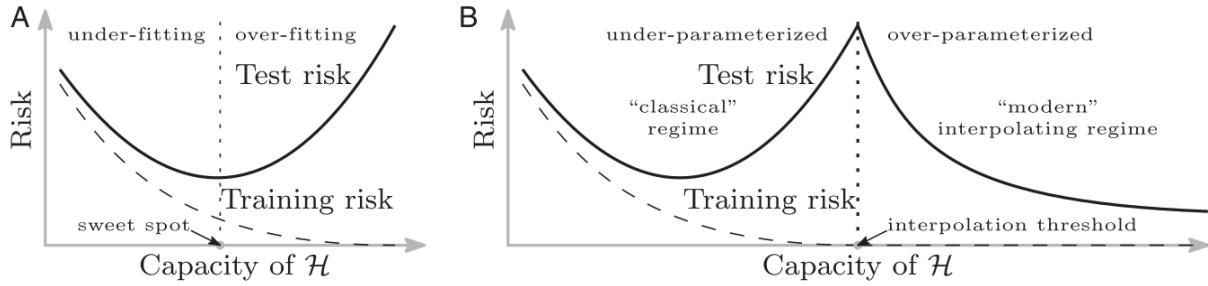


Figure 1.2: Curves for training risk (dashed line) and test risk (solid line). (A) The classical U-shaped risk curve arising from the bias–variance trade-off. (B) The double-descent risk curve, which incorporates the U-shaped risk curve (i.e., the “classical” regime) together with the observed behavior from using high-capacity function classes (i.e., the “modern” interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk. Image taken from reference [20].

1. *Why don't heavily parameterized neural networks overfit the data?*
2. *What is the effective number of parameters?*
3. *Why doesn't backpropagation head for a poor local minima?*

(1) A still unanswered question is how over-parameterized networks show a small generalization error even when a landmark result in Machine Learning (ML) tell us the opposite. The *bias-variance* trade-off predicts that a neural network should be overfitted² in the scenario when the number of free parameters is notably greater than the number of examples in the training set of a supervised learning [19]. Illustrated in figure 1.2-(A), the U-shape curve represents exactly this condition. Note that the sweet spot has two characteristics, first there is a point where the test error reach its minimum value. After this point, the test error increases with the model capacity³. That is why for a significant task to be learned, the selection of the model architecture has always been based on this classic finding, in order to achieve an excellent performance both in the training set and test set, which is the central goal of ML practitioners. However, what we see in Figure 1.2 (B) is not exactly this traditional recipe, i.e, after the training error approximates zero, the test error starts to decrease over again. This phenomenon is known as *double-descent*, and it is genuinely connected to deep learning. The point at which it starts the second descent is called the interpolation threshold, approximately when the number of parameters is equal to the number of samples in training set.

(2) What is the minimum number of samples required for a learning model to generalize well? Typical approach to answer this question involves employing capacity measures. By means of this, it is possible to determine a bound on the generalization error in terms of the number of trainable model parameters relative to the number of examples in training set. Nevertheless, the ability to fit random labels, as known as *memorization*, is one of the recent evidences that demonstrates that usual complexity measures, such as the VC dimension, are not sufficient to explain the generalization performance of state-of-the-art

²Overfitting is expressed when the model has a good accuracy in the training set but a poor one in the test set.

³There are several definitions of capacity one of them is understood as the number of trainable model parameters.

neural networks [21], which raises more questions about the successful learning in deep systems.

(3) Fully understanding the optimization procedure, in deep networks, contains many open problems. The gradient descent of an error function combined with the backpropagation algorithm compose the modern method to train deep neural networks. As a consequence, given an over-parameterized setup, the analysis of the learning trajectories in the surface of loss function is a demanding task. Little is known theoretically about the existence of local minima and their relationships with the test error [22]. This situation persists even for global minima, which it has been characterized with more details in the random labels experiment. The presence of bad global minima⁴ elucidated that *explicit* regularization plays an important role in the generalization error [23].

In the light of those open questions, it is notable that there is a great effort to fully clarify how massive size neural networks work. In special, we address the contributions of the statistical physics community. During the development of a statistical learning theory, the main efforts came from engineering, given the future possible applications, and the mathematical perspective, formalizing it with rigorous results. Despite that, the remarkable success of disordered systems in describing artificial information processing models [24], such as the classic Hopfield neural network [25], concretely positioned as a vital tool in the construction of a learning theory for neural networks. In this sense, bringing again the limitations of classical learning theory in explaining deep nets, it makes us rethink the role of statistical physics in such models. As pointed in [26] titled *Understanding deep learning is also a job for physicists*, solutions of theoretical questions may be solved using a physics-based approach.

To formalize a solid theory in deep learning, three elements and their interactions need to be well understood, structured data, architecture, and training algorithm. For instance, the connection between information processing in the continuum depth limit of deep feedforward neural network and the formal aspects of Renormalization Group has been done [27, 28, 29]. In addition, a concrete example of a recent application lies in the field of condensed matter physics. The success of neural networks in identifying phases of matter [30] has positioned them as an essential tool for reconstructing order parameters of complex systems. This may have interesting consequences, as many physics models are very well understood analytically, it can facilitate making these large black-box models interpretable and explainable. Continuing along this line of reasoning, a new paradigm has recently emerged: physics-informed neural networks [31]. Here, we can incorporate observational, inductive and learning bias to bring prior knowledge of symmetry, conservation and dynamics of physics laws in the training section. Finally, information bottleneck ideas have been employed in support of an information theory of deep learning [32].

1.2 Critical Phenomena

Statistical Physics is a probabilistic framework to extract the thermodynamics (macroscopic) properties of physical systems composed by a huge number of units, leading from the microstates of atoms and molecules to emergent collective behavior described by macrostates. Examples of systems that have been studied come from many fields of natural science, from gases, solids, ferromagnetic materials in physics to the flock of birds

⁴The point where the training error is zero, whereas the test error does not.

in biology. By finding the relevant variables, its main approach involves marginalizing over details of a large arrangement of elements, in the hope of capturing the fundamental aspects on a simplified model. Although it originated as a research field of pure physics, its general character leads to an essential tool in the investigation of a vast number of interdisciplinary phenomena, including models of the brain.

The state of matter can be exhaustively characterized by a very large number, possibly of the order of 10^{23} degrees of freedom $\{x_i\}_{i \in I}$ of a system, where x_i could represent the local magnetic moment of each atom with $i \in \{1, \dots, N\}$. However, a more economical way to overcome this issue is to associate microscopic samples (configurations) to some macroscopic variables through a probabilistic view. For example, energy, density, pressure and so on constitute the primary suitable variables for describing the macroscopic properties of the physical system. In addition, variables, such temperature and external field (magnetic or electric) are *control parameters*, since they are used for definition which specific experimental conditions that a system is being examined. In the context of neural networks, the coupling strengths⁵ are a crucial element for learning and information processing and thus a possible set of variables describing the microstate for analysis of the learning dynamics. Alternatively, the state of the variables describing the neurons can represent the microscopic state. Moreover, the definition of a macroscopic variable of interest, also known as *order parameter*, is vital to depict the state of system. As in the study of *phase transitions* of magnetic materials⁶, where x_i represents a spin, a quantity of interest is the density ρ written by

$$\rho = \frac{1}{N} \sum_{i=1}^N x_i, \quad (1.1)$$

and the susceptibility, a measure of sensitivity to external changes,

$$\chi = \frac{N}{k_B T} (\langle \rho^2 \rangle - \langle \rho \rangle^2), \quad (1.2)$$

where k_B is the Boltzmann constant and T the temperature. Moreover, to gain insights into the microscopic structure, it is pertinent to compute the two-point correlation function

$$\Gamma(\mathbf{r}_i, \mathbf{r}_j) = \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rangle, \quad (1.3)$$

with \mathbf{r}_i representing the position vector of the i^{th} element on site i and $\langle \dots \rangle$ symbolizes the thermal average. Assuming that the correlation function only depends on the distance between of sites, i.e. $\Gamma(\mathbf{r}_i, \mathbf{r}_j) = \Gamma(\mathbf{r}_i - \mathbf{r}_j) = \Gamma(r)$, in general it can follow an exponential decay

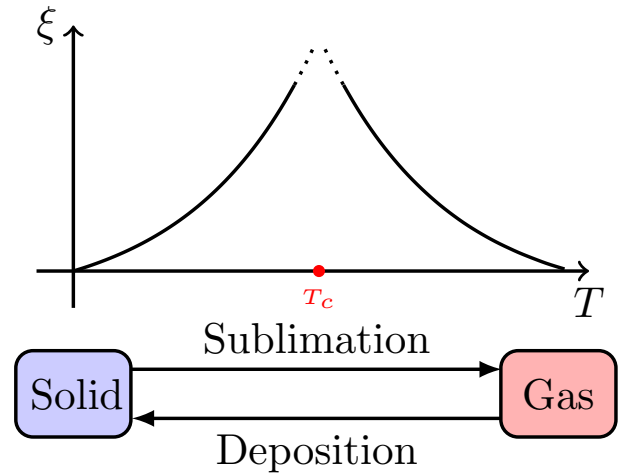
$$\Gamma(r) \sim r^{-\tau} \exp \left\{ -\frac{r}{\xi} \right\} \quad (1.4)$$

when the temperature is distant from the point of phase transition separating ordered and disorder phases represented by the critical temperature T_c or by the critical point (T_c, H_c) , if an external field is taking consideration. Naming ξ as the correlation length which it gives us the typical decay of correlation between two elements of the system with respect to the distance between them, and the exponent τ a constant. From theory of phase transition, at critical point the correlation length goes to infinity. In this situation, the

⁵Also called weights or synaptic couplings.

⁶The reader could have in mind an Ising-like model, but many variations can be considered.

Figure 1.3: Phase transition of matter. T_c represents the critical temperature. In the region close to this point, long-range correlation is observed. The correlation length ξ is a measure of the spatial correlation decay. When $T \rightarrow T_c$, $\xi \rightarrow \infty$.



model become strongly correlated and as a consequence new properties emerge. Within one of them is the aspect of scale-free, where there is not a specific scale to characterize the model. Mathematically speaking, the density, the susceptibility, the correlation length and others relevant variables are governed by power laws (PL). Upon defining the reduced temperature $\epsilon \equiv (T - T_c)/T_c$ and reduced external field $h \equiv (H - H_c)/H_c$, we arrive at the following and respectively equations

$$\langle \rho \rangle|_{h=0} \sim |\epsilon|^\beta, \quad (1.5)$$

$$\chi|_{h=0} \sim |\epsilon|^{-\gamma}, \quad (1.6)$$

$$\xi|_{h=0} \sim |\epsilon|^{-\nu}. \quad (1.7)$$

β , γ , ν are called *critical exponents* and are, in the case of scale-invariance, related by a scaling law. Knowing the entire set of critical exponents, we are able to characterize fully the model near of the continuous phase transition. In particular, the divergence of correlation length expressed by the equation 1.7 is illustrated in Figure 1.3.

Lastly, within our perspective it is important to mention the intricate relation between the susceptibility and the correlation function. As mentioned above, the susceptibility tells us what is the response of density to changes of external influences. Analogous definitions will show to be appropriate when exploring the connection between the external input and the activity of neurons of neural networks. The expression

$$\langle \rho^2 \rangle - \langle \rho \rangle^2 = \langle (\rho - \langle \rho \rangle)^2 \rangle = \frac{1}{N^2} \sum_i (x_i - \langle x_i \rangle) \sum_j (x_j - \langle x_j \rangle) = \frac{1}{N^2} \sum_{ij} \Gamma_{ij}, \quad (1.8)$$

used for the computation of susceptibility, takes into account the correlation between all the elements. Considering time indexing of the state as another dimension, as we shall see, will lead to the analysis of temporal correlations, our first step to describe the activity in the hidden layers of neural network.

1.2.1 Critical Phenomena and the Brain

The idea that the brain operates at a critical point of a phase transition has permeated the field of Neuroscience since the first experimental detection of neuronal avalanches, scale-free bursts of neuronal activity, by Beggs and Plenz [33]. From that time onward, a

collection of evidences has appeared [34], giving support to what is known as the *critical brain hypothesis*. Some characteristics of neuronal avalanches are their size and duration. The size represents the number of neurons fired. Meanwhile, the duration is its lifetime. Thus, Beggs and Plenz breakthrough comes from finding it useful to describe in terms of scale free, i.e power-law (PL), the distributions for the size and duration of avalanches:

$$P(S) \sim S^{-\tau}, \quad (1.9)$$

$$P(T) \sim T^{-\tau_t}, \quad (1.10)$$

respectively, with $\tau = 1.5$ and $\tau_t = 2$. These critical exponents were identified belonging to the same universality classes as those of mean-field directed percolation (MF-DP) models, also termed *branching* networks [35]. Hence, MF-DP has been the canonical model, within the critical brain theory community, to explain the brain's performance, storage and dynamical range, since at the critical point information processing is optimized as shown by Kinouchi and Copelli [36]. In addition, Touboul and Destexhe [37] showed that PL distributions may not be a sufficient signature of critical states. They suggested that a stronger condition to check criticality is by the *crackling noise* scaling relation [38]

$$\frac{1}{\sigma\nu z} = \frac{\tau_t - 1}{\tau - 1}, \quad (1.11)$$

where σ , ν and z are other critical exponents. In a nutshell, σ defines a cutoff in the avalanche size distribution, ν defines the typical length L of the largest avalanche and, z computes the duration T of an avalanche of spatial extent L , expressed as L^z [38]. Finally, the term $1/\sigma z$ is known as the fractal dimension of avalanches. The previous relation 1.11 is derived by observing that the avalanche size and its duration must be coupled through the equation

$$\langle S \rangle \sim T^{\frac{1}{\sigma\nu z}}. \quad (1.12)$$

Thus, when both sides of the equation 1.11, found independently, are the same, the criticality condition is realized.

The Self-Organized Criticality (SOC) theory proposes an explanation of why emergent properties of systems with many interacting elements arise spontaneously. The canonical example to show the relevance of SOC mechanism is by means of the sandpile model introduced by Per Bak, Chao Tang and Kurt Wiesenfeld in seminal 1987 work [39], where scale-invariance cascades of events, known as avalanches, following a power law distribution, cause the system to self-organize towards a critical point of stable and unstable states defined as a continuous absorbing-state phase transition. For the redistribution of grains to happen and for avalanches to be well-defined, SOC says that slow external input (i.e, the number of grains is being added slowly), bulk conservation and boundary dissipation are the essential keys underlying the critical dynamics, which it has been partially linked with the brain functioning and its structure. Nevertheless, the meaning of boundary dissipation in neural systems is not yet clear [40]. Also, the fact that the separation of timescales - between the slow external input and the internal relaxation dynamics of the system (avalanches) - is not concretely applicable in biological systems [41], raises concerns about the sufficiency of SOC mechanics to explain the observed neuronal avalanches and to conclude that the ones are from a criticality. Since the brain is an open system and its activity never ceased, new ideas have emerged suggesting that, in fact, the continuous dynamics of the brain are not at a critical point, but rather orbiting a critical point through a homeostatic mechanism [42], i.e, is self-organizing at the edge of a phase transition.

In addition to the lack of solid theoretical foundations, challenges are faced in the experimental domain, the measurements of avalanches exponents are dependent on experimental techniques, brain region and animal and its conscious state. Inconsistent results have been obtained for distinct avalanches exponents from the same brain region of the same animal species [41], even when both experiments satisfy the *crackling noise* scaling, i.e, both are critical. It leads to puzzling questions: If both experimental technique are compatible, then the critical exponents are only *apparent* or, perhaps, the *crackling noise* isn't a sufficient condition to express signatures of criticality. If the experimental techniques aren't compatible, then we need a standard method to characterize the avalanches exponents. In addition, all experimental techniques are *sub-sampling* the internal brain states and a sub-sampled theoretical model is necessary to reproduce the experimental results [43]. Experimental techniques for detection of avalanches and the criterion for criticality assessment are still unresolved problems [41, 44].

Given this set of empirical evidences and broad-tailed distributions are observed in a wide range of natural phenomena [45], it leads to the assertion that the avalanches obeyed PL are an incomplete description of phase transitions in brains. The problem is not only the characterization of PL's, but also in the theoretical framework. The scaling relations of critical exponents of second order phase transition, at a mean-field level, found in MF-DP, are only compatible with those observed by experimental data on spike avalanches through a *fine-tuning* [46]. In addition to all of this, it is important to stress that modeling brain tissue in terms of networks of stochastic neurons prompts us to inquire if this is a justifiable path to understand capabilities only seen in systems that learn from experience. In others words, a *meaningful* task behind a theoretical model is a pathway to get closer to the role of *function* in biology. Consequently, that makes us consider an AI machine as a possible system to examine questions related to criticality signatures and optimal information processing. Therefore, we seek clarification on the following issue:

Do the time series of internal representations of inputs in a complex artificial neural network, successful in a meaningful task, present critical activity signatures, analogous to the ones found in biological systems?

The answer will most likely differ from a simple yes or no. A clear “yes” would have quite important consequences, as it might help in further identifying criticality as a general signature of successful information processing. In addition, it may illustrate and help in the design of experiments probing the neural time series from live animals, such as mice or rats, obtained by multi-electrode implantation, by allowing rapid test of hypotheses in the artificial set up, that would demand vast resources and times in the biology lab. Our modeling proposal would be an addition to current mathematical models of neural networks, based on more realistic neurons of the Hodgkin-Huxley class. Since these models have not been trained to classify large data sets in a meaningful task, they might fail to reproduce the information processing aspects that lead to optimal information processing at, or near, the edge of criticality, although they might be considered far closer to the experimental neural architecture. A clear “no” may bring questions about whether the correct variables have been identified in the artificial neural networks which can be thought of analogous to the biological ones. An intermediate answer will, hopefully raise new questions that may suggest new experimental research questions.

1.3 Complexity Index

There is no doubt that capturing the essential behavior of complex systems is a challenging effort in the actual scientific scenario. A complex system exhibits emergent phenomena, by which it can not be simply understood in terms of its constituents. Phrasing “The whole is greater than the sum of its parts.”, there is no place for a *reductionist* methodology in the study of such systems which doesn’t take into account the consequences of the interactions. As an example, *flocking* only appears when a group of birds interacts following a *specific* rule, just like consciousness emerges through the sophisticated connection between billions of neurons in different cortical areas [47]. In short, one of the common proprieties observed here it is a complex pattern from a collective group of elements.

An initial approach to understanding this class of models would be to evaluate complexity indices as a way to identify the presence of an emergent process. For example, Integrated Information Theory (IIT), formulated by G. Tononi [48], aims to propose an objective measurement of consciousness over any physical system of units, introducing a quantity called of integrated information Φ . However, the calculation of this quantity may be computationally intractable in most experiments. A concrete approximation of Φ , based on information geometry was proposed by Oizumi *et al.* [49]. From this study, numerous applications have arisen in many fields [50], including the classic Ising model [51]. As an empirical example, an index of brain complexity called the perturbational complexity index (PCI) was shown to be useful in clinical practice, since it was demonstrated to be capable to distinguish level of conscious experience for individuals showing different brain states (wakefulness, dreaming, anesthesia, coma etc.) [52], where the idea underlying it is to measure the spatio-temporal pattern of the brain’s activity after of a transcranial magnetic stimulation (TMS).

Since neural networks are information processing devices, it seems reasonable to question if a complexity index would be useful for understating of learning process. In others words, what is expected of the *activity pattern* when a machine is learning a recognition of images from a dataset? Maybe a quantity that measures complexity is a strong indication that artificial intelligence is an emergence process. It is vital to stress the following idea: the accuracy is already a measure of performance, however, it is only relative to output layer, thus what we want to do is to understand the complexity *inside* the machine. In short, we address the following questions in this dissertation,

Does an artificial intelligence machine become complex as it gets trained?

Could a complexity index be an order parameter of learning process stages in artificial neural networks?

1.4 Materials and Methods

In this project, we intend to investigate the phenomenology of learning processes in artificial intelligence systems and the connections it might have with unsolved problems in Neuroscience. The primary methodology will be the recording of time series of internal representations of an artificial neural network in deep architectures of spiking units performing a recognition task.

Inspired by the analysis of time series from cortical implanted electrodes in rats, either freely behaving or under different levels of anesthesia, our aim is to mimic the procedure

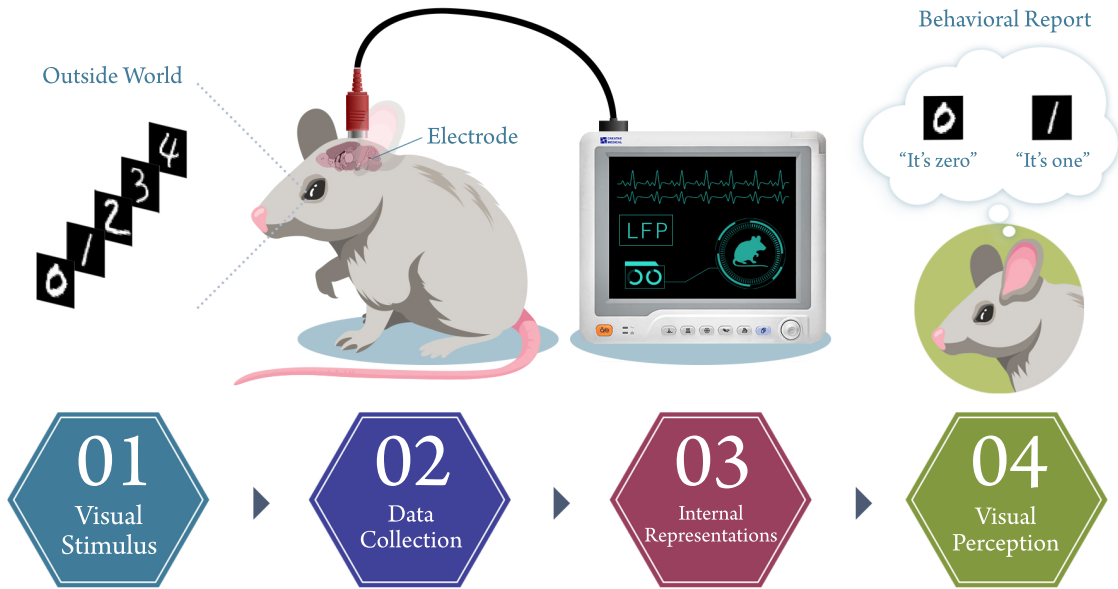


Figure 1.4: Typical experimental setup, using electrodes for studying cortical activity during a visual perception task. We adapted the elements of that study to an artificial neural network environment.

done in a neuroscience laboratory, but now taking an artificial neural network as the main subject of study. As illustrated in Figure 1.4, a typical sensory perception experiment is consisted by some steps: external stimulus, the recording of brain activity and the animal behavior report. In a straightforward manner, it can easily translate for our context taking into consideration the following point: the external stimulus is described as a sequence of image inputs, the data collection of the neural network activity is done by a machine learning framework and the last, the behavioral report, which it is, in this case, a visual perception of a image expressed by the words “This is an image of the number zero”, it will be interpreted by the successful classification of a labeled input. In addition, the term *internal representations* (IR), expressed in Figure 1.4, are the internal states of the animal brain employing a cognition task, i.e, they are all the neuronal representations of the external stimulus from the outside world. Hence, given the classifiable inputs, the IR of a trained deep neural network are the activity of neurons of the internal layers.

Finally, since the feedforward spiking neural network will be use as the primary model, it is important to express the reasons of our choice. There are two points here, the first lies in the architecture. As already explained, there is not yet a well-developed theory for deep learning and so, consequently, the fully connected layers neural network ends up being the main massive model to be explained and interpreted. Through empirical evidences, we intend to answer questions about the connection between the network structure and its information processing. Second, spike units bring us roughly closer to biology neural networks. Now, we are dealing with excitable binary neurons modeled by a dynamics of membrane potential and thus, the feature of recurrent⁷ dynamics, crucial element in biological nets, is present.

⁷In our context, recurrent means a short-term memory.

Chapter 2

Spiking Neural Networks

Pensé en un mundo sin memoria, sin tiempo; consideré la posibilidad de un lenguaje que ignorara los sustantivos, un lenguaje de verbos impersonales y de indeclinables epítetos. Así fueron muriendo los días y con los días los años...

JORGE LUIS BORGES, El Inmortal (1947)

2.1 A Brief Introduction

Artificial neurons usually transmit information through a single continuous and time-less value of an activation function $h(x)$, with x a weighted sum of inputs. In contrast, in biological neural networks, transmission occurs through several biophysical computations involving synaptic interactions, ion channels and variation of membrane potential over time, but primarily through action potential, roughly expressed by spikes. There are several mathematical models of the action potential generating mechanism which in general can be traced back to Hodgkin and Huxley's (HH) seminal contribution. These models vary in mathematical complexity and can lead to high computational costs. A simple representation of occurrence of a spike is by means of a Heaviside step function $\Theta(U(t) - \theta)$, where $U(t)$ the membrane potential in time and θ the firing threshold, illustrated in Figure 2.1a. This curve symbolizes that when the membrane potential reaches a threshold θ , the neuron will fire, expressed by a non-zero value of step function. Of course, the full dynamics of membrane potential over time is behind of that and so the next step will be to establish a model for it. To give a concrete example, Integrate and Fire (IF) models captured the essential properties underlying electric potential difference in nerve cells, resulting a cheap computational power. As it will be employed in our neural network, the next section will be in charge of studying in detail. Choosing an alternative route, since the strong nonlinearities HH lead to numerically very stiff differential equations, an activation function $h(x)$ that represents an average firing rate, permits the use of minimization techniques for gradient based learning algorithms in neural networks, in which the sigmoid and the ReLU¹ curves are the typical transfer functions of widespread use. The sigmoid one is plotted also in Figure 2.1a. The interpretation of θ following

¹ReLU means rectified linear unit.

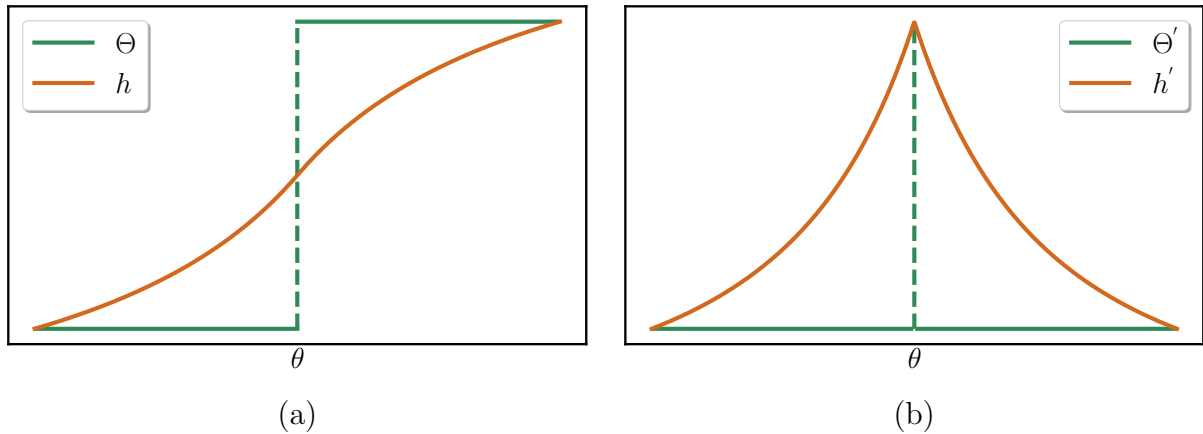


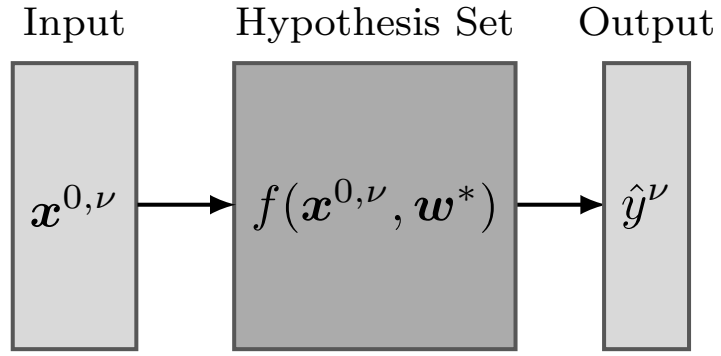
Figure 2.1: The Heaviside step function $\Theta(U(t) - \theta)$ is representing a spike neuron (green curve) and $h(x)$ an activation function of an artificial neuron (orange curve). (a) Forward pass: The process of computation of inputs. We can see that the activation function is a kind of smooth step function. (b) Backward pass: The process of computation of derivative through of backpropagation algorithm. In this situation, the derivative Θ' is zero everywhere except at θ , where it is infinite. Meanwhile, $h'(x)$ has a well-defined derivative across the entire x-axis.

the same idea, the only difference is that θ is not a membrane potential threshold, but threshold of a weighted sum of inputs.

Illustrated in Figure 2.1b, the impossibility of spiking neurons handling the backpropagation algorithm, by virtue of non-differentiable behavior in the step function, referred to as the *dead neuron* problem, led to the use of the artificial neuron as the canonical building block of neural networks for machine learning models. Nevertheless, it doesn't mean that it isn't possible to train a deep neural network of spiking units using variations on the theme of the same methodology. A few relevant methods for training a spiking neural network (SNN) will now be mentioned [53]. The *Shadow Training* algorithm consists in converting a trained ANN to SNN by means of transformation of activation functions into spike rate. The *Surrogate Gradient* applies precisely the backpropagation algorithm into connections between spiking neurons using an approximation of the derivative of the Heaviside function, only in the backward computation. In particular, since we will be focusing on SNN of feedforward architecture, and it would be advantageous to bring well established methods of machine learning for brain-inspired models, the last one will be used.

The major force behind the development of SNNs stems from the need to create an energy efficient model for real-world signal processing in artificial intelligence systems. The inspiration for this comes from how efficiently the brain operates with only about $20W$, and it delivers strong performance on cognition tasks. Going down a different path from the von Neumann architecture, neuromorphic engineering, based in the brain structure, has been developed to set up an efficient and scalable computation for large neural networks [54]. Despite the applications that this can lead to the light of computer science, our attention lies in understating the information processing in these type of models, as already explained.

Figure 2.2: The black box of artificial intelligent machines. $\{\mathbf{x}^{0,\mu}, y^\mu\}$ is the input/output pair, $\hat{y}^\nu = f(\mathbf{x}^{0,\nu}, \mathbf{w}^*)$ is the prediction made by a machine capable to implement the hypothesis set $f_{\mathbf{w} \in \Lambda}$ and \mathbf{w}^* is the parameter found by a machine after the learning process.



2.2 Training a Spiking Neural Network

Three mayor modelling decisions have to be made in order to implement the SNN: the learning problem, the architecture and the learning algorithm. We now discuss these points in detail.

2.2.1 Learning Problem

A training set consists of n independent and identically distributed (i.i.d) input/output pairs, \prime . In a multi-class classification problem, $y^\mu \in \{1, 2, \dots, C\}$ is the class of input $\mathbf{x}^{0,\mu}$, where C is the total number of classes. Given a set of parametric functions $f_{\mathbf{w} \in \Lambda} = \{f(\mathbf{x}^0, \mathbf{w}) : \mathbf{w} \in \Lambda\}$, also referred by *admissible functions* or *hypothesis set*, and any sample of $P(\mathbf{x}^0, y)$, $(\mathbf{x}^{0,\nu}, y^\nu)$, the goal of *supervised learning* is to find a function, $\hat{y}^\nu = f(\mathbf{x}^{0,\nu}, \mathbf{w}^*)$, capable to predict correctly the class of any input $\mathbf{x}^{0,\nu}$, i.e, $\hat{y}^\nu = y^\nu$, illustrated by 2.2. It does not imply that there is only one function, but the one that the best approximates the true output. We commonly designate the term \mathbf{w} as *learnable parameters* or *weights*. The *loss function* $\mathcal{L}(y, \hat{y}^\nu = f(\mathbf{x}^0, \mathbf{w}))$ is a measure of the distance between the prediction made by \hat{y}^ν and the true value y^ν . In order to achieve the goal of learning, we should minimize, over the family $f_{\mathbf{w} \in \Lambda}$, the expected value of the loss function, so-called the *population risk* or *generalization error*

$$e_G(\mathbf{w}) = \int d\mathbf{x}^0 dy P(\mathbf{x}^0, y) \mathcal{L}(y, f(\mathbf{x}^0, \mathbf{w})). \quad (2.1)$$

But in general, we don't have access to the distribution $P(\mathbf{x}^0, y)$, therefore the straightforward method to minimize the risk is via an approximation, so-called *Empirical Risk Minimization* (ERM) written by

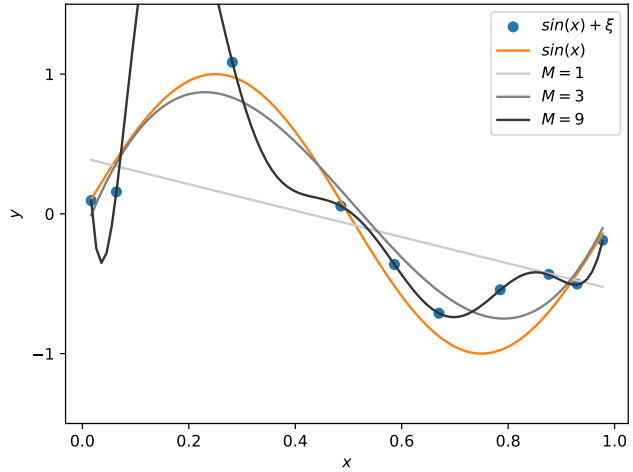
$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \Lambda} \left[\frac{1}{n} \sum_{\mu=1}^n \mathcal{L}(y^\mu, f(\mathbf{x}^{0,\mu}, \mathbf{w})) \right], \quad (2.2)$$

where the quantity being minimized is the *empirical risk*, popularly known by the *training error*

$$e_T(\mathbf{w}) = \frac{1}{n} \sum_{\mu=1}^n \mathcal{L}(y^\mu, f(\mathbf{x}^{0,\mu}, \mathbf{w})). \quad (2.3)$$

One of the issues about minimization is that, typically, once we find the global minima of empirical risk, there is no guarantee that we will also find a low population risk $e_G(\mathbf{w}^*)$. That situation is described when a trained neural network has a poor generalization ability.

Figure 2.3: Polynomial regression. For 10 data points that follow a sine curve affected by Gaussian noise $\mathcal{N}(\mu = 0, \sigma^2 = 0.2)$, overfitting happens when the order polynomial function is $M = 9$.



This is commonly referred to by the community as *overfitting*. One of the main reasons for this phenomenon lies in the choice of model architecture *complexity* $f_{\mathbf{w} \in \Lambda}$, the minimization technique and the nature of the task to be learned. The typical illustration to elucidate this is when we want to fit an observed data set. Imagine that this data roughly follows a sine curve. If we decide to fit a polynomial function to this data set, we must choose the polynomial order appropriately. A high polynomial order gives enough degrees of freedom so that the fitted curve follows *exactly* the observed data, see Figure 2.3. This is a bad scenario, since new data would be far from the curve of best fit by the training set. In other words, one inappropriate choice leads to drastic consequences on generalization ability. That's why the selection of the model architecture is crucial to show an excellent performance both in the training set and the test set, which is the main goal of machine learning.

A direct algorithm to perform the optimization problem defined in equation 2.2 is *gradient descent* (GD) expressed by

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta_n \nabla_{\mathbf{w}} e_T(\mathbf{w}_n) \quad (2.4)$$

where $\eta_n \in \mathbb{R}_+$, the *learning rate*, determines the step size at each interaction. Of course, the last interaction of this algorithm occurs when the gradient is zero $\nabla_{\mathbf{w}} e_T(\mathbf{w}^*) = 0$ implying that we can identify \mathbf{w}^* as local or global minima of empirical risk. Equation 2.4 says that the learnable parameters are updated at each step in the direction of the steepest decrease of the loss function, i.e, in its negative gradient,

$$-\nabla_{\mathbf{w}} e_T(\mathbf{w}) = -\frac{1}{n} \sum_{\mu=1}^n \nabla_{\mathbf{w}} \mathcal{L}(y^\mu, f(\mathbf{x}^{0,\mu}, \mathbf{w})). \quad (2.5)$$

Moreover, in multilayer perceptron, equation 2.4 represents updating the weights and computing of the gradient over the weights between two consecutive layers. As we add more layers to our model, the computational cost of the *full gradient* at each step of GD ends up becoming high. Not only that, this issue becomes more relevant when we increase the number of samples from the training set \mathcal{D} . The term *full gradient* means we need to compute the gradient for all n samples of \mathcal{D} *at each step* of GD. For these reasons, an alternative to overcome this problem is to compute the gradient only on subsets of a previously randomly shuffled training set $B_k \subset \mathcal{D}$ known as *mini-batches*. This approach

is called *stochastic gradient descent* (SGD) [55] and it is one of the most used optimization technique in Machine Learning practice, especially in Deep Learning. Thus, the sequential interaction of the weights adjustments follows the equation

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \frac{1}{|B_k|} \sum_{\mu \in B_k} \nabla_{\mathbf{w}} \mathcal{L}(y^\mu, f(\mathbf{x}^{0,\mu}, \mathbf{w})), \quad (2.6)$$

with $|B_k|$ the mini-batch size. Notice that it is only necessary to compute the gradient of the loss function in each mini-batch. Although the analytical calculation of the gradient can be a straightforward task, a naive strategy of this calculation can increase the computational cost substantially. We repeatedly emphasize in the computational cost considerations because of the *time complexity*, which it is an essential issue for Deep Learning implementation. In fact, we are interested in neural networks with multiples hidden layers, potentially with millions or billions parameters to be adjusted and so with a massive number of mathematical operations required for the computation of all components of the gradient during the convergence of the gradient step descent. In that regard, the *error backpropagation* algorithm is an efficient method for evaluating the gradient of any differentiable multivariate function, in such way that the determination of the sequential chain rule is just a special case of the *reverse mode of accumulation differentiation*. The principal distinction between the *automatic differentiation*² and the *symbolic differentiation*³ becomes clear in the face of a complicated algebraic function. The first one incorporates intermediary functions through elementary arithmetic operations to apply the chain rule repeatedly to each term comprising the global function. The big advantage of this approach lies in the implementation of an efficient code for the computing of the gradient, where it will combine two types of procedure, the *forward pass* representing, given an input, the calculation of the output from all layers in the neural network, and the *backward pass* representing the sequential computation of all *errors* associated of each neuron from the output layer to the input layer. To translate this into equations, let's consider a fully-connected feedforward neural network of artificial neuron units with L layers. The output $y_j^{(l)}$ of the j^{th} neuron in the l^{th} layer is expressed by

$$\begin{cases} z_j^{(l)} &= \sum_k w_{jk}^{(l)} y_k^{(l-1)} \\ y_j^{(l)} &= \sigma(z_j^{(l)}), \end{cases} \quad (2.7)$$

where $w_{jk}^{(l)}$ is the synaptic weight between the k^{th} neuron in the $(l-1)^{\text{th}}$ layer and the j^{th} neuron in the l^{th} layer. $\sigma(\cdot)$ is the activation function and $z_j^{(l)}$ is the weighted sum of inputs. Thus, one of the components of the gradient of the loss function $\mathcal{L}(\mathbf{w})$ is the partial derivative

$$\frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}}, \quad (2.8)$$

that is, the sensibility of loss with respect to learnable parameter $w_{jk}^{(l)}$. Using the chain rule and the definition done in equation 2.7, we obtain

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}} &= \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{jk}^{(l)}} \\ &= \delta_j^{(l)} y_k^{(l-1)}, \end{aligned} \quad (2.9)$$

²It combines two modes: forward and reverse accumulation.

³a.k.a algebraic differentiation.

with $\delta_j^{(l)} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}}$ representing the contribution to the *error* related to the j^{th} neuron in the l^{th} layer. In simpler terms, with the purpose of calculating all the partial derivatives in 2.8, we must establish an equation for the error of all neurons, since $y_k^{(l)}$ has already computed in the forward pass. Stated differently,

$$\begin{aligned} \delta_j^{(l)} &= \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \\ &= \sum_k \frac{\partial \mathcal{L}}{\partial y_k^{(l)}} \frac{\partial y_k^{(l)}}{\partial z_j^{(l)}} = \frac{\partial \mathcal{L}}{\partial y_j^{(l)}} \frac{\partial y_j^{(l)}}{\partial z_j^{(l)}} \\ &= \frac{\partial \mathcal{L}}{\partial y_j^{(l)}} \sigma'(z_j^{(l)}), \end{aligned} \quad (2.10)$$

of course, with the condition

$$\frac{\partial y_k^{(l)}}{\partial z_j^{(l)}} = 0 \text{ if } k \neq j. \quad (2.11)$$

However, as the output of neural network is a combination of function composition and the matrix multiplication, it is expected to find a unique expression that combines the error associated of l^{th} layer and the $(l+1)^{\text{th}}$ layer. In fact, applying again the chain rule in the error equation and using equation 2.7, we obtain

$$\begin{aligned} \delta_j^{(l)} &= \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} = \sum_k \frac{\partial \mathcal{L}}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} = \sum_k \delta_k^{(l+1)} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} \\ &= \sum_k w_{kj}^{(l+1)} \delta_k^{(l+1)} \sigma'(z_j^{(l)}), \end{aligned} \quad (2.12)$$

an equation of the error of l^{th} as function of the error of $(l+1)^{\text{th}}$ next layer. As we can see this previous equation has a nice interpretation, the computing of the error is backwards, i.e, from the output layer to input layer. This is the core of the error backpropagation algorithm [56]. To put it briefly, in order to calculate the gradient of a loss function over the parameters, we must start the calculation of the error associated in the output layer, the L^{th} layer, and then the error associated in the $(L-1)^{\text{th}}$ layer and so on. By employing this strategy, all partial derivatives needed to compute the gradient are evaluated easily.

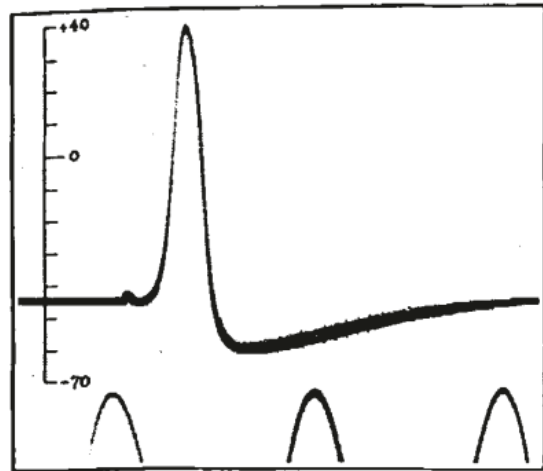
In short, we have the following equations of the error backpropagation technique in multilayer neural network with L layers

$$\begin{cases} \delta_j^{(L)} = \frac{\partial \mathcal{L}}{\partial y_j^{(L)}} \sigma'(z_j^{(L)}) \\ \delta_j^{(L-1)} = \sum_k w_{kj}^{(L)} \delta_k^{(L)} \sigma'(z_j^{(L-1)}) \\ \frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}} = \delta_j^{(l)} y_k^{(l-1)}. \end{cases} \quad (2.13)$$

2.2.2 Neural Modeling

Nonlinear spatio-temporal partial differential equations (PDE) are obtained through a detailed description including ionic conductance and morphology of neural cells (see e.g

Figure 2.4: An experimental measurement of the action potential. Image taken from reference [58].



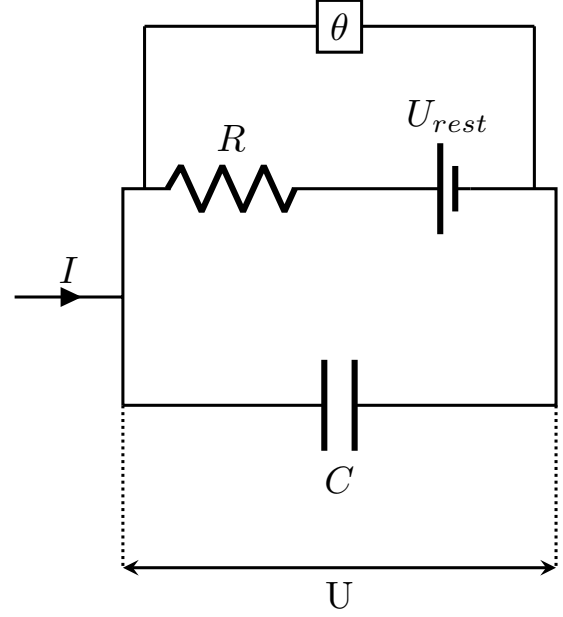
[57] and references therein). Thus, it seems impractical modeling a neural network using a realistic description of the neurons, at least if we are studying the computational dynamics and not the physiological details. For the sake of simplification, we can remove the space dependence in the PDE, so-called spatial discretization process, which leads to a set of coupled nonlinear ordinary differential equations (ODE), known as conductance-based models. The classical example to understand the ionic conductance roles in the neuron activity is the Hodgkin–Huxley model. The groundbreaking success in the Hodgkin–Huxley’s experimental measurements was the first modeling of ionic mechanism involved an action potential of the nerve cell membrane [58], indicated in Figure 2.4. Even so, conductance-based model is still highly computationally costly for the theoretical study of macroscopic dynamics of a population neuron, indicating that we should seek another neural modeling for large-scale training of deep neural networks. Some other limitations are

1. Dynamical systems with large numbers of control parameters makes qualitative analysis an arduous task. Conductance-based models are parameterized in a high-dimensional space [57], suffering from the *curse of dimensionality* [59].
2. Conductance-based models are *experimentally constrained*. It says that experimentally well-defined parameters of neurons from some region of the brain, or even from another living being, can be used to represent the biophysical mechanisms of completely distinct neurons, leading to what we call *Frankenstein models*⁴.

It is reasonable to search for a simplified and general purpose model. The first step would be to set aside all neuron’s biophysical dependencies from conductance-based models and only concern on the phenomenology of action potentials. In such manner, the integrate-and-fire (IF) neuron is equivalent to the electrical circuit with only a capacitance, characterized by the dynamics of the nerve cell’s membrane potential as an integration process of synaptic and external inputs, and by a reset mechanism, where the membrane potential is returned to its resting potential after reaching the firing threshold. While the IF model describes satisfactory single spike events, a key ingredient for the dynamics of the membrane potential is still missing, the leaky current. The leaky integrate-and-fire (LIF) [57] neuron follows the same principle of IF model, integration

⁴A term commonly used in the Laboratory of Neural Systems, FFCLRP.

Figure 2.5: LIF neural modeling. R , C , U_{rest} , I , θ are the resistor, capacitor, resting potential, current input and threshold firing within reset mechanism, respectively.



of inputs combined with a reset mechanism, the only difference is in a new element of resistance to the electrical circuit representing the leaky, an RC circuit as illustrated in figure 2.5. Accordingly, the total input current in that circuit can be expressed by

$$I(t) = I_C(t) + I_R(t), \quad (2.14)$$

where $I_C(t)$ and $I_R(t)$ represent the electric current flowing through the capacitor and resistor, respectively. These currents can be denoted by the following equations

$$\begin{cases} I_C(t) &= C \frac{dU(t)}{dt}, \\ I_R(t) &= \frac{U(t) - U_{rest}}{R}, \end{cases} \quad (2.15)$$

with $U(t)$ the membrane potential at time t and U_{rest} the resting potential. Now, defining the variable $\tau = RC$ as the time constant of the neuron, equation 2.14 is formulated by

$$\tau \frac{dU(t)}{dt} = -[U(t) - U_{rest}] + RI(t) \quad (2.16)$$

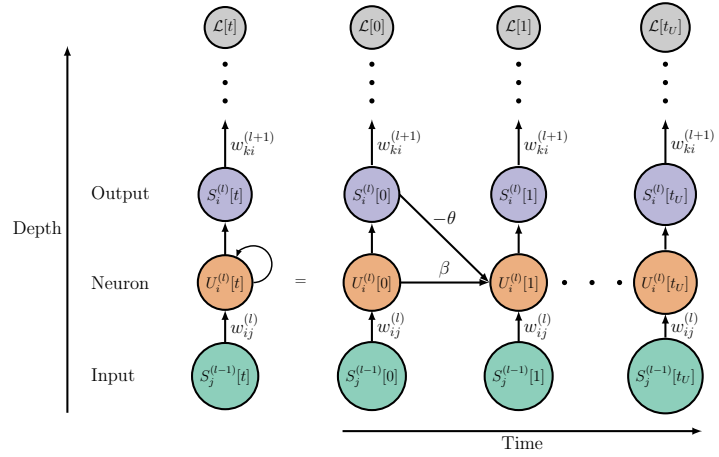
a first-order linear ODE. In terms of the biophysical aspects of nerve cells, the expression 2.16 is called the equation of a *passive membrane*. Nevertheless, equation 2.16 is not the final description of the dynamics of the LIF neuron, we must define a value of firing threshold θ and a reset mechanism for the resting potential. After that, the description of the LIF neuron is completed. Additionally, from this point we will assign $U_{rest} = 0$ across the entirety of this project without loss of generality. Hence, the solution of equation 2.16 for current input constant I is

$$U(t) = RI + [U(0) - RI]e^{-\frac{t}{\tau}}. \quad (2.17)$$

We are concerned with implementing that spiking neuron model as the building block of a SNN, so the starting point would be to do a numerical integration of equation 2.16. Since numerical precision is not crucial in our context⁵, the simple Euler method is enough

⁵Computing speed is more important.

Figure 2.6: Unfolded computational graph of the LIF neuron with a reset by subtraction of threshold θ . Loop arrow symbolizes the dynamics of membrane potential in time t . Depth indicates the direction of the output layer. $(l)^{\text{th}}$ is the consecutive layer of the $(l - 1)^{\text{th}}$ layer. $\mathcal{L}(t)$ is the loss function computed at step time t . Finally, t_U is the last step time of membrane potential.



for our modelling purposes. Therefore, when choosing a Δt small enough, we obtain that the membrane potential at the next time step is

$$U[t + \Delta t] = \left(1 - \frac{\Delta t}{\tau}\right) U[t] + \frac{\Delta t}{\tau} RI[t]. \quad (2.18)$$

We designate the term $\left(1 - \frac{\Delta t}{\tau}\right) = \beta$ as an estimate of the inverse time constant $\tilde{\beta} = e^{-t/\tau}$ for condition when $\Delta t \ll \tau$. The meaning of β becomes clear when the total input is zero, where the remaining components in the numeric solution 2.18 and in the solution of original equation 2.16 are respectively

$$\begin{cases} \frac{U[t + \Delta t]}{U[t]} = \left(1 - \frac{\Delta t}{\tau}\right) = \beta \\ \frac{U(t)}{U(0)} = e^{-t/\tau} = \tilde{\beta}, \end{cases} \quad (2.19)$$

thus β and $\tilde{\beta}$ are the ratio between two successive membrane potential values. Due to the association between the variables β and τ determined by this prior equation, we will only utilize β as a membrane potential decay measure. Consequently, the expression 2.18 results on

$$U[t + \Delta t] = \beta U[t] + (1 - \beta) RI[t]. \quad (2.20)$$

At present, let's adjust the resistance value to $R = 1$ with the aim of reducing the number of parameters, that means we measure resistances in units of R . Let's see that this will not impact in the numerical simulation, as long as the definition of the membrane potential decay is related to the product of the resistance and the capacitance, $\tau = RC$. Then, the selection of an appropriate value of τ is more than enough. As described in reference [53], two non-physiological assumptions are made here:

1. The input current is adjusted by one time step increment, i.e, $I[t] \longrightarrow I[t + 1]$.
2. The element $I[t + 1]$ is interpreted as $I[t] = \mathbf{w}\mathbf{S}[t]$, where \mathbf{w} is the connection matrix between two consecutive layers and $\mathbf{S}[t]$ is any from of input. As consequence, the term $(1 - \beta)$ will be assimilated by \mathbf{w} .

In the first assumption, the purpose to incorporate an increment into the time step of the input current focuses on the equivalence of the recurrent dynamics of membrane

potential, characterized by the fact that the state of neuron is determined solely by its previous state. This can be seen by considering the following structure of a dynamical system, with $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(T)}$ representing states over time

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \mathbf{w}), \quad (2.21)$$

where f a general function, $\mathbf{x}^{(t)}$ the sequential inputs received by the network and \mathbf{w} the parameters used for the evaluation of the sequential states. This feature is illustrated in Figure 2.6 and it is indicated by the loop arrow.

The second assumption ensure that the input $\mathbf{S}[t]$ is scaled only by the connections matrix \mathbf{w} . The justification behind this is that, while we are defining $\mathbf{S}[t]$ as the firing of the presynaptic neurons, in general, $\mathbf{S}[t]$ can assume various input forms such voltage. And so, decoupling \mathbf{w} and $(1 - \beta)$ favors the computation of next step of membrane potential over biological precision.

Going forward, we need to pay attention to the reset mechanism after the neuron reaches the firing threshold. Reset by subtraction of threshold and reset to zero the membrane potential are the most frequently utilized in the Computational Neuroscience literature. Typically, the second one promotes sparsity in the network activity, as the majority of neuron states spend more in quiescence, below the firing threshold. Ultimately, we are fully prepared to write the dynamics of i^{th} membrane potential in the l^{th} layer $U_i^{(l)}$ of LIF neuron with a reset by subtraction of threshold, i.e,

$$\begin{cases} S_i^{(l)}[t] = \Theta(U_i^{(l)}[t] - \theta) = \begin{cases} 1 & \text{if } U_i^{(l)}[t] > \theta \\ 0 & \text{otherwise,} \end{cases} \\ U_i^{(l)}[t+1] = \beta U_i^{(l)}[t] + \sum_{j=1}^{N^{(l-1)}} w_{ij}^{(l)} S_j^{(l-1)}[t+1] - \theta S_i^{(l)}[t], \end{cases} \quad (2.22)$$

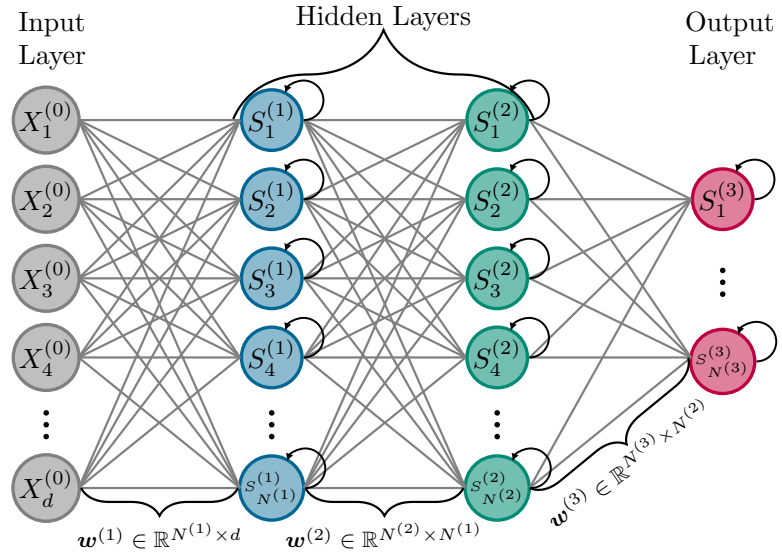
where $S_j^{(l-1)}$ is the presynaptic input of the j^{th} neuron in the previous layer ($l-1$), $w_{ij}^{(l)}$ is the adjustable synaptic coupling between the j^{th} neuron in the previous layer ($l-1$) and the i^{th} neuron in layer (l) and β controls the membrane potential decay. These two expressions provide a complete description of the LIF neuron. In figure 2.7, we show a deep multilayer spiking neural network.

2.2.3 Learning Algorithm

As discussed in the preceding section 2.2.1, in order to successfully address a learning problem, we must confront an optimization problem involving the minimization the expected loss function within a parameter space. In this way, one can proceed to construct a learning algorithm for our model, it will be necessary to compute the loss function through spike neurons. To achieve this, first, the precise definition of the neural code must be well done. Once the neuronal firing communication is defined, the computation of the response of neurons in the output layer is feasible as a recognition of the label of an image and thus enabling a comparison between that predicted response by neural network and the true targets. This will form the initial basis for pattern recognition in SNN.

Through sensory perception investigations, distinct firing patterns recording are associated with different external sensory stimuli, such as visual, sounds and smells. In some way, sensory information about the outside world is encoded by means of neuronal

Figure 2.7: The diagram shows the fully connected feed forward spiking neural network. Each unit $S_i^{(l)}$ is a LIF spiking neuron and the loop arrow symbolizes the time-dependent changes of the membrane potential.



activity ensuring their accurate perception. If that's true, what is the nature of the neural code?

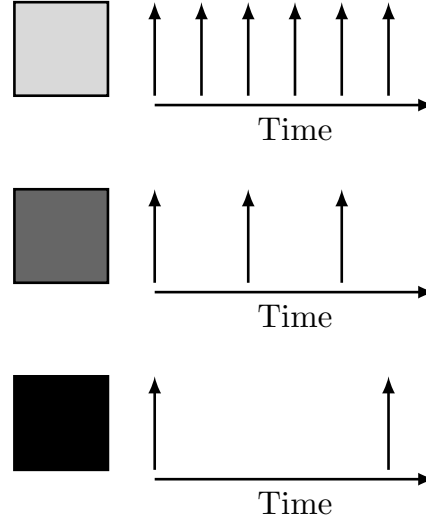
Answering this question is beyond the scope of this dissertation. However, regardless of *what* the neural code of the nervous system is, what can be ascertained is that the brain employs either distinct or combination of neural codes depending on the nature of the stimulus and the cognition task being performed. The principal ones encompass rate coding and temporal coding. In special, the second one arises when information can be assigned to the high temporal resolution in a sequence of spikes. Each temporal structure of a spike train carries a meaningful information [57]. On the other hand, rate coding relates to the transmission of information through the neuron firing rates. For instance, in Figure 2.8, each bright pixel of light represents the stimulus strength that a peripheral visual system receives, and it is encoded proportionally with a neuron firing rate response. Bright light corresponds to a higher firing frequency response than dark light. The initial procedure of neural code is known as input coding and the final one as output decoding. The last step will be relevant when we seek to comprehend the associations of neuronal activity patterns and the behavioral report of an animal, for example. Here, it will be crucial in our classification problem in such a way that each response of output layer will be connected with a recognition of image, 'It's zero' or 'It's shoes'. This is our behavioral report, a visual perception of a labeled input. For reasons of simplification and of general purpose, rate coding will be implemented.

Given an input/output pair represented by a labeled image $(\mathbf{x}^{0,\nu}, y^\nu)$ and a trainable SNN denoted by its parameters \mathbf{w} , we will count how many spikes are observed in each i^{th} neuron of the output layer through the dynamics of the membrane potential over t_U discrete time steps. After that, we select the index of neuron that has the highest count as the label predictor $\hat{y}(\mathbf{x}^{0,\nu})$, i.e.,

$$\mathbf{s}(\mathbf{x}^{0,\nu}) = \sum_{t=0}^{t_U} \mathbf{S}[\mathbf{x}^{0,\nu}, t] \implies \hat{y}(\mathbf{x}^{0,\nu}) = \arg \max(\mathbf{s}), \quad (2.23)$$

where $\mathbf{S}[\mathbf{x}^{0,\nu}, t] \in \{0, 1\}^C$ represents the response of output layer at t time step, $\mathbf{s}(\mathbf{x}^{0,\nu})$ is the spike count of output layer over t_U time steps and C is the number of neurons in the output layer, or rather, the number of classes/labels. Consequently, the rate coding is converted through the argument of the maxima spike count of the output layer. However,

Figure 2.8: Rate neural code.
Each intensity of light triggers
a rate spike.



this is insufficient to establish a metric to quantify the predictor error. One way would be to associate each i^{th} neuron spike count s_i to a softmax function written as

$$p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i = \frac{e^{s_i}}{\sum_{j=1}^C e^{s_j}} \quad (2.24)$$

which can be interpreted as the probability of the prediction of a label y^μ when an input $\mathbf{x}^{0,\nu}$ is presented to the network model with parameters \mathbf{w} . It is clear that the normalization is satisfied with respect to the number of classes C

$$\sum_{i=0}^C p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i = 1, \text{ with } 0 \leq p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i \leq 1. \quad (2.25)$$

Therefore, we denoted by $P = \{p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i\}_{i=0}^C$ the probability distribution of the prediction. Correspondingly, the true class distribution is written by $Q = \{q(y^\mu | \mathbf{x}^{0,\nu})_i\}_{i=0}^C$. One point worth noting is the distinction between P and Q , which lies in the conditional \mathbf{w} of P while the components of Q consist of a one-hot vector, meaning that each sample belongs to one of the C classes. In light of this, a measure to calculate the loss function can be defined based on how one probability distribution P differs from Q in the point of view of information theory. Following this idea, one of the most natural distance between two probability distribution is the Kullback–Leibler (KL) divergence $K(Q||P)$ written by

$$\begin{aligned} K(Q||P) &= \sum_{i=0}^C q(y^\mu | \mathbf{x}^{0,\nu})_i \log \left[\frac{q(y^\mu | \mathbf{x}^{0,\nu})_i}{p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i} \right] \\ &= \sum_{i=0}^C q(y^\mu | \mathbf{x}^{0,\nu})_i [\log q(y^\mu | \mathbf{x}^{0,\nu})_i - \log p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i] \\ &= \sum_{i=0}^C \underbrace{q(y^\mu | \mathbf{x}^{0,\nu})_i \log q(y^\mu | \mathbf{x}^{0,\nu})_i}_{\text{independent of the parameter } \mathbf{w}} - \sum_{i=0}^C q(y^\mu | \mathbf{x}^{0,\nu})_i \log p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i. \end{aligned} \quad (2.26)$$

Note that the first term does not depend on the parameter of the model, hence it won't be required to incorporate it into the final KL expression, as we desire to minimize the KL distance with respect to \mathbf{w} . In other words, the learning algorithm in Machine Learning

consists in the minimization of the distance of the true probability distribution (Q) from the prediction one (P). Therefore, the error function \mathcal{L} between the prediction of the model and the ground truth label is expressed by

$$\mathcal{L}(Q, P) = - \sum_{i=0}^C q(y^\mu | \mathbf{x}^{0,\nu})_i \log p(y^\mu | \mathbf{x}^{0,\nu}, \mathbf{w})_i. \quad (2.27)$$

This expression is called the *cross-entropy* of the probability distribution Q relative to distribution P . Henceforth, we are prepared to apply the concepts already seen here about the optimization algorithm in artificial neural network, but now in spiking neural network.

It became evident that the error backpropagation algorithm arises naturally in the context of feedforward networks composed by artificial neurons units. To put it another way, the function composition and the continuity of the same is vital to compute the chain rule. Since now, we are dealing with neural networks with the same architecture but composed by spike neuron units, a new procedure must be defined in order to correctly compute the gradient of the loss function. More specifically, there is a need to focus on the temporal dynamics and the non-differentiability of spikes neurons. Methods for dealing with temporal dynamics of internal states of a network are already familiar and firmly established, considering the fact that this holds for RNN. The *error backpropagation through time* (BPTT) [60] algorithm evaluates the gradient of error function starting the chain rule calculation from the output layer and propagating backwards through the neural network by means the entire path of unfolded computational graph (Figure 2.6) of each unit in all previous layers. In figure 2.6 the forward computation consists of the sequential input $w_{ij}^{(l)} S_i^{(l-1)}[0], \dots, w_{ij}^{(l)} S_i^{(l-1)}[t_U]$ and the respective loss values $\mathcal{L}[0], \dots, \mathcal{L}[t_U]$. As a result, the *overall* loss function is calculated by summing all the losses at each time step

$$\mathcal{L} = \sum_{t=0}^{t_U} \mathcal{L}[t]. \quad (2.28)$$

Based on this, we are capable of formulating the partial derivatives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} &= \sum_{t=0}^{t_U} \frac{\partial \mathcal{L}[t]}{\partial w_{ij}^{(l)}} \\ &= \sum_{t=0}^{t_U} \frac{\partial \mathcal{L}[t]}{\partial S_j^{(l)}[t]} \frac{\partial S_j^{(l)}[t]}{\partial U_j^{(l)}[t]} \frac{\partial U_j^{(l)}[t]}{\partial w_{ij}^{(l)}} \\ &= \sum_{t=0}^{t_U} \frac{\partial \mathcal{L}[t]}{\partial S_j^{(l)}[t]} \frac{\partial S_j^{(l)}[t]}{\partial U_j^{(l)}[t]} \underbrace{\left(\sum_{s=0}^t \frac{\partial U_j^{(l)}[t]}{\partial U_j^{(l)}[s]} \frac{\partial U_j^{(l)}[s]}{\partial w_{ij}^{(l)}} \right)}_{\text{prior influence}}. \end{aligned} \quad (2.29)$$

We observe that this is almost identical to what was seen in the error backpropagation algorithm, with the only difference being the need to compute the *prior influence* caused by recurrent dynamics of neurons. This becomes more clear when we look closely at the term expressed by $\frac{\partial U_j^{(l)}[t]}{\partial U_j^{(l)}[s]}$ involving the influence of the membrane potential of all previous times. Therefore, using the chain rule, that term can be rewritten by

$$\frac{\partial U_j^{(l)}[t]}{\partial U_j^{(l)}[s]} = \prod_{t \geq i \geq s} \frac{\partial U_j^{(l)}[i]}{\partial U_j^{(l)}[i-1]} \quad (2.30)$$

as a product of multiple factors determined. Accordingly, the prior influence can be separate into long term and short term dependencies, where the first one refers to the significant contribution of terms $\frac{\partial U_j^{(l)}[t]}{\partial U_j^{(l)}[s]}$ for $s \ll t$, while the second one applies only to $k \approx t$. The long term has the potential to cause challenges in the optimization techniques depending on the value assumed of each component in that product. To be more precise, if each one displays a value close to zero, the result of that product may end up being close or even zero and it could *kill* the gradient, interrupting the gradient descent learning. The same happens to large values, where results in an unbounded increase in the norm of the gradient. Such issues are known as the *vanishing* and the *exploding* gradient problem [61], respectively.

The last step is in the application of the reverse-mode differentiation and the obtaining of the complete gradient over all layers is done in a simple manner. Still, attention should be given to the second term of equation 2.29. The non-differentiability of LIF neuron leads a significant problem. As illustrated in Figure 2.1b, the derivative of the Heaviside function exhibits the following behavior

$$\frac{\partial S_j^{(l)}}{\partial U_j^{(l)}} = \begin{cases} 0 & \text{if } U_j^{(l)} < \theta, \\ \infty & \text{if } U_j^{(l)} = \theta, \\ 0 & \text{if } U_j^{(l)} > \theta, \end{cases} \quad (2.31)$$

namely, a Dirac delta function. When the membrane potential assumes the threshold value, $U_j^{(l)} = \theta$, the computation of loss function grows indefinitely. However, most of the time, the membrane potential is set to zero as it rarely precisely reach the threshold. In this way, the gradient of the loss function will almost always be zero, and so, the learning will not take place, or rather, it will lead to the *dead neuron* problem.

One alternative for addressing the dead neuron issue involves substituting the derivative of the Heaviside with a differentiable function. As it is traditional to use S-shape curve to represent the step function, the derivative of a sigmoid function will be a good candidate for replacing the derivative of the Heaviside function. Although this procedure is an approximation of gradient of loss function, its calculation will be well performed and consequently the learning will unfold. The point here is that the substitution *exclusively* happens in the backward computation, there is no modification in the forward computation step. Again, the figure 2.1b displays the described operation. This technique is known by *Surrogate Gradient* and was popularised by [62]. Putting this solution into equations, we considered the following threshold-shifted sigmoid curve and its respectively derivative.

$$\tilde{S} = \frac{1}{1 + e^{\theta - U(t)}}, \quad \frac{\partial \tilde{S}}{\partial U} = (1 - \tilde{S})\tilde{S}, \quad (2.32)$$

where the replacement is done when the error backpropagation is being applied, i.e,

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \frac{\partial \mathcal{L}}{\partial S_j^{(l)}} \frac{\partial \tilde{S}_j^{(l)}}{\partial U_j^{(l)}} \frac{\partial U_j^{(l)}}{\partial w_{ij}^{(l)}}. \quad (2.33)$$

It is relevant to mention that surrogate gradient eliminates the problem of error propagation, but the weights are only updated when there is spiking activity. In the absence of activity, the neuron output is $S_j^{(l)} = 0$ and thus the middle term of equation 2.33 is zero.

Therefore, it is important to adjust a sufficiently large time so that the membrane potential dynamics emit spikes. Moreover, through empirical practices, it has been observed that the arctangent function as the most suitable choice to adapt the surrogate gradient [53]. Among the top candidates are triangular functions, fast sigmoid and sigmoid functions. In special, the arctangent and its derivative as a function of membrane potential is written respectively by

$$\tilde{S} = \frac{1}{\pi} \arctan(\pi U), \quad \frac{\partial \tilde{S}}{\partial U} = \frac{1}{\pi} \frac{1}{1 + (\pi U)^2}. \quad (2.34)$$

2.2.4 Implementing

Applying all the concepts outlined in the preceding sections, a fully connected spiking neural network, see Figure 2.7, was subject to supervised learning with the goal of recognizing images from standard datasets, the MNIST and the Fashion-MNIST. Both tasks had the same structure with respect to data set size, training and testing splits, composed of 60.000 training and 10.000 testing 28x28 gray scale images of 0 to 9 handwritten digits and of 10 categories of fashion products, as demonstrated respectively on table 2.1 and 2.2.

Hyperparameters were chosen according to table 2.3. The training section was composed by an epoch, a pass over the training set, where the neural network attained an accuracy above of 80% depending on dataset. Accuracy is measured by attribution the classification label to the output neuron with the highest firing rate, the spike count. Figure 2.9 shows the loss and accuracy as a function of the number of subsets (mini-batches) of previously randomly shuffled training set and test set. Training time refers to each interaction over mini-batches of the dataset. Training shows three distinct phases. A short initial phase where the network still cannot implement the task, a fast rising effective learning phase, and an almost saturated phase showing good performance. After training, the confusion table becomes practically a diagonal matrix, indicating that the model classification is most likely a true positive. Figure 2.10 shows the firing rates of the correct output showing different learning time thresholds for the different categories being learned.

To examine the performance of a classification algorithm in more detail, a confusion matrix can be construed. Confusion matrix is a table that shows all the numbers of true positive and false positive match according to the data labels. The Figure 2.9.C is illustrated the confusion matrix over MNIST test set after the training. On the matrix's diagonal, it is the true positive, outside it is the false positive. Thus, the more diagonal this table is, the less “confusion” the model has in classifying all the labels of images. According to this figure, we can check that the trained model is well optimized to recognize all digits.

Label	0	1	2	3	4	5	6	7	8	9
Sample										

Table 2.1: MNIST dataset labeled pattern.

Hyperparameters	
Neural Network	Neuron
$d = 28 \times 28$	1st Order Leaky Integrate and Fire Neuron
$N^{(1)} = 300, N^{(2)} = 300, N^{(3)} = 10$	$\beta = 0.5$
Batch size = 128	$\theta = 1$
Epochs = 1	$t_U = 25$
$\eta = 5 \times 10^{-4}$	
Optimizer: Adam	Reset Mechanism: Subtract
Accuracy metric: Spike Count	Backward pass: Fast Sigmoid
Loss: Cross Entropy Spike Count	

Table 2.3: Hyperparameters used on training. We used standard training for networks with $N^{(1)}, N^{(2)}$ neurons in the 2 hidden layers.


Label	0	1	2	3	4	5	6	7	8	9	
Sample											

Table 2.2: Fashion-MNIST dataset labeled pattern.

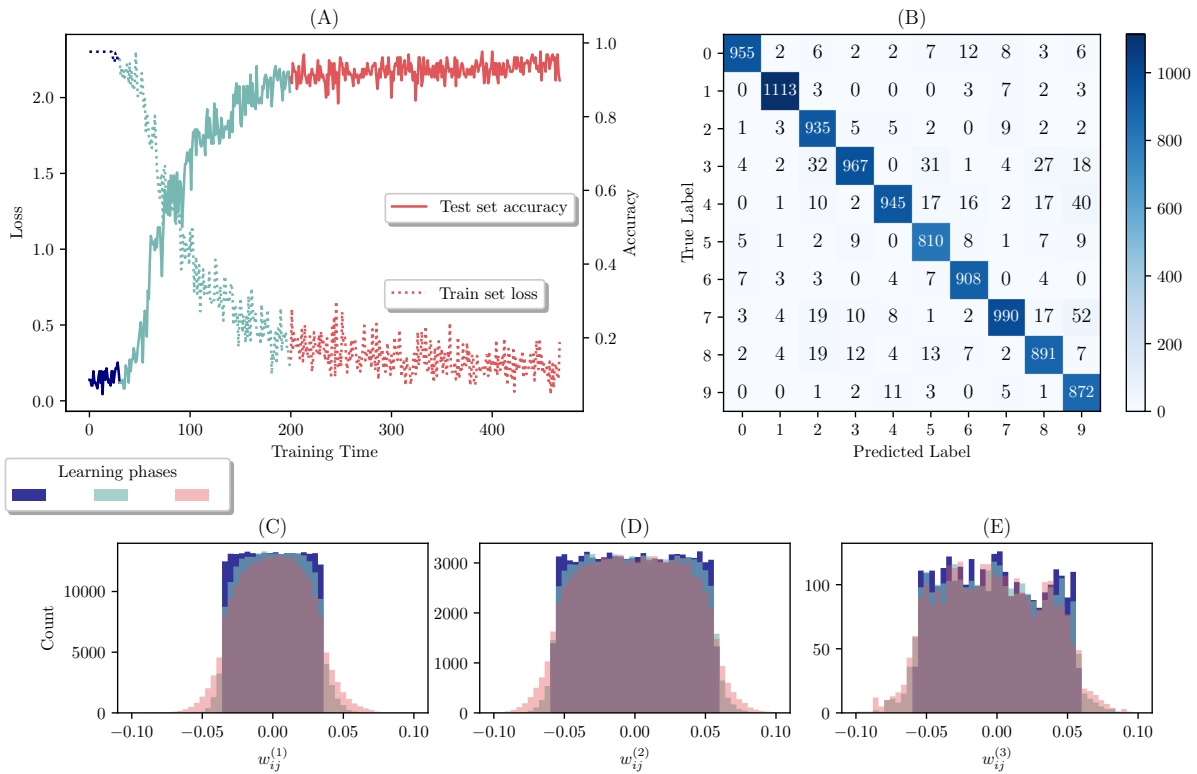


Figure 2.9: Training session over MNIST train and test set. (A) Loss (left axis) and Accuracy (right axis). Three different phases can be seen: an initial plateau or slowly rising accuracy phase, a fast rising and a saturated plateau. (B) Confusion matrix over the test set after the training. (C), (D) and (E) The evolution of the empirical distributions of weights between the layers, during training.

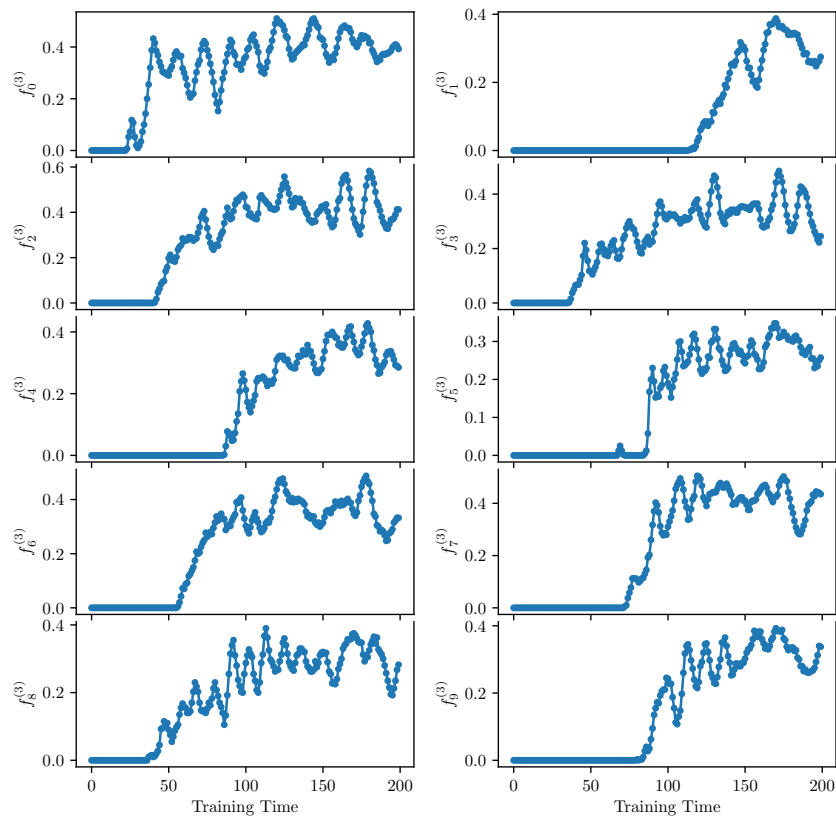


Figure 2.10: Evolution with training time of the firing rate $f_k^{(3)}$ of output neuron k when input patterns of type k are presented. Note that it takes different training times until the network can identify different categories. MNIST dataset.

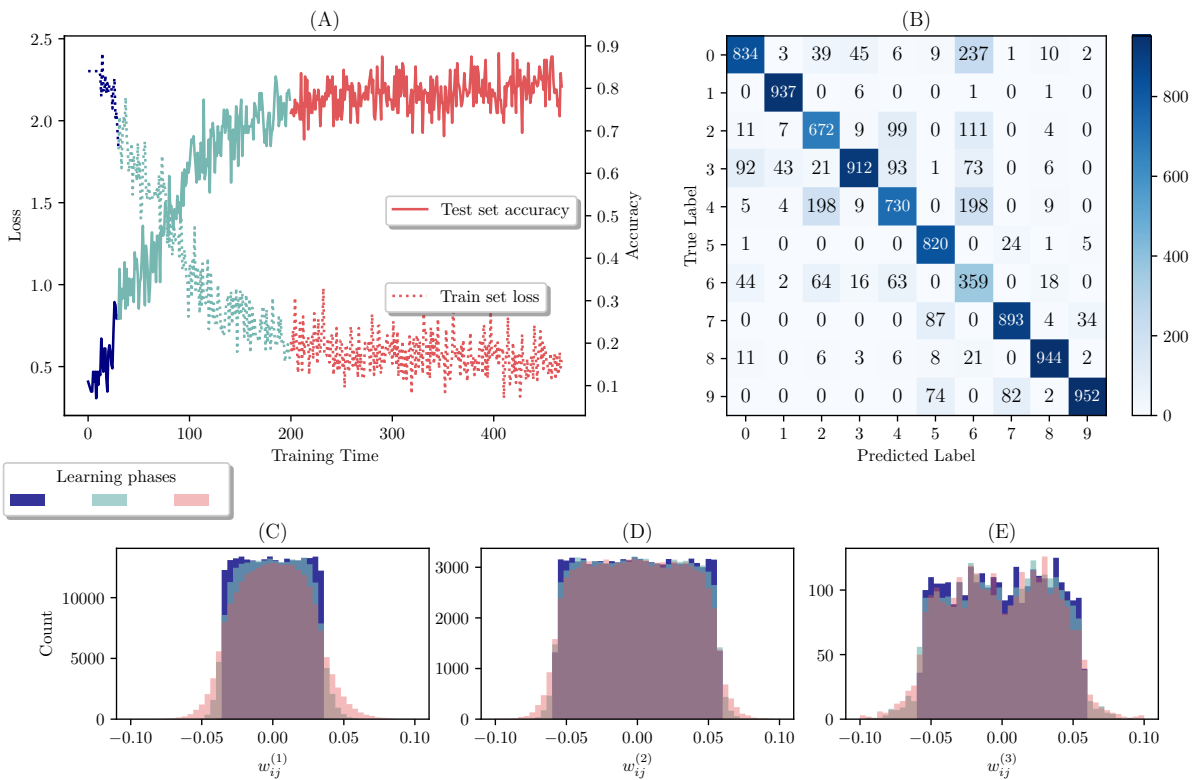


Figure 2.11: Training session over Fashion-MNIST train and test set. (A) Loss (left axis) and Accuracy (right axis). Three different phases can be seen: an initial plateau or slowly rising accuracy phase, a fast rising and a saturated plateau. (B) Confusion matrix over the test set after the training. (C), (D) and (E) The evolution of the empirical distributions of weights between the layers, during training.

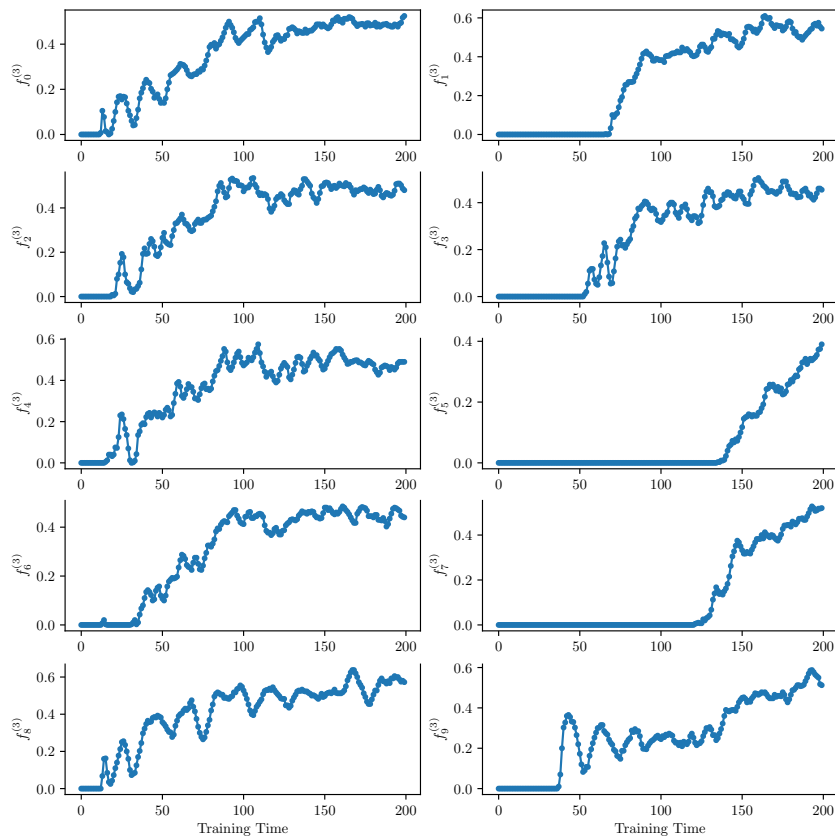


Figure 2.12: Evolution with training time of the firing rate $f_k^{(3)}$ of output neuron k when input patterns of type k are presented. Note that it takes different training times until the network can identify different categories. Fashion-MNIST dataset.

Chapter 3

Internal Representations Analysis

The most beautiful thing we can experience is the mysterious. It is the source of all art and science.

ALBERT EINSTEIN, Living Philosophies (1931)

Let's specify the relevant variables for our study. The activity in the l^{th} layer is described by a state vector of an $N^{(l)}$ -dimensional space at time t

$$\mathbf{S}^{(l)}(t) = [S_1^{(l)}(t), \dots, S_{N^{(l)}}^{(l)}(t)], \quad (3.1)$$

where $N^{(l)}$ is the number of neurons and $S_i^{(l)} = \{0, 1\}$, the binary output. Also, we can define the following macroscopic variables (aggregated): the activity density in the l^{th} layer at time t

$$\rho^{(l)}(t) = \frac{1}{N^{(l)}} \sum_{i=1}^{N^{(l)}} S_i^{(l)}(t), \quad (3.2)$$

and the membrane potential density in the l^{th} layer at time t

$$u^{(l)}(t) = \frac{1}{N^{(l)}} \sum_{i=1}^{N^{(l)}} U_i^{(l)}(t). \quad (3.3)$$

The characterization of internal representations will be done by measuring the time series of activity fluctuations. Thus, we use the following notation for the time average at time t of any quantity f in a time window of width Δt

$$\langle f(t) \rangle = \frac{1}{\Delta t} \sum_{t'=t}^{t+\Delta t} f(t'). \quad (3.4)$$

When f , for fixed t , is an array, so is the above expression and the sums are point wise. The fluctuation of a quantity around its average is $\bar{f} = f - \langle f \rangle$. The un-normalized truncated correlation is

$$c_{f,g}(t, \tau, \Delta t) = \left\langle \bar{f}(t) \overline{\bar{g}(t - \tau)} \right\rangle, \quad (3.5)$$

$\bar{f} \bar{g}$ is a dot product for arrays, and the normalized correlation of fluctuations or the truncated correlation is

$$C_{f,g}(t, \tau, \Delta t) = \frac{c_{f,g}}{\sqrt{c_{f,f} c_{g,g}}}. \quad (3.6)$$

Table 3.1: The “movies”

Movie type	Composition	Description	δt	Measures
M_0	Random inputs	New random image every time step	none	ξ_τ
M_S	2 or more classes	Few slow transitions between classes	$\delta t \approx 400-500$	ξ_τ
M_F	2 or more classes	Several fast transitions between classes	$\delta t \sim t_U \sim 25$	Avalanches and Complexity Index

For M_S and M_F , δt is the number of presentations of random choices of examples in the same class between transitions. It measures the variability of the environment presented to the NN.

In the simulations we measure $C_{\mathbf{S}^{(l)}\mathbf{S}^{(l)}}(t; \tau, \Delta t)$, $C_{\rho^{(l)}\rho^{(l')}}(t; \tau, \Delta t)$ and $C_{u^{(l)}u^{(l')}}(t; \tau, \Delta t)$, functions of the three parameters t , τ and Δt . Significantly, our objective is to explore how temporal correlation of fluctuations changes with different delays values τ , since the detection of *long-range* temporal correlation carries significant consequences. From an analogy to spins systems, we expect these correlations to decay with τ as the product of a fast decaying exponential with characteristic time ξ_τ and a slow decaying algebraic function. If criticality is approached ξ_τ grows and there is a crossover to purely algebraic decay, but only in the thermodynamic limit as shown in Figure 1.3. For finite systems an exponential function is a good model, and we consider representing the broad-tails distribution

$$C(\tau; t, \Delta t) \sim \exp \left\{ -\frac{\tau}{\xi_\tau} \right\}, \quad (3.7)$$

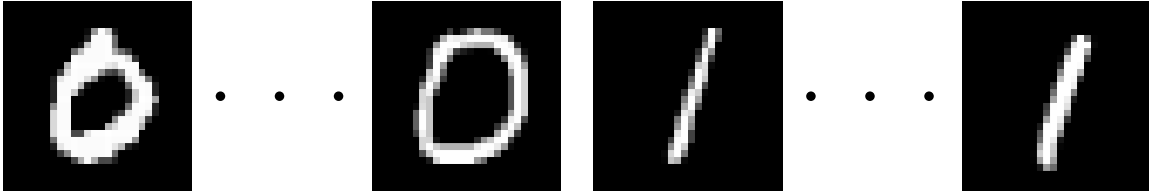
where ξ_τ is the correlation time, a rate of relative decrease of temporal correlation as the distant of two temporal points in the time series increases. Moreover, we would like to stress the parameters of the temporal correlation in equation 3.7. Fixing t and Δt , the decay of the temporal correlation can see in function of τ . As a result, for an indexed time phenomena, the correlation time can be computed for each time step t . Besides of that, fixing Δt , we can also observe how the temporal correlation behaves in relation to τ and t . By doing so, we gain a comprehensive overview of the temporal correlation decay over time.

We use “movies”, sequences of images described in Table 3.1 as inputs to the SNN under a condition of *free dynamics*. This means that the recurrent dynamics of membrane potential is imposed by hand as the input change. The set of numerical experiments is shown in Table 3.2. α is a parameter introduced to simulate partial blocking of the input to the network. Every pixel of an input pixel is multiplied by α in the interval $(0, 1]$. It is a simplistic model of the effect of anesthesia acting only on the receptors of input layer. It is used just to study the pattern recognition of a dim input and not as a model of the effects of a particular type of anesthesia itself.

Table 3.2: Simulations.

Training Phase	Movie	Conditions
early	M_0	$\alpha = 1$
intermediate	M_0, M_S, M_F	$0 < \alpha \leq 1$
saturated	M_0, M_S, M_F	$0 < \alpha \leq 1$

The intensity of the “anesthesia” is $1 - \alpha$.

Figure 3.1: A movie of M_S type.

3.1 Label Transition

We explore the following question: what do we observe in activity of the internal layers of a trained spiking neural network performing a recognition task when there is a significant change in the external input? At this moment, significant change means a classifiable label change in the film presentation. In Figure 3.1, we see a sequence of images of the label ‘0’ and suddenly a sequence of images of the label ‘1’, a movie of M_S type. For intervals comprising ~ 500 frames random examples of one of the ten categories is shown, then for another ~ 500 frames, examples from another category are presented. Nothing interesting happens except for a small region in time starting at the transition with large and persistent fluctuations, see Figure 3.2.

In Figure 3.3 we show result from a typical run with several input class changes. The neural network is working satisfactorily as shown in Figure 3.3.A, since the output cell with the dominant $U^{(3)}$ is the correct one. Figure 3.3.B shows a very large increase in ξ_τ following the input label transitions. A similar behavior, with same peaks, is seen for the IR in the first hidden layer. While the neural network moves rapidly toward the new classification, IR show persistent fluctuations, around the transitions. However, is this peak in correlation time appears for all label transition?

In Figure 3.4, we show the maximum correlation time in the interval of a slow transition among all label combinations from the dataset. Although one would expect a matrix more homogeneous, the one for spike activity $\mathbf{S}^{(l)}$ is much more filled by correlation time for both hidden layers. This has an interesting answer, since we are computing the temporal correlation through the expression 3.5, an element-wise product between state vectors of layer is being performed. Therefore, realizing that operation, we are not missing any feature of neurons activity, since averages represent the state of a system with large N units by a single number. However, this results can be interpreted as follows: we should not focus on individual cases of label transitions, but rather observe all transitions of all measured variables. Each variable represents a part of the whole picture, so if we don’t observe a correlation time for the membrane potential variable, we may see it for the spike activity, and so on. Finally, we did the same for the Fashion-MNIST, see Figure

3.5. In this case, the presence of correlation time in transitions is much more evident for all variables. That can be associated with the complexity and richness of this dataset, it is no wonder that Fashion-MNIST is used as a benchmark for machine learning models, see Figure 2.11.

In light of these results, we can draw some preliminary conclusions that will be useful going forward. Firstly, we observed that a long decay in temporal correlation emerges when we change the labels of the images. Secondly, the characterization of these long-tail distributions is not the same for each category transition, meaning that some exhibit longer or shorter correlation times. Therefore, we can consider that the dataset labels are represented as different basins of attraction. As we change the input category, the system converges to a new attractor. To clarify further, for example, using t-Distributed Stochastic Neighbor Embedding (t-SNE) [63], a popular dimensionality reduction technique, we can visualize how each class of the dataset is spread in a two- or three-dimensional space, useful for humans to interpret data geometrically. In Figure 3.6, we notice that the distances between each MNIST label are different. This also confirms our result. Different values of correlation time lead to the conclusion that the basins of attraction have different distances. It is important to note that this is not the only possible visualization, there are many ways to apply dimensionality reduction methodology, whether preserving linear structure (principal component analysis, PCA), global geometry (multidimensional scaling, MDS), or topology (t-SNE).

One last comment remains unexplained. It is interesting to note that all transitions of the MNIST involving category “1” possess a very pronounced peak in correlation time. Additionally, this is the last label to be learned in the training process, as seen in Figure 2.10. Finally, looking at the confusion matrix, Figure 2.9 (B), it is the label with the best classification. Using other clustering methods, known as Distributional Reduction (DistR) [64], we were able to observe that label “1” has a significant distance from the other labels, see Figure 3.7. Perhaps this could tell us that there is no smooth and short interpolation involving images of label “1” and the rest [64], meaning that strongly correlated activity emerges as we move towards the attraction basin of class “1”. However, this is not verified in Fashion-MNIST dataset.

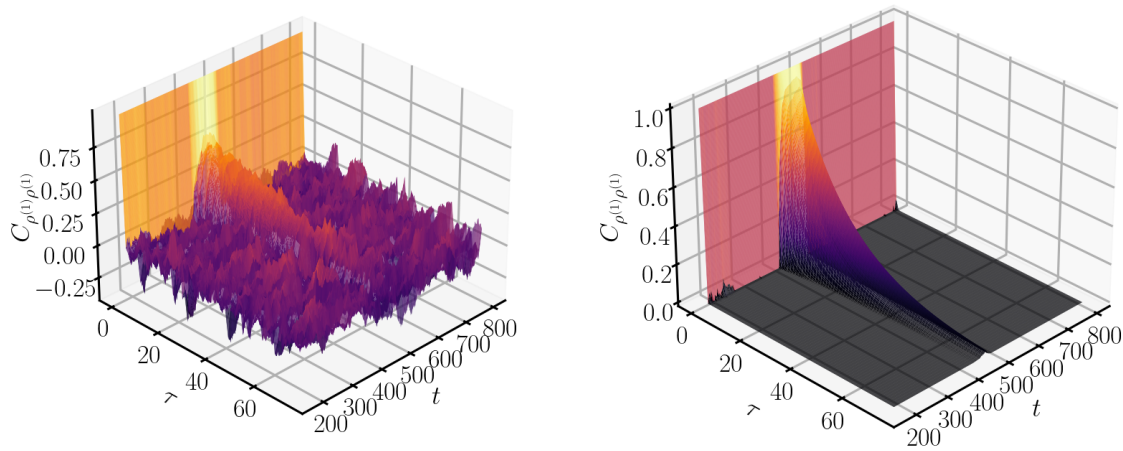


Figure 3.2: IR fluctuations temporal correlation as a function of τ measured on a sliding window centered at time t for the single image change movie. Left: the raw measured correlations. Right: For each t , the best fit of an exponential decay $e^{-\frac{\tau}{\xi_\tau}}$, equation 3.7. At the region of the transition of images, the decay time ξ_τ increases a few orders of magnitude

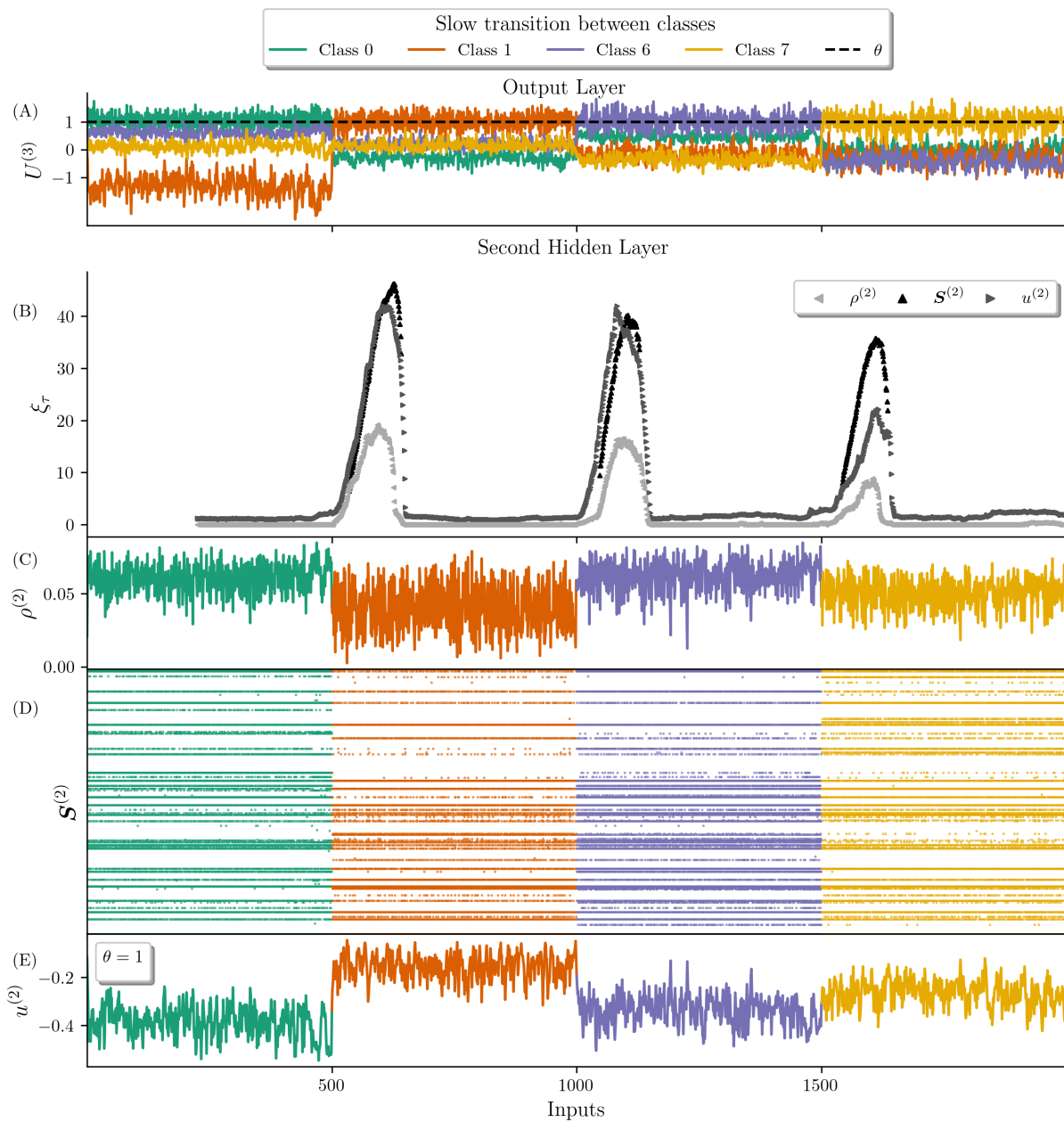


Figure 3.3: The M_S movie input (no anesthesia $\alpha = 1$) contains a sequence of 500 random images of label "0" (green), then 500 of label "1" (orange), 500 of label "6" (purple), and 500 of "7" (yellow) from the MNIST database. (A) The membrane potential of output neurons responsible for classification of labels during the movie, (B) The characteristic time of decay of correlations of the fluctuations of the average activity, of spikes and average membrane potentials of the second hidden layer. (C) Average activity. (D) Spike train: raster plot of a few neurons of hidden layer 2. (E) Average membrane potential.

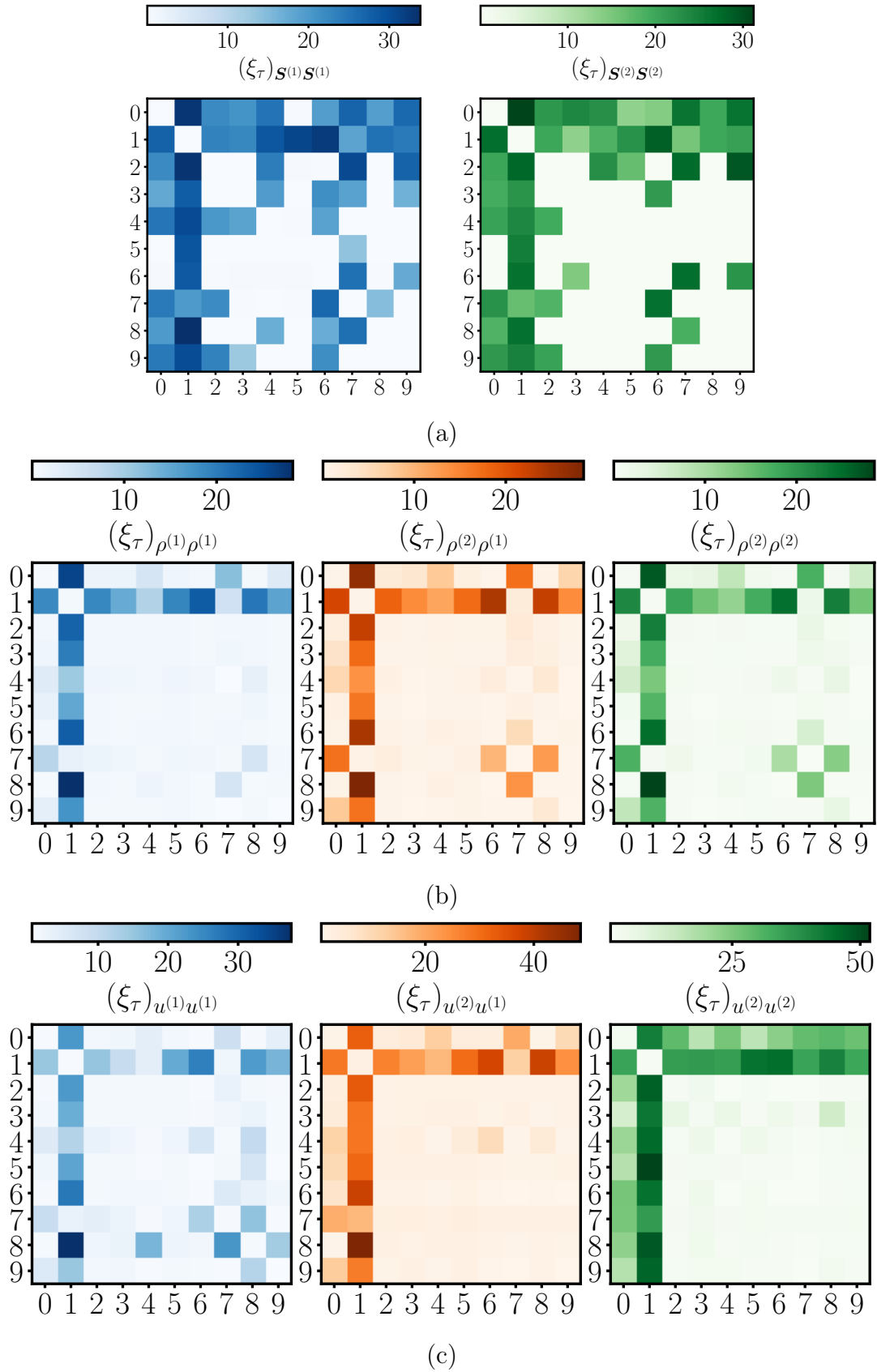


Figure 3.4: Correlation time for label transition matrix over MNIST dataset. (a) Correlation time in the activity $S^{(l)}$. (b) Correlation time in the activity density time series. (c) Correlation time in the membrane potential density time series.

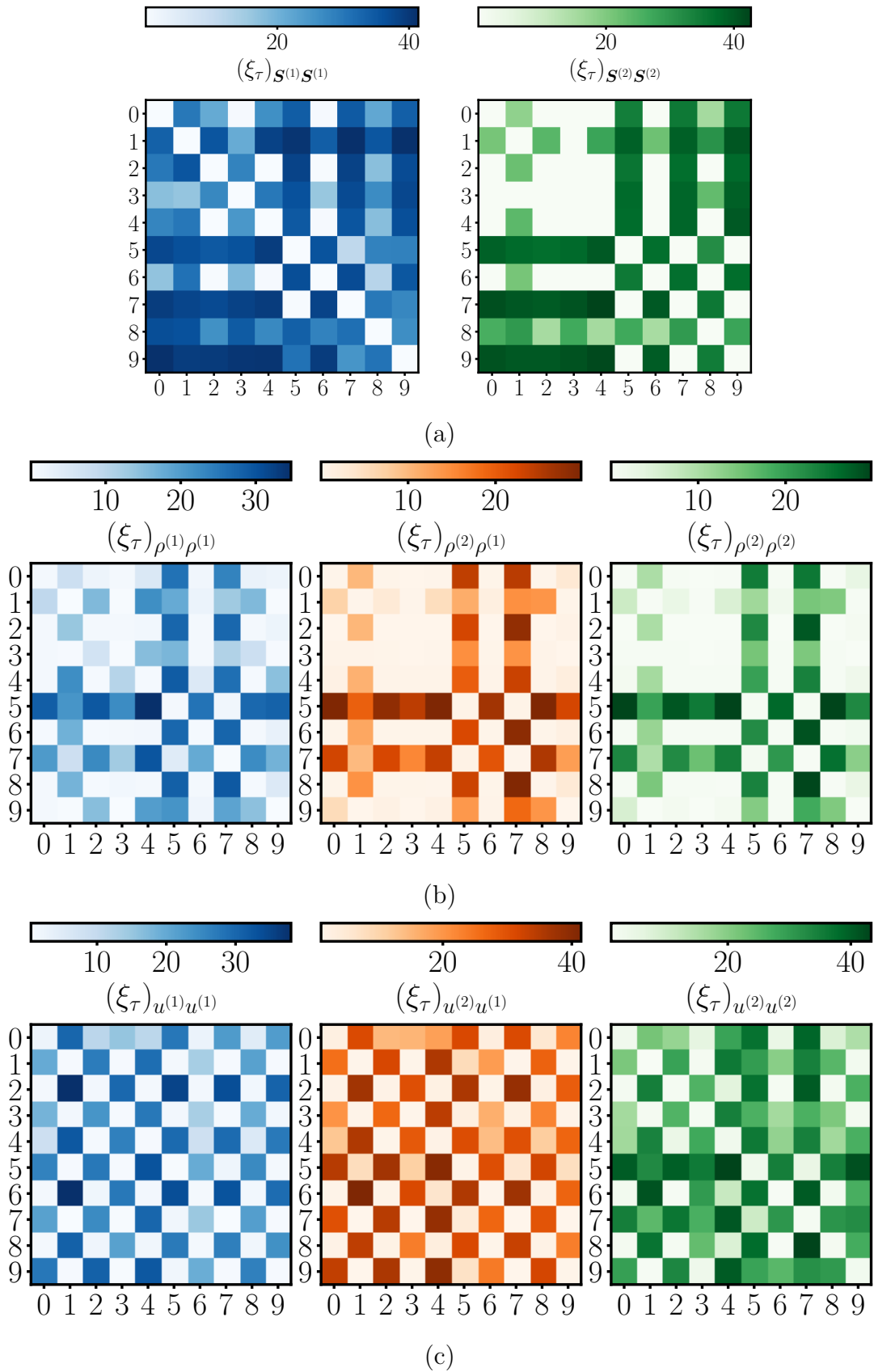
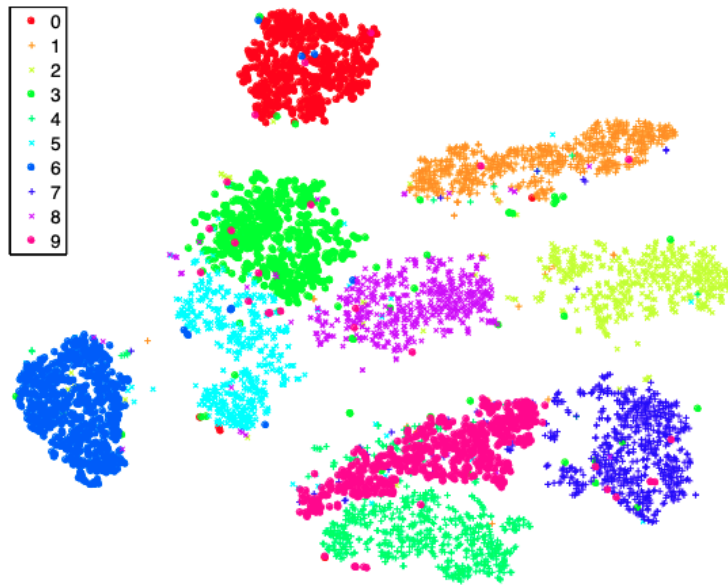
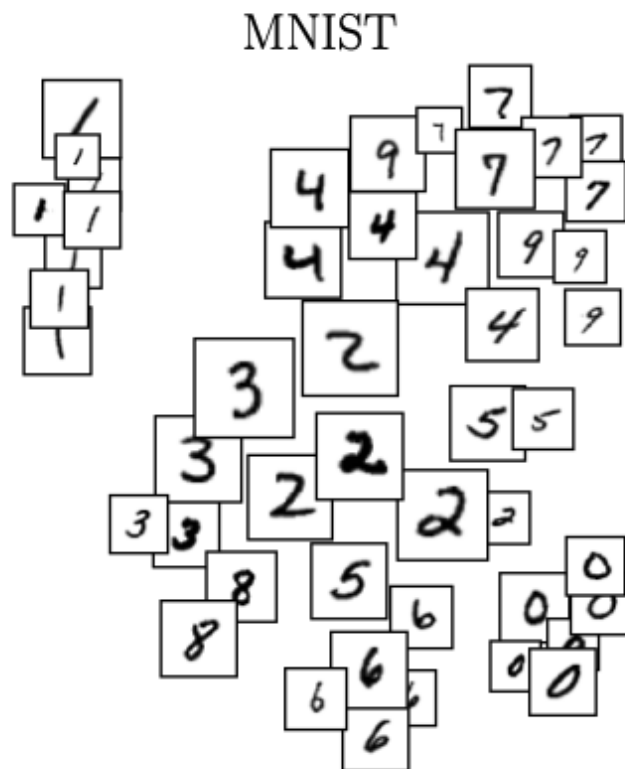


Figure 3.5: Correlation time for label transition matrix over Fashion-MNIST dataset. (a) Correlation time in the activity $\mathbf{S}^{(l)}$. (b) Correlation time in the activity density time series. (c) Correlation time in the membrane potential density time series.



(a) Visualization by t-SNE.

Figure 3.6: Visualization MNIST by t-SNE. Image and caption from reference [63].

Figure 3.7: Example of 2-dimensional embeddings produced by DistR using the SEA (Symmetric Entropic Affinity) similarity for C_X (input similarity matrix) and the Student's kernel for C_Z (embedding similarity). Image and caption from reference [64].

3.2 Learning Process

Given a previously trained model, persistent fluctuations appear when there is a change of label in sequential inputs. Is the signature of long-tailed distributions exclusive to classifiers in saturated training phase? At this moment, we aim to elucidate the relationship between the long-range correlation in a slow transition between labels and the learning process. In special, we show results in Figure 3.8 for a training time step of 25, when as Figure 2.10 shows, only category “0” is partially recognizable in the output layer. It may be expected that since the category attractors are not yet fully developed, persistent fluctuations should not appear. However, despite not achieving correct classification over all data labels, such persistence occurs during the transition of input categories. This is observed in distinct cases, first during the transition of a correctly classified input to one not yet learned (“0” \rightarrow “1”). Second, when the neural network still has not learned both input labels (“1” \rightarrow “6” \rightarrow “7”). This is evidence that during the learning process, attractors of the dynamics are partially present in the hidden layer, while still sub-threshold in the output layer. In other words, despite the low accuracy in the classification performed at the output layer, the internal layers are conducting significant information processing.

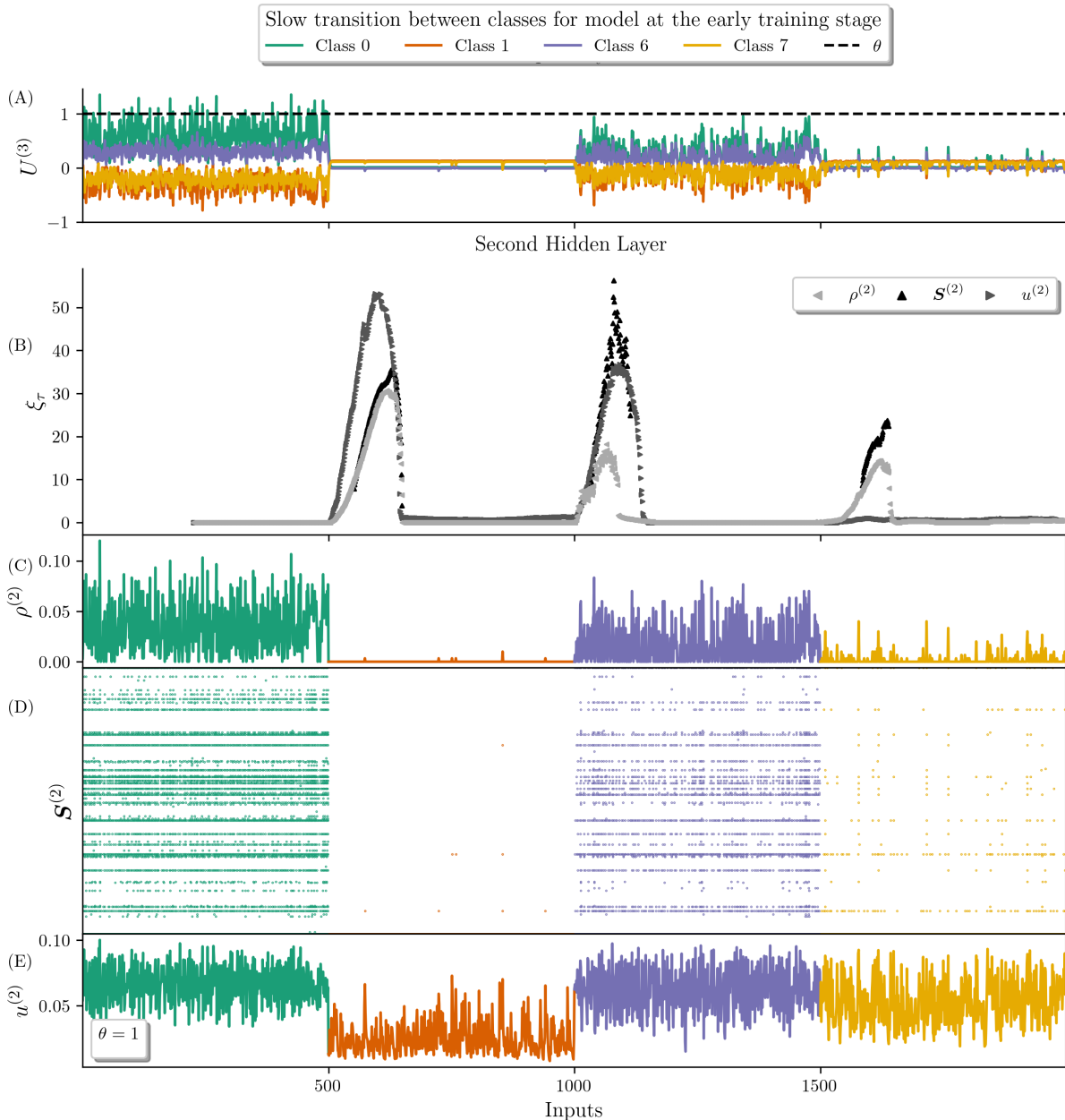


Figure 3.8: Same as Figure 3.3 but for a partially trained, where only “0” is recognizable. Transition regions show persistent fluctuations, independently of output recognition.

3.3 Anesthetized Perception

Another experiment consist on sensory perception under a very simple model of the effect of “anesthesia”. A movie of M_S type is presented again to a trained neural network, but now the stimulus strength of the external input is decreased (multiplied by a factor $\alpha \leq 1$) until eventually the model ceases to classify correctly the input. α can be interpreted in two ways, first as the fraction of input neurons that have not been blocked by an anesthesia, or second as the brightness (or intensity) of an image. Figure 3.9 shows again that for inputs on which the output is not correct due to diminished input sensibility, the sub-threshold images can elicit persistent correlations with large characteristic times.

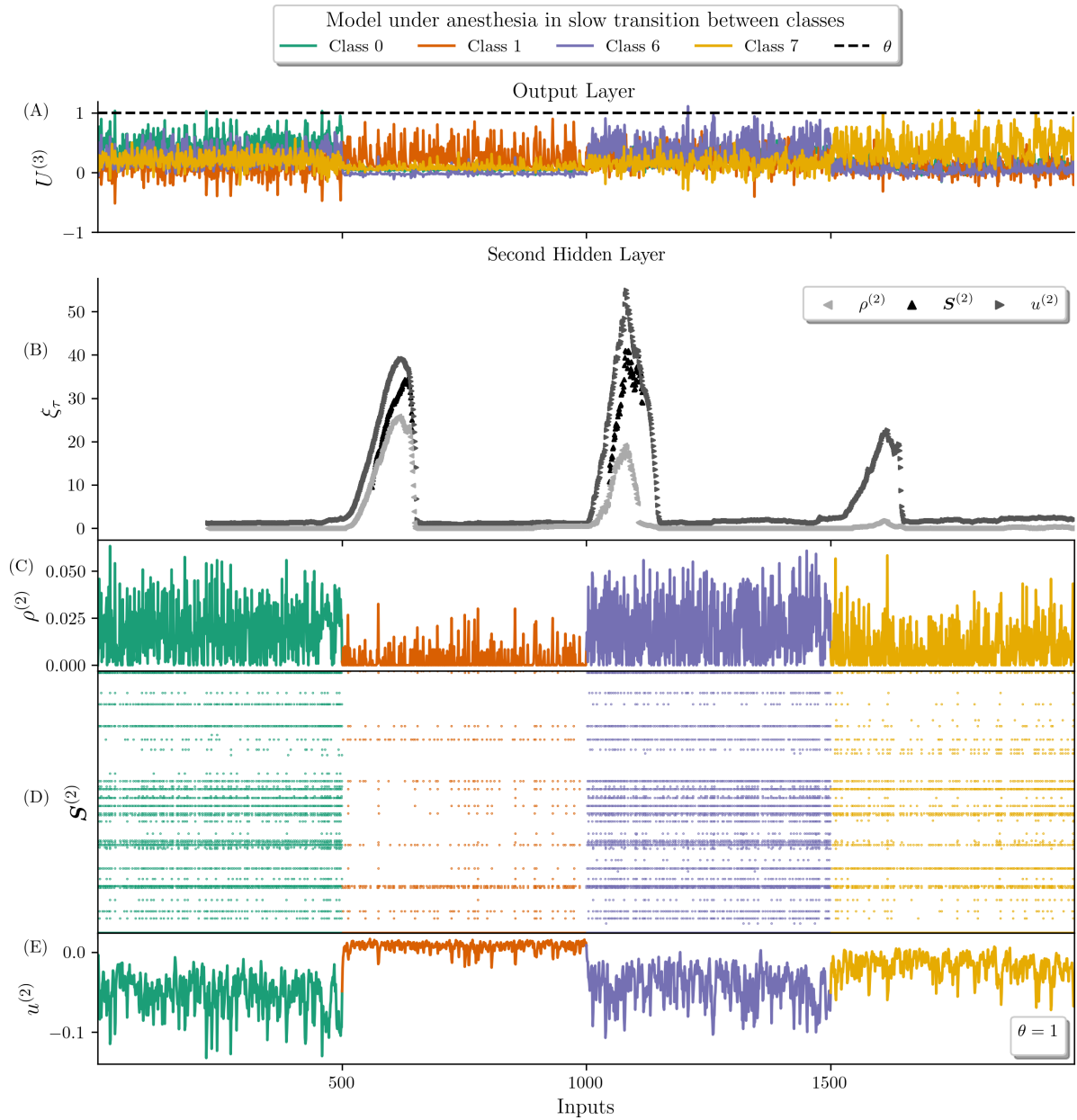


Figure 3.9: Same as Figure 3.3 but with neural network under effect of anesthesia, $\alpha = .4$, with an M_S input. (A) The network is not able to correctly categorize the inputs, which generate sub threshold activity at the output layer: there is no communication of the computation to the outside world. (B) Nevertheless, the transition of inputs generates persistent fluctuation correlation. (C, D) show the activity in the second hidden layer, but (E) Average membrane potential $u^{(2)}$.

This result can be interpreted by drawing on ideas from studies in the field of neurobiology. The state of coma in a person can be attributed as a state where there is no behavioral report, but relevant activity within the internal states of the brain is detected. Bringing this into our experimental context, the neural network does not exhibit any response in the output layer, but as there is detection of image labels in the hidden layers (peaks in correlation time), we can assert that, similarly to biological neural networks, artificial neural networks can perform “unconscious” pattern recognition. Although this may be a crude equivalence, the output layer can provide one more contribution to this argument. Note that all activity is slightly below the firing threshold. As we increase or decrease the effect of anesthesia, the output layer will respond according to each anesthesia condition. Moreover, we can further add to our argument the dynamics of the membrane potential of the output neurons. Notice that as we change the label of the image, correlated fluctuations persist. Put differently, correlated sub-threshold oscillations are evident in the final layer, namely the recognition layer.

3.4 Random Inputs

To ensure that the increase of the characteristic time of decay of temporal correlation is a signature of labeled input transitions, we need to examine the network activity response in the presence of random inputs not associated with any specific dataset category. In Figure 3.10 shows results for a combination of movie of type M_0 and some random choices of examples in the same class of dataset. The green color represents the random images, in which each pixel follows a uniform distribution between 0 and 1 $\mathcal{U}(a = 0, b = 1)$. We notice there is no presence of correlation time in the temporal series of activity when the random images of film is being exposed. However, long-range correlation emerges during the transition from random to labeled input. This has a similar interpretation to previous cases, where the search for an attractor basin leads to persistent activity.

Now, let’s conduct another similar experiment. A movie composed of random images of distinct probability distributions will be presented. Gaussian and uniform distributions were chosen for simplicity. In short, we will compile a movie consisting of 500 images from each distribution. Figure 3.11 and 3.12 display the results for the first and second hidden layer, respectively. As can be observed, there is no increase in the correlation time when we alter the probability distribution of random images. This suggests that activity persistence only occurs when we are in a scenario where significant changes in the external environment take into account only labeled inputs within the dataset in question. Additionally, it is observed that certain moments in the average membrane potential temporal series of the first layer exhibit a non-zero characteristic time. This implies that the first layer may be more sensitive to network inputs, but there is no indication that this sensitivity is associated with significantly different inputs (different distributions).

Finally, we examined the classification of random images done by the output layer. Despite images having different probability distributions, the neural network provides the same response, often classifying random images as “2” or “3”, and occasionally as “8”. This may be associated with the size of the basin of attraction of these labels, since for any random input, its classification is the same. Leading that the peak of correlation time only appears when the recognition of the input changes.

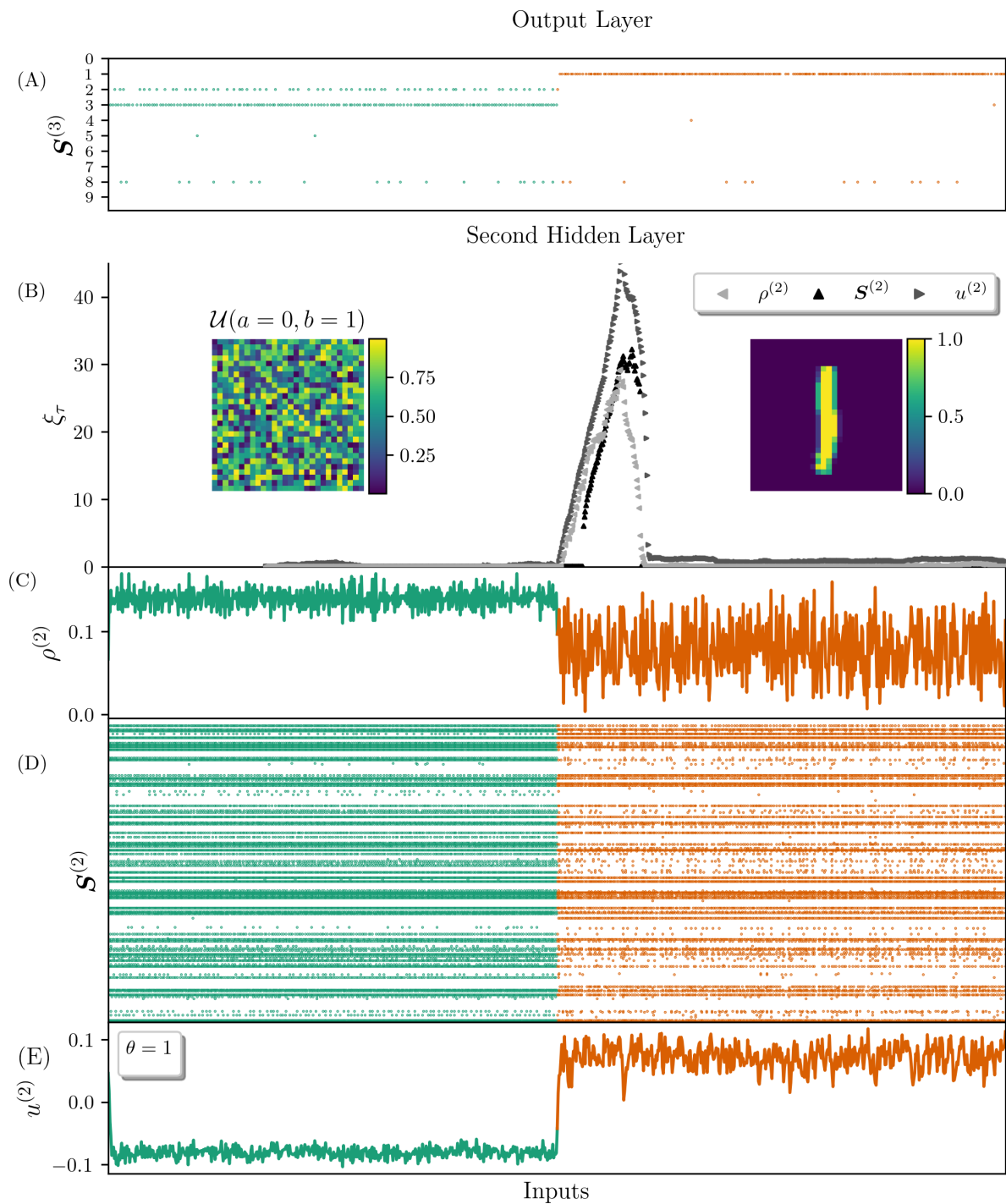


Figure 3.10: The movie input (no anesthesia $\alpha = 1$) contains a sequence of 500 random images (M_0 type) following a uniform distribution (green) and 500 of "1" (orange) from the MNIST database. (A) The membrane potential of output neurons responsible for classification of labels during the movie, (B) The characteristic time of decay of correlations of the fluctuations of the average activity, of spikes and average membrane potentials of the second hidden layer. (C) Average activity. (D) Spike train: raster plot of a few neurons of hidden layer 2. (E) Average membrane potential.

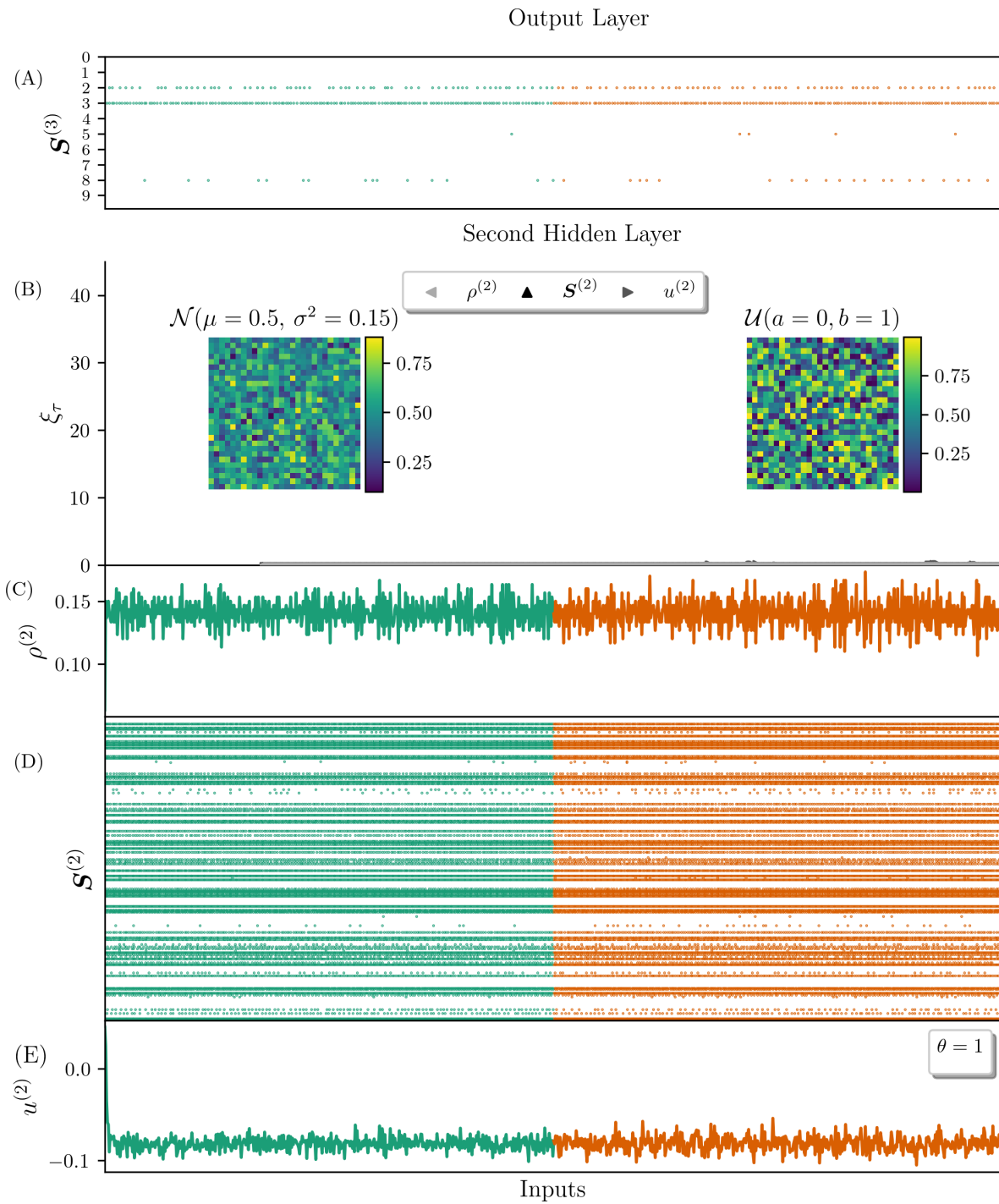


Figure 3.11: The movie input (no anesthesia $\alpha = 1$) contains a sequence of 500 random images (M_0 type) following a Gaussian distribution (green) and 500 random images (M_0 type) following a uniform distribution (orange). (A) The membrane potential of output neurons responsible for classification of labels during the movie, (B) The characteristic time of decay of correlations of the fluctuations of the average activity, of spikes and average membrane potentials of the second hidden layer. (C) Average activity. (D) Spike train: raster plot of a few neurons of hidden layer 2. (E) Average membrane potential.

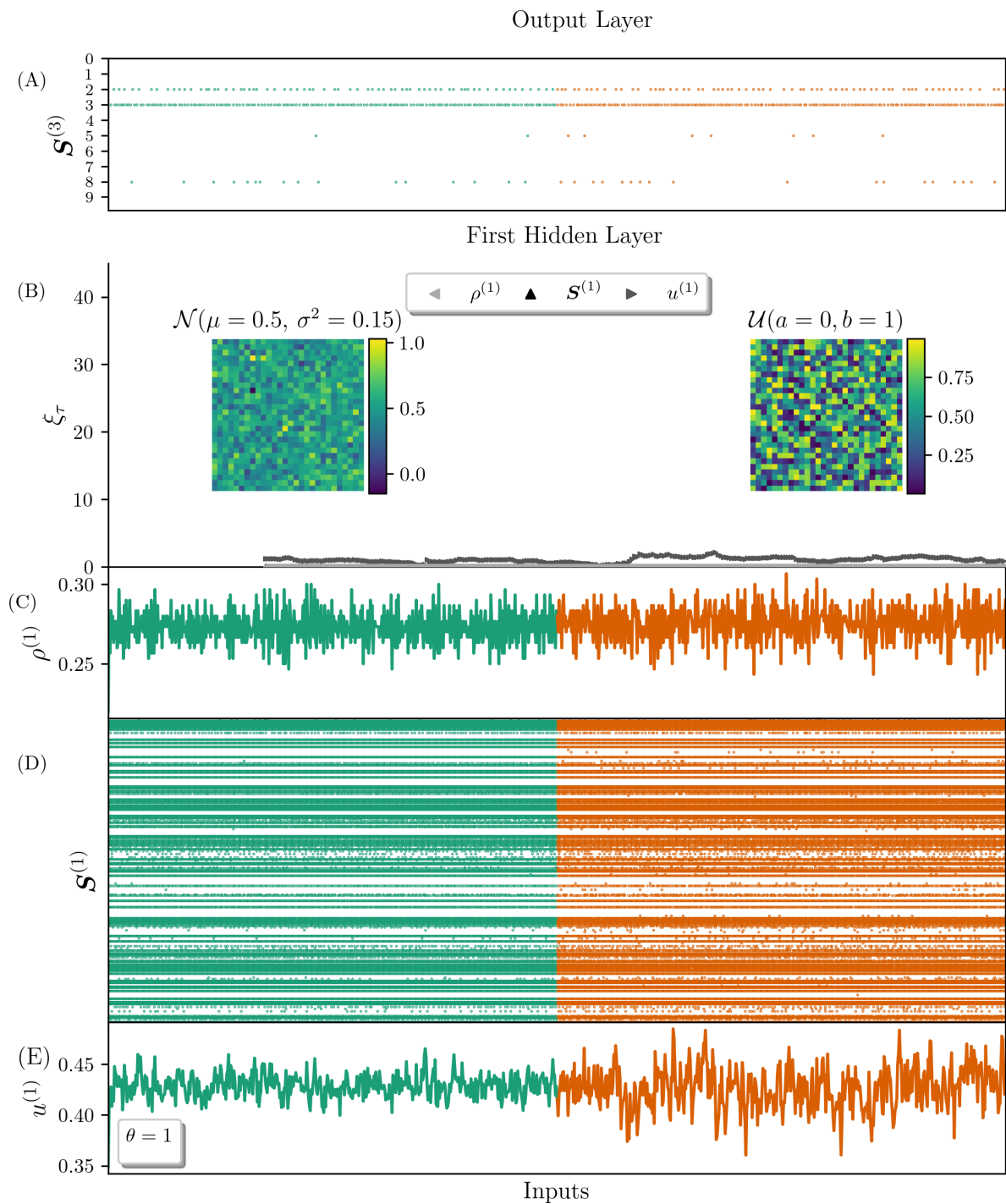
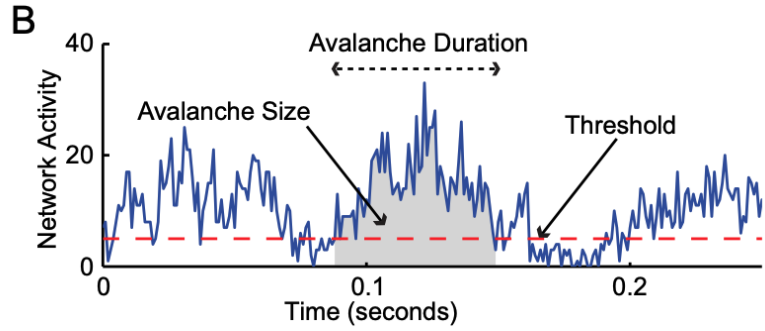


Figure 3.12: Same as Figure 3.11, but for the first hidden layer.

3.5 Avalanches

In a context of large systems composed of simple units subject to constant external inputs, an avalanche can be defined as a fast chain reaction involving these units between long periods of silence to bring the system to a new state of equilibrium. This means that there is a separation of timescales between the lifetime of avalanches and the inputs made by the external environment. As mentioned earlier, for a complete understanding

Figure 3.13: An avalanche starts and ends when integrated network activity crosses a threshold value. Image taken from reference [65].



of avalanche dynamics, a boundary of dissipation and volume conservation must be taken into account. Therefore, to establish a good definition of avalanches, we must integrate these elements into our artificial neural network context.

First, we should note that, in contrast to numerous models, in our case, spontaneous activity is absent, as activity is only observed when inputs are presented to the network. As an immediate implication, the notion of relaxation time present in models explained by SOC is not present. This means that the activity for our model does not lead to any equilibrium state, since in a supervised learning process the objective is only to characterize the images of the dataset into categories. Therefore, it becomes clear that the classical definition of avalanches will not be exhibited by our neural network, since we do not have stable or unstable configurations. Thus, we must seek a better understanding of how avalanches can be incorporated into our situation.

Before that, let's recall that criticality revolves around detecting power-law distributions for the relevant variables of our problem. For example, we say that the system operates at the critical point if the avalanches of a system with the characteristics mentioned above follow a power-law. As detecting peaks in the characteristic time of internal activity leads to the distribution of long tails in regimes when label transitions are slow, it is evident that we must delve deeper and ask ourselves how we could make a more refined statistical characterization of these correlated fluctuations and link them with criticality. We can proceed as follows: avalanches in our context will be restricted to the analysis of neural network activity. Inspired in the previous work [65], we say that a ρ avalanche in the hidden layers, is happening when $\rho(t) = \sum_l \rho^{(l)}(t)$, summed over hidden layers, is above a threshold \bar{R} . An avalanche's size is the integrated area below $\rho(t)$ between the crossings to $\rho > \bar{R}$ and back to $\rho < \bar{R}$, and the duration is the difference in time between those consecutive occurrences, as illustrated in Figure 3.13. Through this definition, it is evident that avalanches are subject to the chosen threshold value, and as a result, their size and duration distribution as well. What can be considered is selecting a range of thresholds where the distribution of avalanches is very similar. This gives us confidence in the statistics that govern the network activity. Based on reference [65], the average of activity will be used as the threshold \bar{R} .

As our model has finite size, signs of criticality are not just about looking in the distribution of avalanche size and duration. It is important to keep in mind that each distribution observed is linked to a specific network size. One consequence is that the larger the number of neurons in the layer, the greater the maximum size of the avalanche present in the temporal series of activity. Known as a cutoff, this will follow a power law with respect to the network size if criticality is present. Taking this into consideration, we use the M_F movie to generate several fast input transitions so that the final output alternates between different categories, to simulate a free moving agent in a rich environ-

ment. This input generates a large amount of variability in the internal representations and avalanches will be measured. Histograms are shown in Figure 3.14.A and B. While we kept the number neurons in the hidden layers equal $N^{(l)} = N$ for all hidden layers, we ran simulations with different values of N (50 and from 100, 200, \dots to 1200).

Each point in Figure 3.14.C shows the duration and size of an avalanche. For the characterization of power laws we use a standard procedure [66]. The size and duration exponents τ, τ_t give a crackling exponent [38]

$$\gamma = \frac{\tau_t - 1}{\tau - 1} \approx 1.44. \quad (3.8)$$

A direct fit of a power law in Figure 3.14.C yields a different $\gamma^* \approx 1.15$. The uncertainty of the fits, from [66] are too optimistic, being rather of the order of .2, not 0.03. It is not possible to affirm that power laws are present in this case. Note that the procedures that yield these exponents are very sensitive, see e.g. [67], not only to choices of maximum and minimum values, but also to sub-sampling and to the noise process that is used to model the fluctuations in the data. Since we are not defending this to be a signature of criticality we don't delve into the details of the fit. If however the reader finds this a compelling argument for criticality, which we don't, it is clearly not in the class of directed percolation.

Finally, we also tried to fit the avalanches distributions when the neural network was not trained, i.e, when the accuracy is zero, as a control experiment. The result is shown in Figure 3.15. In this situation, we are not able to fit a power law distribution, indicating that heavy-tail distribution is associated with optimal information processing.

3.6 Perturbational Complexity Index

Perturbational Complexity Index (PCI) is an empirical complexity-related index of brain's information processing. Based in the early Tononi's studies about the relationship between consciousness and complexity [68], its purpose is to define an objective measurement of conscious experience. In short, PCI calculation is done by two steps: first, perturbing the cortex through a transcranial magnetic stimulation (TMS) and second, a complexity algorithm to measure the recorded spatio-temporal neural activity of the brain. Compressing the *binary* matrix of significant source $SS(x, t)$ ¹ by means of the normalization of the Lempel-ziv complexity, the typical PCI curve increases monotonically in time, illustrated in Figure 3.16. The temporal evolution of PCI curve shows that it is possible to distinguish consciousness and unconsciousness patients. It can be done by two ways, using the rate of change of PCI, or we could take the stationary PCI value at the end of stimulus and associate each number of conscious experience to a complexity marker.

Why did we choose PCI? Well, consciousness is inherent a subjective experience and the definition of a marker for objective determination of conscious state is beyond interesting. The last argument is that it is fully tailored to our neural network model. We are using spiking units network, and so, a binary spatio-temporal activity is generated by a sequence of inputs. Easily come to mind if it is possible to establishment a complexity index in artificial intelligence systems.

The methodology that will be used here is essentially composed of the same elements. The only different is in the perturbation, since, obviously, there is no TMS in our context.

¹ (x, t) is the position and the moment that TMS evokes significant cortical activity.

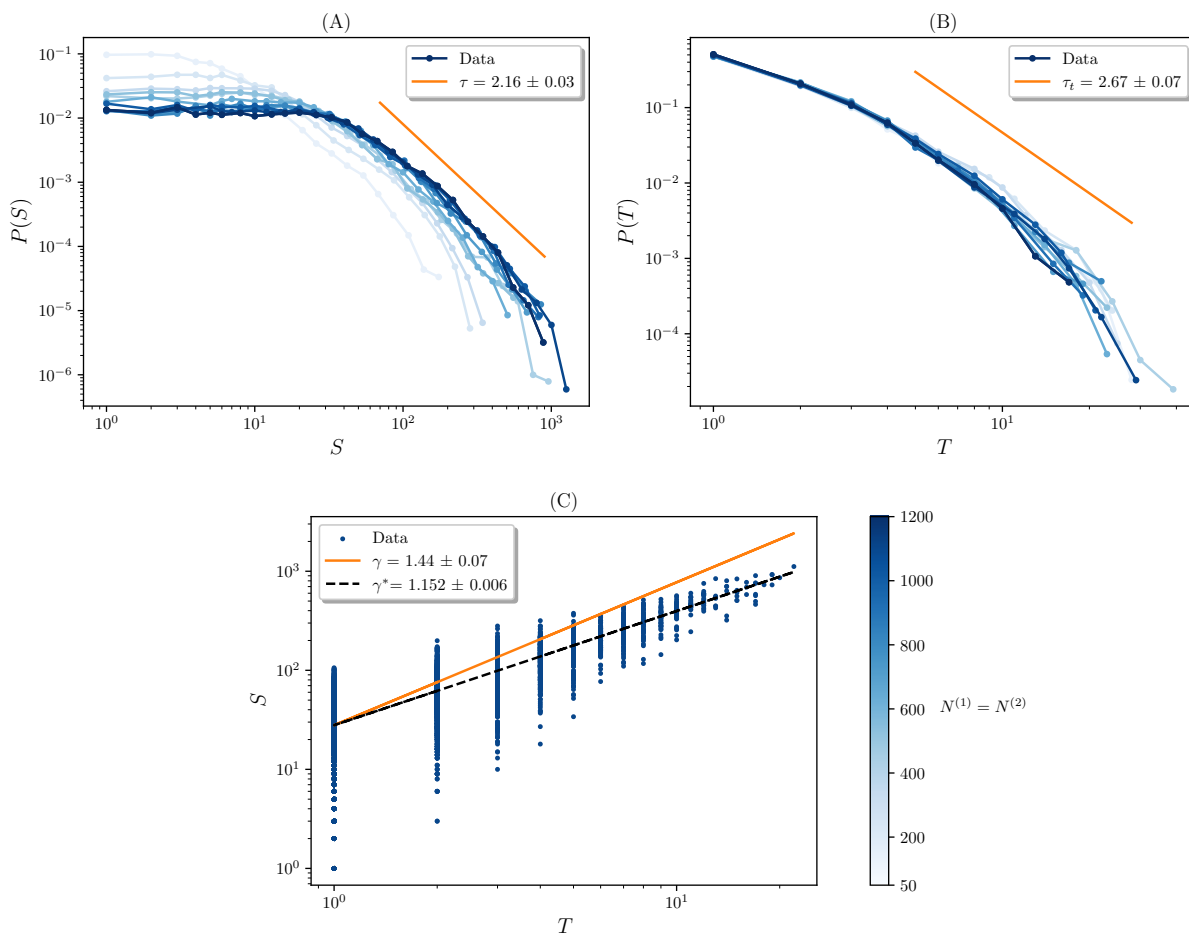


Figure 3.14: (A) Size and (B) duration avalanches. Power law fit using [66] are very sensitive to the choices of the minimum and maximum values.

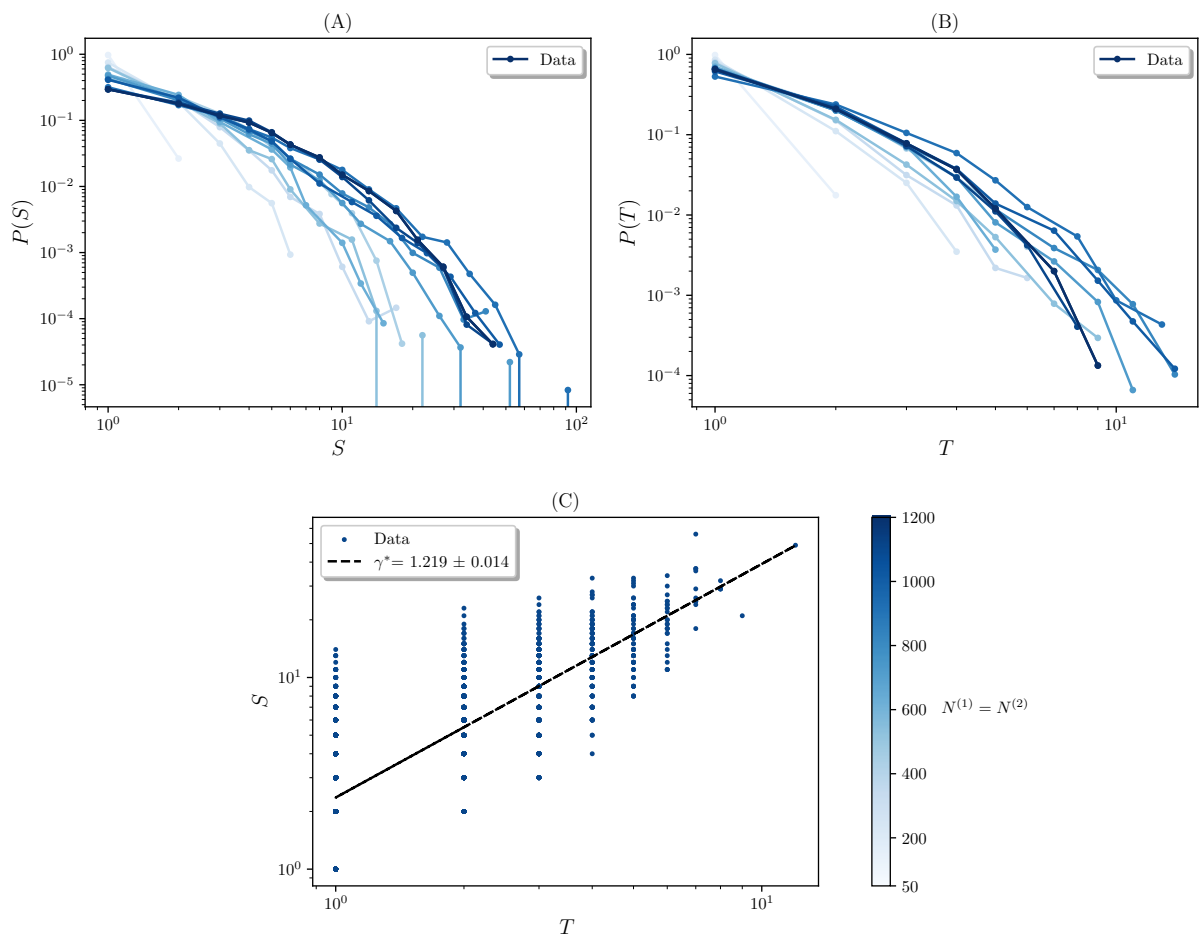
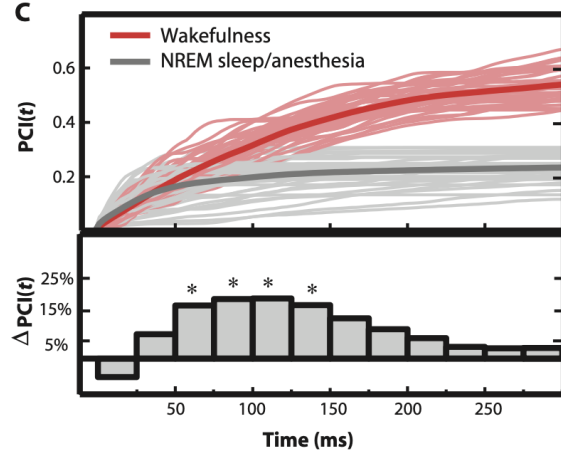


Figure 3.15: (A) Size and (B) duration avalanches. Here, the neural network was not trained, i.e, the accuracy is zero. As a consequence, power law fit was not fitted accurately.

Figure 3.16: The temporal evolution of PCI, $PCI(t)$, was constructed by calculating the cumulative time series of the normalized Lempel-Ziv complexity of SS. The rate of complexity divergence between the conscious and unconscious groups, $DPCI(t)$, was calculated from single-subject differences between the temporal evolution of PCI during wakefulness and loss of consciousness, with 25-ms time bins. Upper panel: Single-subject curves of $PCI(t)$ calculated during both wakefulness (light red lines) and loss of consciousness (light gray lines). The dark gray and red lines represent averaged $PCI(t)$. Lower panel: Percentages of $DPCI(t)$ generated in each temporal bin and the statistical significance (asterisks) with respect to the average value across bins (* $P = 0.002$, Mann-Whitney). Image and caption took by reference [52].



For that reason, the complexity index will be computed only in situations where external parameters change. Based on experiments investigated previously, such as learning process and effect of anesthesia, we aim to determine the meaning of complexity for each experimental condition.

3.6.1 Complexity of Learning Process

Complexity of learning process reeferes how significant the information processing of hidden layers is during the training. Following the same previous procedure, we will train again a classifier spiking neural network with the same parameters as the table 2.3. Consequently, an accuracy curve very similar to the Figure 2.9 (A) will be obtained, with three significant phases of learning. Thus, for each training time, a fast transition between classes movie M_F will be presented to neural network and then the PCI of hidden layers will be recorded.

As shown in Figure 3.17, we observe that for both hidden layers, PCI increases as the learning process approaches the saturated phase. However, at the beginning of training, the PCI for each layer exhibits distinct behaviors. In the early stages of training, during the first learning phase, the PCI of the first layer increases significantly, but decreases as it reaches the second learning phase. Conversely, for the second layer, the PCI starts from zero and increases continuously. As we move from the second training phase to the final training phase, PCI increases uniformly.

Before understanding the PCI's behavior, let's revisit Figure 2.9 (C). Throughout the learning process, the empirical distribution of synaptic weights in the first layer changes much more significantly than in the second layer. Initially following a uniform distribution, the histogram of the first layer gradually transitions towards a Gaussian-like distribution by the end of the last training phase. Consequently, the response of the first layer's activity, during the learning, to sequential inputs will not be equal to that of the second

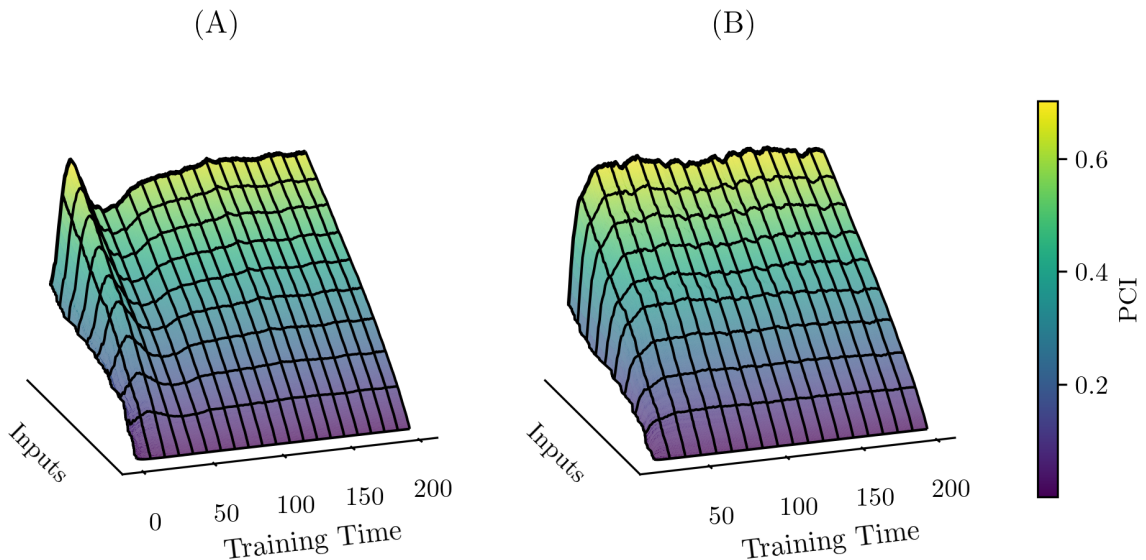


Figure 3.17: PCI in the hidden layers for each training time step over MNIST dataset. We use M_F movie as inputs. (A) First hidden layer. (B) Second hidden layer.

layer. Given that the second layer is closer to the output layer, its response tends to be more cohesive, exhibiting a smoother behavior. Bringing the results from the slow transition between classes section, we argue that the formation of the attraction basin is more relevant in the first layer than in the second one.

3.6.2 Complexity of Perception

We would like to address how information processing is affected by the anesthesia effect. For this experimental condition, the same methodology was employed as in the previous experiment. For each anesthesia dose effect, we presented an M_F type of movie, and then we recorded the PCI for each hidden layer. Figure 3.18 shows the results.

Upon first glance, it is apparent that the PCI evolution curve closely resembles the pattern observed in training time. Again, the second layer has a much more uniform behavior than the first one. Additionally, The PCI of the first layer begins to increase continuously just at the moment when the accuracy of neural network classification starts to rise ($\alpha \approx 0.35$), see Figure 3.18 (C). This phenomenon was also observed in the learning process experiment, as it occurs when the network moves to the second phase of training, during which the accuracy starts to increase from zero.

Besides the similarity in the PCI curve, there is also a resemblance in the accuracy curve, which takes on an S shape. It is quite intriguing, considering that in the anesthesia experiment, the synaptic weights between layers remain the same, whereas in the learning process, they do not. Nevertheless, the hidden layers' response to both regimes seems to be quite similar.

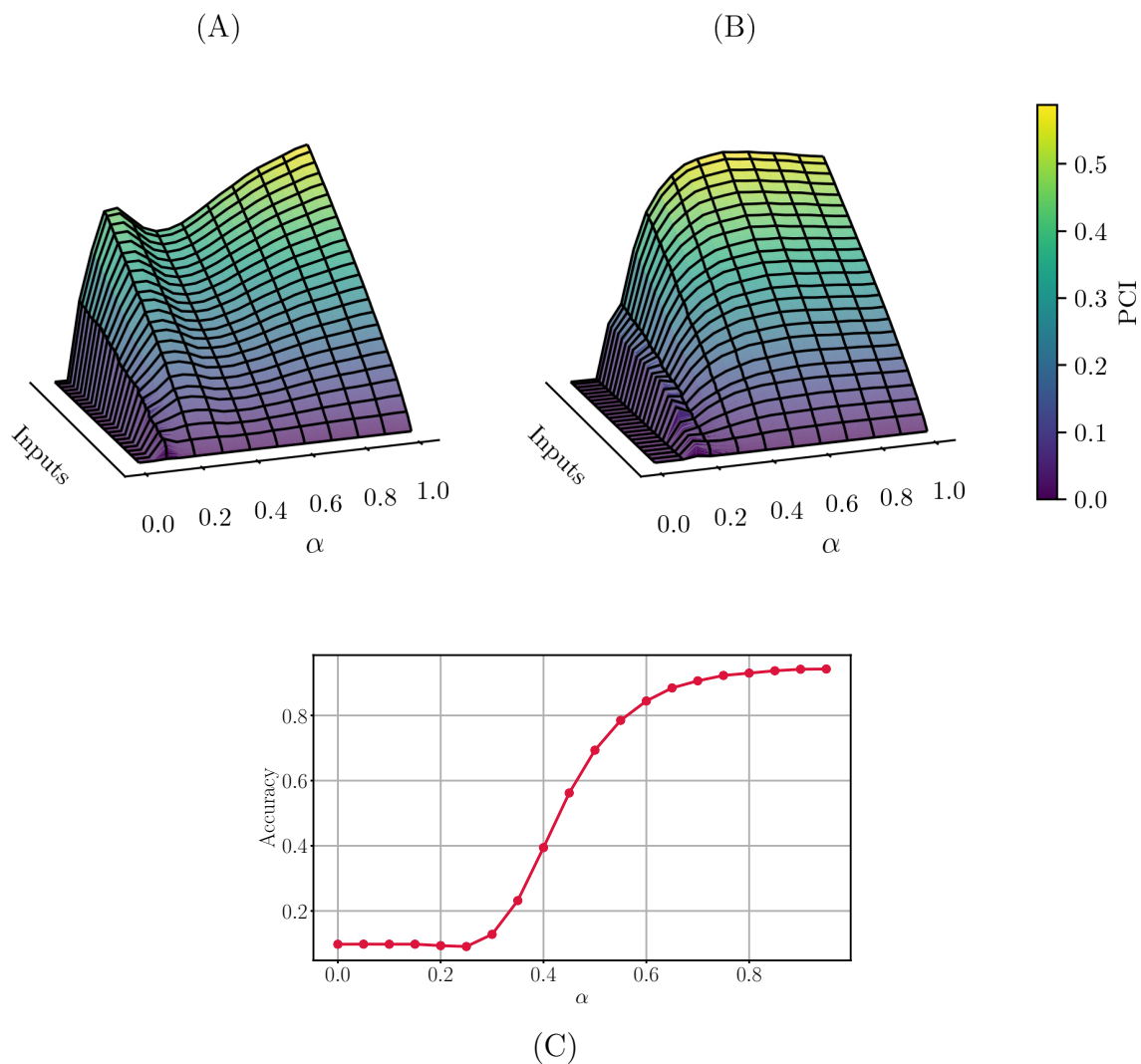


Figure 3.18: PCI in the hidden layers for each effect of anesthesia over MNIST dataset. We use M_F movie as inputs. (A) First hidden layer. (B) Second hidden layer. (C) Accuracy (output layer) in function of anesthesia effect.

Chapter 4

Conclusion

‘You cannot make a machine to think for you.’ This is a commonplace that is usually accepted without question. It will be the purpose of this paper to question it.

ALAN TURING, Intelligent Machinery, A Heretical Theory (1951)

In this study we trained a spiking neural network with deep architecture as a classifier on standard databases and subjected it to different experimental conditions, recording time series of the activity of the hidden layers. We show that persistent correlations in the internal representations (IR) activity appears associated to the transition when inputs of different labels, belonging to the dataset, are presented to the neural network. This is a signature of broad tail distribution of the temporal correlation of fluctuations of internal activity. We interpret this phenomenon in the light of an attractor basin dynamics. The neural network learns the concept associated of an image label, and we can investigate the dynamics within its internal layers. When inputs from distinct categories prompt the system to converge towards new attractors, generating long-range correlations, the output layer response follows this dynamics by correctly classifying the new presented image. This is demonstrated when we present random inputs. In this scenario, transitions of images of distinct distributions do not produce long tails distribution in IR, as there is no search for any attractor basin. We illustrate that transitioning to a new attractor may differ depending on the label, as the dataset classes are scattered in a high-dimensional space following a specific structure, which is reflected in the peak size of the correlation time.

The aim of this study was also to investigate how the learning process occurs in the hidden layers, taking into account the three training phases. We demonstrate that the formation of attraction basins occurs in the regime where recognition accuracy is still zero. This leads us to argue that the first training phase is crucial element. In the last phase, where the accuracy plateau is reached, the attraction basins are already formed. Thus, we can conclude that despite no modification occurring in the output layer response, significant information processing is already taking place in the early layers.

We then address whether the large increase of ξ_τ is a criticality signature. We have defined avalanches and tried to fit power laws to the empirical distributions of size and duration in function of the network size, for the partially and fully trained neural network as well as for the anesthetized model. In addition, the exponents of the power law distributions of the coupling between the size and duration of avalanches, γ and γ^* , were also estimated. Since the exponents are not equal, taking into account the fitting error,

crackling noise scaling relations are not satisfied, a strong suggestion that there is no criticality. In light of these findings, it is evident that the search for a new attractor leads to large fluctuations without criticality.

We extended this study in the article entitled “*Internal Representations in Spiking Neural Networks, criticality, and the Renormalization Group*”, attached in the appendix of this dissertation, where we provide examples of non-critical physical systems that exhibit long-tailed distributions. This includes the dynamics of a particle undergoing Brownian motion in a slowly varying time-dependent potential, with two attraction basins which periodically alternate in strength. There is an increase in the characteristic correlation time of the particle’s position as its trajectory moves from one minimum of the potential to another. Additionally, magnetic systems, such as the Ising model on an infinite lattice subjected to an external field, are considered. Outside the critical temperature, as the external field changes slowly, the expected value of magnetization shifts between two ordered phases, and consequently, fluctuations become correlated on a timescale larger than the change in the external field. Finally, all of this is interpreted from the standpoint of renormalization group theory. Simulations using the Monte Carlo Renormalization Group (MCRG) method on a 2D Ising model with nearest-neighbor interactions subjected to a slowly varying uniform external field showed that susceptibility peaks, for conditions outside or at the critical point, are more pronounced in the renormalized system. This has some implications. First, the role of the external field in magnetic systems resembles input patterns in neural networks. In other words, as we change the label of the image, correlation peaks emerge without signs of criticality. Second, in a neural network with feedforward architecture, the multiple internal layers serve to filter the microscopic configurations of the images and represent them in macroscopic states in order to perform classification correctly.

Also, the perturbational complexity index (PCI) was measured in the hidden layers when the classifier is processing several fast transition of inputs between classes of dataset in scenarios such as the training section and under the effect of anesthesia. It was expected that the complexity of the system would increase as a significant task is tackled, implying that “complexity meets learning”. However, we found even more intriguing results for understanding the role of hidden layers in representing information from the outside world. For both experimental conditions, the behavior of the PCI is similar. The complexity of activity of second layer tends to have a smoother behavior, while the first layer is less cohesive when the model is in the early stages of learning or when the effect of anesthesia is high. This indicates that the representation of images in the first hidden layer is more susceptible to changes in learning situations or conditions where incoming information is reasonably different from what has been learned, since this layer is closer to the input layer. As we go deeper into the network, the representation becomes simpler, as the output layer represents an image of many pixels (degrees of freedom) in just one label. This is directly connected to the evolution of the empirical distribution of synaptic weights between layers during the training, where in the first layer the change is greater than in the second. This means that as we add more layers, fewer modifications of the connections between neurons closer to the output layer will be observed. In short, these experiments say that PCI can distinguish spatio-temporal patterns of activity across different information processing regimes, which it is an important methodology to quantify complexity inside artificial neural networks.

Finally, feedforward neural networks are certainly not models of the brain, but, certainly the cortex of freely behaving and even anesthetized animals, are constantly stimu-

lated by changing inputs, which have to be processed by extracting relevant information to drive behavioral decisions. This can induce changes in the basin of attraction of the different concepts being identified and this search for the new attractor, generate persistent fluctuations of neuronal activity. A concrete example is the result in [69], where the “stimulus onset quenches neural variability”, which they claim to be a rather general property of the cortex.

References

- [1] Warren S. McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: <https://doi.org/10.1007/BF02478259> (cit. on p. 7).
- [2] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, 2002. ISBN: 0805843000 (cit. on p. 7).
- [3] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386. DOI: <https://doi.org/10.1037/h0042519> (cit. on p. 7).
- [4] M. Minsky and S. Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 1969. DOI: 10.7551/mitpress/11301.001.0001 (cit. on p. 8).
- [5] Paul Werbos. “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University”. PhD thesis. Jan. 1974 (cit. on p. 8).
- [6] David E. Rumelhart, James L. McClelland, and the PDP Research Group. *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. MIT press, 1986. DOI: <https://doi.org/10.7551/mitpress/5236.001.0001> (cit. on pp. 8, 9).
- [7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. on p. 8).
- [8] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (cit. on p. 8).
- [9] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961. URL: <https://doi.org/10.1038/nature16961> (cit. on p. 8).
- [10] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: <https://doi.org/10.1038/s41586-021-03819-2> (cit. on p. 8).

- [11] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001. DOI: [10.1017/CB09781139164542](https://doi.org/10.1017/CB09781139164542) (cit. on p. 9).
- [12] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, NY, 2000. ISBN: 978-0-387-98780-4. DOI: <https://doi.org/10.1007/978-1-4757-3264-1> (cit. on p. 9).
- [13] Peter L. Bartlett and Shahar Mendelson. “Rademacher and Gaussian Complexities: Risk Bounds and Structural Results”. In: *Computational Learning Theory*. Ed. by David Helmbold and Bob Williamson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 224–240. ISBN: 978-3-540-44581-4 (cit. on p. 9).
- [14] Teuvo Kohonen. “An introduction to neural computing”. In: *Neural Networks 1.1* (1988), pp. 3–16. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(88\)90020-2](https://doi.org/10.1016/0893-6080(88)90020-2). URL: <https://www.sciencedirect.com/science/article/pii/0893608088900202> (cit. on p. 9).
- [15] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals and Systems 2.4* (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: [10.1007/BF02551274](https://doi.org/10.1007/BF02551274). URL: <https://doi.org/10.1007/BF02551274> (cit. on p. 9).
- [16] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks 2.5* (1989), pp. 359–366. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8). URL: <https://www.sciencedirect.com/science/article/pii/0893608089900208> (cit. on p. 9).
- [17] Avrim L. Blum and Ronald L. Rivest. “Training a 3-node neural network is NP-complete”. In: *Neural Networks 5.1* (1992), pp. 117–127. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(05\)80010-3](https://doi.org/10.1016/S0893-6080(05)80010-3). URL: <https://www.sciencedirect.com/science/article/pii/S0893608005800103> (cit. on p. 9).
- [18] L. Breiman. “Reflections After Refereeing Papers for NIPS. In The Mathematics of Generalization: Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning.” In: 1992 (cit. on p. 9).
- [19] Stuart Geman, Elie Bienenstock, and René Doursat. “Neural Networks and the Bias/Variance Dilemma”. In: *Neural Computation 4.1* (Jan. 1992), pp. 1–58. ISSN: 0899-7667. DOI: [10.1162/neco.1992.4.1.1](https://doi.org/10.1162/neco.1992.4.1.1). eprint: <https://direct.mit.edu/neco/article-pdf/4/1/1/812244/neco.1992.4.1.1.pdf>. URL: <https://doi.org/10.1162/neco.1992.4.1.1> (cit. on p. 10).
- [20] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences 116.32* (2019), pp. 15849–15854. DOI: [10.1073/pnas.1903070116](https://doi.org/10.1073/pnas.1903070116). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1903070116>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1903070116> (cit. on p. 10).
- [21] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. In: *CoRR abs/1611.03530* (2016). arXiv: [1611.03530](https://arxiv.org/abs/1611.03530). URL: <http://arxiv.org/abs/1611.03530> (cit. on p. 11).

- [22] Anna Choromanska et al. “The Loss Surfaces of Multilayer Networks”. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Guy Lebanon and S. V. N. Vishwanathan. Vol. 38. Proceedings of Machine Learning Research. San Diego, California, USA: PMLR, Sept. 2015, pp. 192–204. URL: <https://proceedings.mlr.press/v38/choromanska15.html> (cit. on p. 11).
- [23] Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. “Bad Global Minima Exist and SGD Can Reach Them”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 8543–8552. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/618491e20a9b686b79e158c293ab4f91-Paper.pdf (cit. on p. 11).
- [24] Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, July 2001. ISBN: 9780198509417. DOI: 10.1093/acprof:oso/9780198509417.001.0001. URL: <https://doi.org/10.1093/acprof:oso/9780198509417.001.0001> (cit. on p. 11).
- [25] J J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554> (cit. on p. 11).
- [26] Lenka Zdeborová. “Understanding deep learning is also a job for physicists”. In: *Nature Physics* 16.6 (June 2020), pp. 602–604. ISSN: 1745-2481. DOI: 10.1038/s41567-020-0929-2. URL: <https://doi.org/10.1038/s41567-020-0929-2> (cit. on p. 11).
- [27] Cédric Bény. *Deep learning and the renormalization group*. 2013. arXiv: 1301.3124 [quant-ph] (cit. on p. 11).
- [28] Nestor Caticha. “Entropic Dynamics in Neural Networks, the Renormalization Group and the Hamilton-Jacobi-Bellman Equation”. In: *Entropy* 22.5 (2020). ISSN: 1099-4300. DOI: 10.3390/e22050587. URL: <https://www.mdpi.com/1099-4300/22/5/587> (cit. on p. 11).
- [29] Satoshi Iso, Shotaro Shiba, and Sumito Yokoo. “Scale-invariant feature extraction of neural network and renormalization group flow”. In: *Phys. Rev. E* 97 (5 May 2018), p. 053304. DOI: 10.1103/PhysRevE.97.053304. URL: <https://link.aps.org/doi/10.1103/PhysRevE.97.053304> (cit. on p. 11).
- [30] Juan Carrasquilla and Roger G. Melko. “Machine learning phases of matter”. In: *Nature Physics* 13.5 (May 2017), pp. 431–434. ISSN: 1745-2481. DOI: 10.1038/nphys4035. URL: <https://doi.org/10.1038/nphys4035> (cit. on p. 11).
- [31] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999118307125> (cit. on p. 11).
- [32] Ravid Shwartz-Ziv and Naftali Tishby. “Opening the black box of deep neural networks via information”. In: *arXiv preprint arXiv:1703.00810* (2017) (cit. on p. 11).

- [33] John M. Beggs and Dietmar Plenz. “Neuronal Avalanches in Neocortical Circuits”. In: *Journal of Neuroscience* 23.35 (2003), pp. 11167–11177. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.23-35-11167.2003. eprint: <https://www.jneurosci.org/content/23/35/11167.full.pdf>. URL: <https://www.jneurosci.org/content/23/35/11167> (cit. on p. 13).
- [34] Heinz Georg Schuster. *Criticality in neural systems*. John Wiley & Sons, 2014. ISBN: 9783527651009. DOI: 10.1002/9783527651009 (cit. on p. 14).
- [35] Theodore Edward Harris. *The Theory of Branching Processes*. Springer Berlin, Heidelberg, 1963. ISBN: 978-3-642-51868-3 (cit. on p. 14).
- [36] Osame Kinouchi and Mauro Copelli. “Optimal dynamical range of excitable networks at criticality”. In: *Nature Physics* 2.5 (May 2006), pp. 348–351. ISSN: 1745-2481. DOI: 10.1038/nphys289. URL: <https://doi.org/10.1038/nphys289> (cit. on p. 14).
- [37] Jonathan Touboul and Alain Destexhe. “Power-law statistics and universal scaling in the absence of criticality”. In: *Phys. Rev. E* 95 (1 Jan. 2017), p. 012413. DOI: 10.1103/PhysRevE.95.012413. URL: <https://link.aps.org/doi/10.1103/PhysRevE.95.012413> (cit. on p. 14).
- [38] James P. Sethna, Karin A. Dahmen, and Christopher R. Myers. “Crackling noise”. In: *Nature* 410.6825 (Mar. 2001), pp. 242–250. ISSN: 1476-4687. DOI: 10.1038/35065675. URL: <https://doi.org/10.1038/35065675> (cit. on pp. 14, 55).
- [39] Per Bak, Chao Tang, and Kurt Wiesenfeld. “Self-organized criticality: An explanation of the $1/f$ noise”. In: *Phys. Rev. Lett.* 59 (4 July 1987), pp. 381–384. DOI: 10.1103/PhysRevLett.59.381. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.59.381> (cit. on p. 14).
- [40] Juan A Bonachela and Miguel A Muñoz. “Self-organization without conservation: true or just apparent scale-invariance?” In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.09 (Sept. 2009), P09009. DOI: 10.1088/1742-5468/2009/09/P09009. URL: <https://dx.doi.org/10.1088/1742-5468/2009/09/P09009> (cit. on p. 14).
- [41] Mauricio Girardi-Schappo. “Brain criticality beyond avalanches: open problems and how to approach them”. In: *Journal of Physics: Complexity* 2.3 (Sept. 2021), p. 031003. DOI: 10.1088/2632-072X/ac2071. URL: <https://dx.doi.org/10.1088/2632-072X/ac2071> (cit. on pp. 14, 15).
- [42] Osame Kinouchi, Renata Pazzini, and Mauro Copelli. “Mechanisms of Self-Organized Quasicriticality in Neuronal Network Models”. In: *Frontiers in Physics* 8 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.583213. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2020.583213> (cit. on p. 14).
- [43] Viola Priesemann, Matthias HJ Munk, and Michael Wibral. “Subsampling effects in neuronal avalanche distributions recorded in vivo”. In: *BMC Neuroscience* 10.1 (Apr. 2009), p. 40. ISSN: 1471-2202. DOI: 10.1186/1471-2202-10-40. URL: <https://doi.org/10.1186/1471-2202-10-40> (cit. on p. 15).

- [44] J Wilting and V Priesemann. “25 years of criticality in neuroscience — established results, open controversies, novel concepts”. In: *Current Opinion in Neurobiology* 58 (2019). Computational Neuroscience, pp. 105–111. ISSN: 0959-4388. DOI: <https://doi.org/10.1016/j.conb.2019.08.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0959438819300248> (cit. on p. 15).
- [45] Miguel A. Muñoz. “Colloquium: Criticality and dynamical scaling in living systems”. In: *Rev. Mod. Phys.* 90 (3 July 2018), p. 031001. DOI: 10.1103/RevModPhys.90.031001. URL: <https://link.aps.org/doi/10.1103/RevModPhys.90.031001> (cit. on p. 15).
- [46] Tawan T. A. Carvalho et al. “Subsampled Directed-Percolation Models Explain Scaling Relations Experimentally Observed in the Brain”. In: *Frontiers in Neural Circuits* 14 (2021). ISSN: 1662-5110. DOI: 10.3389/fncir.2020.576727. URL: <https://www.frontiersin.org/articles/10.3389/fncir.2020.576727> (cit. on p. 15).
- [47] Danielle S Bassett and Michael S Gazzaniga. “Understanding complexity in the human brain”. In: *Trends in Cognitive Sciences* 15.5 (2011), pp. 200–209. DOI: <https://doi.org/10.1016/j.tics.2011.03.006> (cit. on p. 16).
- [48] Giulio Tononi. “An information integration theory of consciousness”. In: *BMC Neuroscience* 5.1 (Nov. 2004), p. 42. ISSN: 1471-2202. DOI: 10.1186/1471-2202-5-42. URL: <https://doi.org/10.1186/1471-2202-5-42> (cit. on p. 16).
- [49] Masafumi Oizumi, Naotsugu Tsuchiya, and Shun-ichi Amari. “Unified framework for information integration based on information geometry”. In: *Proceedings of the National Academy of Sciences* 113.51 (2016), pp. 14817–14822. DOI: 10.1073/pnas.1603583113. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1603583113>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1603583113> (cit. on p. 16).
- [50] Shun-ichi Amari. *Information Geometry and Its Applications*. Springer Tokyo, 2016. DOI: <https://doi.org/10.1007/978-4-431-55978-8> (cit. on p. 16).
- [51] Otavio Citton and Nestor Caticha. *Integrated Information, a Complexity Measure for optimal partitions*. 2023. arXiv: 2304.14316 [cond-mat.stat-mech] (cit. on p. 16).
- [52] Adenauer G. Casali et al. “A Theoretically Based Index of Consciousness Independent of Sensory Processing and Behavior”. In: *Science Translational Medicine* 5.198 (2013), 198ra105–198ra105. DOI: 10.1126/scitranslmed.3006294. eprint: <https://www.science.org/doi/pdf/10.1126/scitranslmed.3006294>. URL: <https://www.science.org/doi/abs/10.1126/scitranslmed.3006294> (cit. on pp. 16, 58).
- [53] Jason K. Eshraghian et al. “Training Spiking Neural Networks Using Lessons From Deep Learning”. In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054. DOI: 10.1109/JPROC.2023.3308088 (cit. on pp. 19, 26, 32).
- [54] Paul A. Merolla et al. “A million spiking-neuron integrated circuit with a scalable communication network and interface”. In: *Science* 345.6197 (2014), pp. 668–673. DOI: 10.1126/science.1254642. eprint: <https://www.science.org/doi/pdf/10.1126/science.1254642>. URL: <https://www.science.org/doi/abs/10.1126/science.1254642> (cit. on p. 19).

- [55] Léon Bottou. “Large-Scale Machine Learning with Stochastic Gradient Descent”. In: *Proceedings of COMPSTAT’2010*. Ed. by Yves Lechevallier and Gilbert Saporta. Heidelberg: Physica-Verlag HD, 2010, pp. 177–186. ISBN: 978-3-7908-2604-3 (cit. on p. 22).
- [56] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: <https://doi.org/10.1038/323533a0> (cit. on p. 23).
- [57] Peter Dayan and Laurence F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2001. ISBN: 9780262041997. URL: <https://mitpress.mit.edu/9780262041997/> (cit. on pp. 24, 28).
- [58] A. L. HODGKIN and A. F. HUXLEY. “Action Potentials Recorded from Inside a Nerve Fibre”. In: *Nature* 144.3651 (Oct. 1939), pp. 710–711. ISSN: 1476-4687. DOI: 10.1038/144710a0. URL: <https://doi.org/10.1038/144710a0> (cit. on p. 24).
- [59] Richard Ernest Bellman. *Dynamic Programming*. USA: Dover Publications, Inc., 2003. ISBN: 0486428095 (cit. on p. 24).
- [60] Paul J. Werbos. “Generalization of backpropagation with application to a recurrent gas market model”. In: *Neural Networks* 1.4 (1988), pp. 339–356. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(88\)90007-X](https://doi.org/10.1016/0893-6080(88)90007-X). URL: <https://www.sciencedirect.com/science/article/pii/089360808890007X> (cit. on p. 30).
- [61] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1310–1318. URL: <https://proceedings.mlr.press/v28/pascanu13.html> (cit. on p. 31).
- [62] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63. DOI: 10.1109/MSP.2019.2931595 (cit. on p. 31).
- [63] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: <http://jmlr.org/papers/v9/vandermaaten08a.html> (cit. on pp. 41, 46).
- [64] Hugues Van Assel et al. *Distributional Reduction: Unifying Dimensionality Reduction and Clustering with Gromov-Wasserstein Projection*. 2024. arXiv: 2402.02239 [cs.LG] (cit. on pp. 41, 46).
- [65] Simon-Shlomo Poil et al. “Critical-State Dynamics of Avalanches and Oscillations Jointly Emerge from Balanced Excitation/Inhibition in Neuronal Networks”. In: *Journal of Neuroscience* 32.29 (2012), pp. 9817–9823. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5990-11.2012. eprint: <https://www.jneurosci.org/content/32/29/9817.full.pdf>. URL: <https://www.jneurosci.org/content/32/29/9817> (cit. on p. 54).

- [66] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. “powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions”. In: *PLOS ONE* 9.1 (Jan. 2014), pp. 1–11. DOI: 10.1371/journal.pone.0085777. URL: <https://doi.org/10.1371/journal.pone.0085777> (cit. on pp. 55, 56).
- [67] Leonardo Dalla Porta and Mauro Copelli. “Modeling neuronal avalanches and long-range temporal correlations at the emergence of collective oscillations: Continuously varying exponents mimic M/EEG results”. In: *PLOS Computational Biology* 15.4 (Apr. 2019), pp. 1–26. DOI: 10.1371/journal.pcbi.1006924. URL: <https://doi.org/10.1371/journal.pcbi.1006924> (cit. on p. 55).
- [68] Giulio Tononi and Gerald M Edelman. “Consciousness and complexity”. In: *science* 282.5395 (1998), pp. 1846–1851. DOI: 10.1126/science.282.5395.1846. eprint: <https://www.science.org/doi/abs/10.1126/science.282.5395.1846>. URL: <https://www.science.org/doi/abs/10.1126/science.282.5395.1846> (cit. on p. 55).
- [69] Mark M. Churchland et al. “Stimulus onset quenches neural variability: a widespread cortical phenomenon”. In: *Nature Neuroscience* 13.3 (Mar. 2010), pp. 369–378. ISSN: 1546-1726. DOI: 10.1038/nn.2501. URL: <https://doi.org/10.1038/nn.2501> (cit. on p. 63).

Appendices

Appendix A

Article

Internal Representations in Spiking Neural Networks, criticality and the Renormalization Group

João Henrique de Sant'Ana* Nestor Caticha†

Instituto de Física, Universidade de São Paulo
São Paulo, Brazil

May 8, 2024

Abstract

Optimal information processing in peripheral sensory systems has been associated in several examples to the signature of a critical or near critical state. Furthermore, cortical systems have also been described to be in a critical state in both wake and anesthetized experimental models, both *in vitro* and *in vivo*. We investigate whether a similar signature characterizes the internal representations (IR) of a multilayer (deep) spiking artificial neural network performing computationally simple but meaningful cognitive tasks, using a methodology inspired in the biological setup, with cortical implanted electrodes in rats, either freely behaving or under different levels of anesthesia. The increase of the characteristic time of the decay of the correlation of fluctuations of the IR, found when the network input changes, are indications of a broad-tailed distribution of IR fluctuations. The broad tails are present even when the network is not yet capable of performing the classification tasks, either due to partial training or to the effect of a low dose of anesthesia in a simple model. However, we don't find enough evidence of power law distributions of avalanche size and duration. We interpret the results from a renormalization group perspective to point out that despite having broad tails, this is not related to a critical transition but rather similar to fluctuations driven by the reversal of the magnetic field in a ferromagnetic system. Another example of persistent correlation of fluctuations of a non critical system is constructed, where a particle undergoes Brownian motion on a slowly varying potential.

Keywords: Neural Networks, Internal representations, Statistical Mechanics, Renormalization, Critical Brain.

1 Introduction

The idea that the brain operates at a critical point of a phase transition has been present in the field of Neuroscience since the first experimental description of neuronal

*joao.henrique.santana@usp.br

†ncaticha@usp.br

avalanches, with its size and duration scale-free bursts of neuronal activity, by Beggs and Plenz [1]. Further extensive activity has discussed and given theoretical support [2, 3, 4, 5, 6, 7, 8] to the *critical brain hypothesis*. A possible evolutionary reason for criticality is that it has been shown to be related to optimal information processing, in the sense of maximum dynamic range in neural networks of excitable units that model sensory systems, as shown by Kinouchi and Copelli [2], where criticality is associated to the network topology, i.e. bond percolation of the couplings. Self organized criticality can occur due to learning in a recurrent neural network, driven by random inputs [8]. However, experimental or numerical signatures such as power laws and scaling are possible without criticality, with examples [9, 10, 11] in general areas of biology or in particular in neural networks models. It is interesting to further investigate putative critical behavior in systems with slowly decaying correlations and broad tail distribution.

In this paper we investigate the statistics of fluctuations in the internal layers of a feed-forward Neural Network of spiking neurons when the network is performing a “meaningful cognitive” task of pattern recognition. This is done (see details in section 2) to mimic the data collection in an experiment of cortical implanted electrodes in rats exposed to a free environment illustrated in Figure 1. We study the time truncated-correlation of fluctuations of the average spike activity, of trains of spikes and average membrane potentials of the internal representations in the network’s hidden layers and measure their characteristic decay times, generically denoted ξ . Large values, as are typically associated to criticality, can be seen when the time dependent input changes and the classifier network changes the classification. Networks exposed to a sequence of different images in the same category do not present such “diverging” (much longer) ξ ’s. We also investigate size and duration of neuronal activity avalanches. These broad-tailed distributions are apparently similar to those which have been summoned to justify the term critical, are only found when the network classifies a changing environment. However, despite the allure of calling this a (near-) critical state, we are doubtful about this interpretation. In section 4 we show a simple dynamical model with broad tailed distribution of fluctuations of its state, associated to the shifts of the basin of attractions, similar to the dynamics the network undergoes, when the category of the input changes. This adds to the models that shows that long tailed distributions can occur without critical behavior. This very low dimensional system serves only as a simple metaphor for the appearance of broad tails, so we search for systems with non critical collective behavior with broad tails and their description using Statistical Mechanics and the Renormalization Group.

The methods of Statistical Mechanics have been extensively applied [12, 13, 14, 15, 16] to study computational properties of artificial neural networks. This allows the description of brain inspired dynamical systems in the language of emergent collective phenomena. The Statistical Mechanics description of emergent properties of the thermodynamics state is analogous to concept formation [17, 12, 13]. The methods of Statistical Mechanics have been extensively applied [12, 13, 14, 15, 16] to study computational properties of artificial neural networks. This allows the description of brain inspired dynamical systems in the language of emergent collective phenomena. The Statistical Mechanics description of emergent properties of the thermodynamics state is analogous to concept formation [17, 12, 13].

The Renormalization Group (RG) is the main theoretical tool to analyze systematically the multiple scales, spatial and/or temporal, relevant in the critical region of a second order phase transition. In general the RG is a map from measures to measures or

between Hamiltonians. Here we adopt the simplifying description as a map between configurations, which despite being limited in the scope of what the RG is, avoids discussing peculiarities of the renormalized measures or Hamiltonians, which may only occur, if at all, in the thermodynamic limit. As no single scale is dominant, power laws are associated to the critical state. Since there are simpler theoretical methods useful in the noncritical regions of parameter space, there is a tendency to associate the RG to the study of the critical region, despite being useful elsewhere. Globally we can look at the RG as a classifier which maps microstates into order parameters that characterize thermodynamic states in the whole range of parameters.

The evolution of probability distributions under exact RG transformations [18, 19] are an example of entropic dynamics [20]. The RG works by systematically marginalizing the degrees of freedom of a joint distribution of the degrees of freedom in all scales. The distribution of degrees of freedom at the coarser scale results from a MaxEnt implementation of constraints imposed by the coarsening procedure. This amounts to implementing a filtering process, where the filters are chosen *a priori*. A feed-forward neural network, with either deep or shallow architecture, also implements a filtering process [21, 22, 23, 24, 25], with the difference that the filters are designed automatically during the learning process and not imposed by prior knowledge. Of course, for the RG, prior knowledge about the adequate filters came from the seminal work of Kadanoff [26], Wilson [27] and others. This identification is specially clear in the Monte Carlo version of real space RG (MCRG) [28, 29] which we discuss in 4.2 as an example of a convolutional neural network. An application of the MCRG to a 2d Ising model with a slowly changing external field, shows that large susceptibility, and the broad tail distribution of fluctuations it entails, can occur despite being clearly away from the critical region. In the case where in addition to the external field, there is a random field contribution at each site (normal, $\mu = 0, \sigma = R$) and $T = 0$ [30, 31], the field reversal dynamics is only truly critical with power laws of the Barkhausen noise at a critical value R . The transition is due to the reversal of the external magnetic field, which plays the role of the input pattern to the neural network. In our study, broad-tailed distributions are not associated to a critical transition, but rather to the shifting nature of basins of attraction that accompanies changes in the images presented to the network.

2 Materials and Methods

2.1 Experimental models: the neural networks

We constructed a classifier using a *deep* architecture, i.e a multilayered feed-forward fully connected, from layer to layer, neural network of spiking neurons shown in Figure 1.b. It was subject to supervised learning with the goal of recognizing images from standard datasets, the MNIST and the Fashion-MNIST. Both tasks had the same structure with respect to data set size, training and testing splits, composed of 60.000 training and 10.000 testing 28x28 gray scale images of 0 to 9 handwritten digits and of 10 categories of fashion products, respectively.

We use a simple and general purpose neuron model for large-scale training of artificial neural networks. The leaky integrate-and-fire (LIF) neuron, that goes back to Lapique [32] (see [33, 34]), captures the essential behavior of the electric potential difference in nerve cells, it is computational cheap and adequate for our purposes. The LIF follows

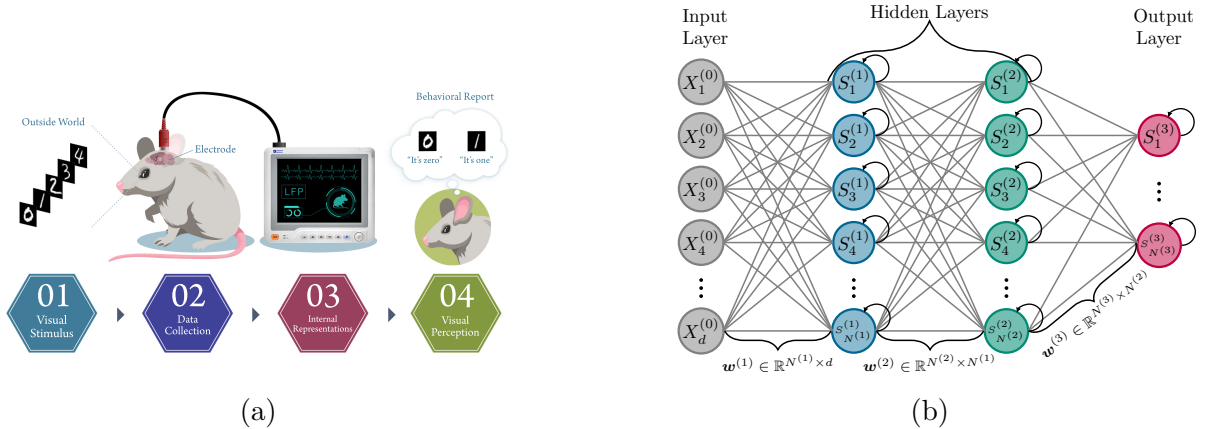


Figure 1: (a) Typical experimental setup, using electrodes for studying cortical activity during a visual perception task. (b) This experiment is simulated with an artificial neural network setup. The diagram shows the fully connected feed forward spiking neural network used in this paper. Each unit is a LIF spiking neuron and the loop arrow symbolizes the time-dependent changes of the membrane potential. The “electrodes” collect information from the hidden layers.

the principle of integration of inputs combined with a reset mechanism. The dynamics of each i^{th} LIF neuron in the l^{th} layer, with a reset by subtraction of threshold θ , is written as a discrete time step equation

$$\begin{cases} S_i^{(l)}[t] = \Theta(U_i^{(l)}[t] - \theta) = \begin{cases} 1 & \text{if } U_i^{(l)}[t] > \theta \\ 0 & \text{otherwise,} \end{cases} \\ U_i^{(l)}[t+1] = \beta U_i^{(l)}[t] + \sum_{j=1}^{N^{(l-1)}} w_{ij}^{(l)} S_j^{(l-1)}[t+1] - \theta S_i^{(l)}[t], \end{cases} \quad (1)$$

where $S_j^{(l-1)}$ is the presynaptic input of the j^{th} neuron in the previous layer ($l-1$), $w_{ij}^{(l)}$ is the adjustable synaptic coupling between the j^{th} neuron in the previous layer ($l-1$) and the i^{th} neuron in layer (l) and β controls the membrane potential decay.

To determine an appropriate learning algorithm for this class of neural network, neuronal communication must be precisely defined. Rate coding relates to the transmission of meaningful information only by means of neuron firing rates, i.e, the response of spike train frequency increases as the sum of weighted input increases. We implemented rate coding, which can be simply adapted to our model. The output layer has $n = 10$ neurons, so that in the classification problem each individual response of output neurons is connected to the “recognition” of the category of the input image: ‘It is a zero’ or ‘It is a shoe’. This is the behavioral report, a visual perception of a labeled input. For training, we used the error backpropagation through time (BPTT) [35] with the surrogate gradient approximation [36, 37]. $S = \Theta(U(t) - \theta)$, as a function of U is a step function, and as a function of t is a spike, firing when the potential becomes larger than the threshold and coming back to zero after the subsequent reset of the potential. This function is substituted by a differentiable function \tilde{S} , to be used only in the backward computation

Hyperparameters	
Neural Network	Neuron
$d = 28 \times 28$	1st Order Leaky Integrate and Fire Neuron
$N^{(1)} = 300, N^{(2)} = 300, N^{(3)} = 10$	$\beta = 0.5$
Batch size = 128	$\theta = 1$
Epochs = 1	$t_U = 25$
$\eta = 5 \times 10^{-4}$	
Optimizer: Adam	Reset Mechanism: Subtract
Accuracy metric: Spike Count	Backward pass: Fast Sigmoid
Loss: Cross Entropy Spike Count	

Table 1: Hyperparameters used on training. We used standard training for networks with $N^{(1)}, N^{(2)}$ neurons in the 2 hidden layers.

step. A component of the gradient of the loss function \mathcal{L} is:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \sum_{t=0}^{t_U} \frac{\partial \mathcal{L}(t)}{\partial S_j^{(l)}(t)} \frac{\partial \tilde{S}_j^{(l)}(t)}{\partial U_j^{(l)}(t)} \frac{\partial U_j^{(l)}(t)}{\partial w_{ij}^{(l)}}, \quad (2)$$

where t_U is an integration time over the dynamics of the membrane potential. For the gradient, we use a threshold-shifted sigmoid curve,

$$\tilde{S} = \frac{1}{1 + e^{\theta - U(t)}}, \quad \frac{\partial \tilde{S}}{\partial U} = (1 - \tilde{S})\tilde{S}. \quad (3)$$

Hyperparameters were chosen according to table 1. The training section was composed by an epoch, a pass over the training set, where the neural network attained an accuracy above of 80% depending of dataset. Accuracy is measured by attribution the classification label to the output neuron with the highest firing rate. Figure 2 shows the loss and accuracy as a function of the number of subsets (mini-batches) of previously randomly shuffled training set and test set. Training time refers to each interaction over mini-batches of the dataset. The confusion matrix, a table of true positive and false positive matches according to the data labels is shown in Figure 2.C. Training shows three distinct phases. A short initial phase where the network still cannot implement the task, a fast rising effective learning phase, and an almost saturated phase showing good performance. After training, the confusion table becomes practically a diagonal matrix, indicating that the model classification is most likely a true positive. Figure 3 shows the firing rates of the correct output showing different learning time thresholds for the different categories being learned.

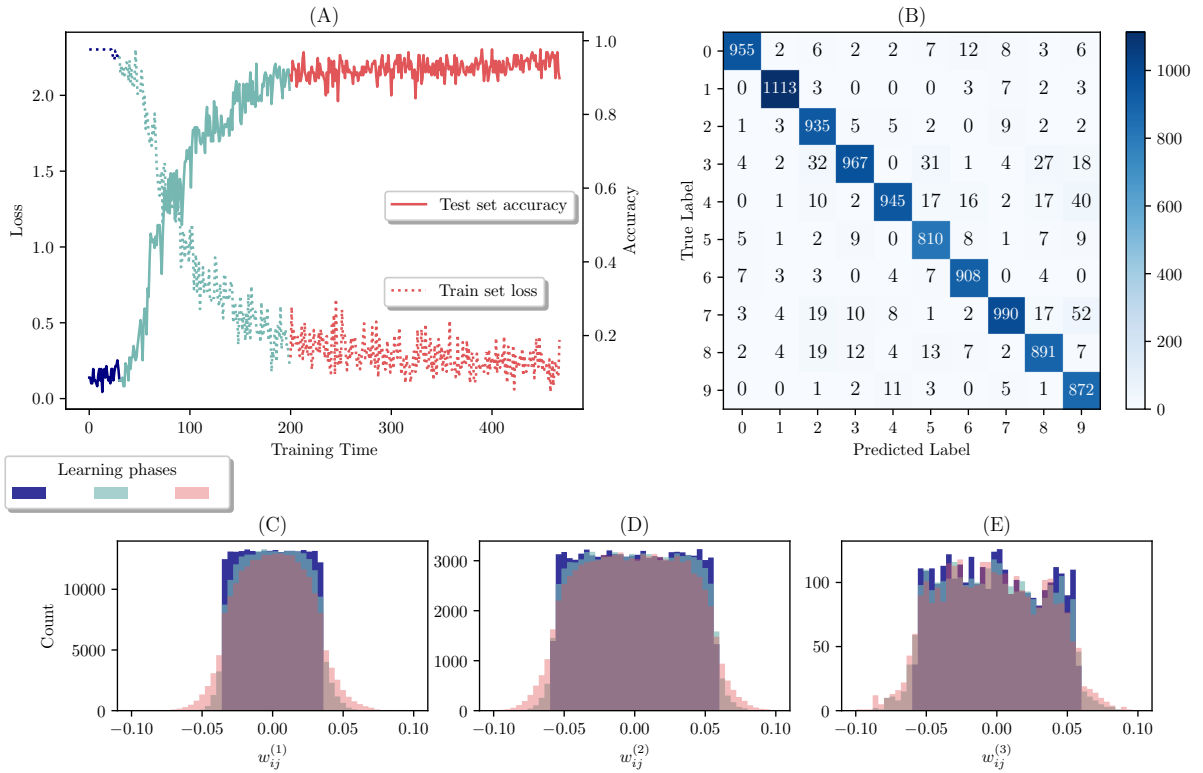


Figure 2: Training session over MNIST train and test set. (A) Loss (left axis) and Accuracy (right axis). Three different phases can be seen: an initial plateau or slowly rising accuracy phase, a fast rising and a saturated plateau. (B) Confusion matrix over the test set after the training. (C), (D) and (E) The evolution of the empirical distributions of weights between the layers, during training, show no sign of topological criticality. The results for the Fashion-MNIST data set are not shown since they are not significantly different.

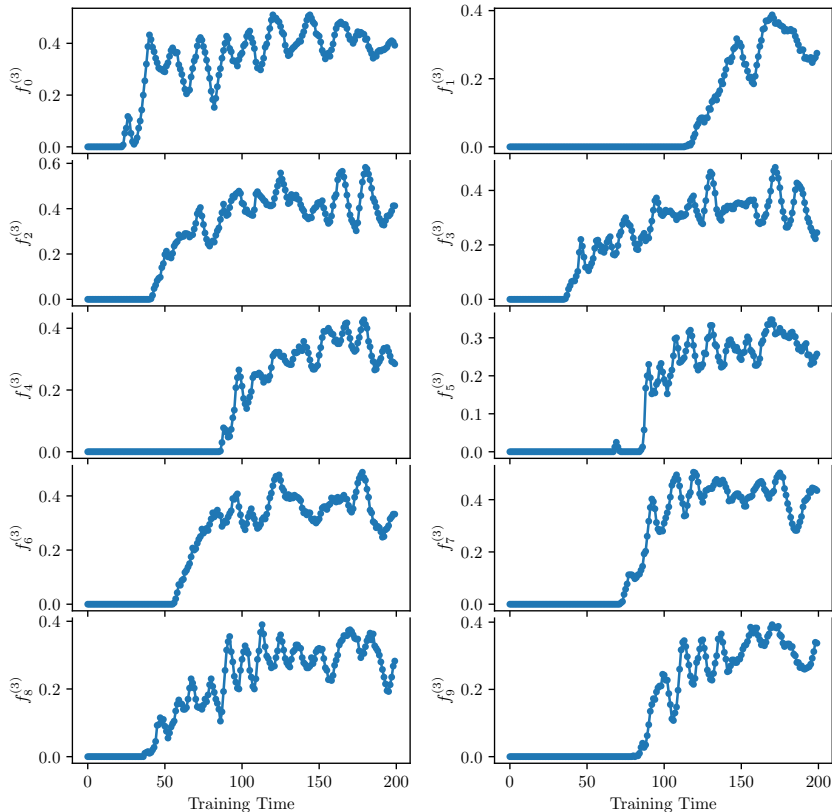


Figure 3: Evolution with training time of the firing rate $f_k^{(3)}$ of output neuron k when input patterns of type k are presented. Note that it takes different training times until the network can identify different categories. MNIST dataset.

2.2 The mathematical characterization

We use the following notation for the time average at time t of any quantity f in a time window of width Δt

$$\langle f(t) \rangle = \frac{1}{\Delta t} \sum_{t'=t}^{t+\Delta t} f(t'). \quad (4)$$

When f , for fixed t , is an array, so is the above expression and the sums are point wise. The fluctuation of a quantity is $\bar{f} = f - \langle f \rangle$. The un-normalized truncated correlation is

$$c_{f,g}(t, \tau, \Delta t) = \langle \bar{f}(t) \bar{g}(t - \tau) \rangle, \quad (5)$$

$\bar{f} \bar{g}$ is a dot product for arrays, and the normalized correlation of fluctuations or the truncated correlation is

$$C_{f,g}(t, \tau, \Delta t) = \frac{c_{f,g}}{\sqrt{c_{f,f} c_{g,g}}}. \quad (6)$$

The spike activity in the l^{th} layer is described by an $N^{(l)}$ -dimensional array at time t

$$\mathbf{S}^{(l)}(t) = \left[S_1^{(l)}(t), \dots, S_{N^{(l)}}^{(l)}(t) \right], \quad (7)$$

where $N^{(l)}$ is the number of neurons and $S_i^{(l)} = \{0, 1\}$, the binary output specifying whether neuron i has fired at t . To characterize the fluctuations of internal representations in layers l^{th} and k^{th} we use $C_{\mathbf{S}^{(l)}, \mathbf{S}^{(k)}}(t, \tau, \Delta t)$.

We also define the following macroscopic variables (aggregated): the average activity of the l^{th} layer at time t

$$\rho^{(l)}(t) = \frac{1}{N^{(l)}} \sum_{i=1}^{N^{(l)}} S_i^{(l)}(t), \quad (8)$$

and the average membrane potential in the l^{th} layer at time t

$$u^{(l)}(t) = \frac{1}{N^{(l)}} \sum_{i=1}^{N^{(l)}} U_i^{(l)}(t). \quad (9)$$

In the simulations we measure $C_{\mathbf{S}^{(l)} \mathbf{S}^{(l)}}(t; \tau, \Delta t)$, $C_{\rho^{(l)} \rho^{(l')}}(t; \tau, \Delta t)$ and $C_{u^{(l)} u^{(l')}}(t; \tau, \Delta t)$, functions of the three parameters t , τ and Δt . From an analogy to spins systems, we expect these correlations to decay with τ as the product of a fast decaying exponential with characteristic time ξ_τ and a slow decaying algebraic function. If criticality is approached ξ_τ grows and there is a crossover to purely algebraic decay, but only in the thermodynamic limit. For finite systems an exponential function is a good model and we consider

$$C(\tau; t, \Delta t) \sim \exp \left\{ -\frac{\tau}{\xi_\tau} \right\}. \quad (10)$$

ξ_τ plays a similar role as the correlation length. Fixing t and Δt , we can see how the temporal correlation varies with τ . The correlation time can be computed for each time step t from a fit to equation 10.

We also define what we mean in this context by avalanches of activity and analyze their distribution. We say that a ρ avalanche in the hidden layers, is happening when $\rho(t) = \sum_l \rho^{(l)}(t)$, summed over hidden layers, is above its time average \bar{R} . An avalanche's size is the integrated area below $\rho(t)$ between the crossings to $\rho > \bar{R}$ and back to $\rho < \bar{R}$, and the duration is the difference in time between those consecutive occurrences.

2.3 Simulations

We use ‘‘movies’’, sequences of images described in Table 2 as inputs to the NN during the simulations. The set of numerical experiments is shown in Table 3. α is a parameter introduced to simulate partial blocking of the input to the network. Every pixel of an input pixel is multiplied by α in the interval $(0, 1]$. It is a simplistic model of the effect of anesthesia acting only on the input layer. It is used just to study the pattern recognition of a dim input and not as a model of the effects of a particular type of anesthesia itself.

Table 2: The “movies”

Movie type	Composition	Description	δt	Measures
M_0	Random inputs	New random image every time step	none	ξ_τ
M_S	2 or more classes	Few slow transitions between classes	$\delta t \approx 400-500$	ξ_τ
M_F	2 or more classes	Several fast transitions between classes	$\delta t \sim t_U \sim 25$	Avalanches and Complexity Index

For M_S and M_F , δt is the number of presentations of random choices of examples in the same class between transitions. It measures the variability of the environment presented to the NN.

Table 3: Simulations.

Training Phase	Movie	Conditions
early	M_0	$\alpha = 1$
intermediate	M_0, M_S, M_F	$0 < \alpha \leq 1$
saturated	M_0, M_S, M_F	$0 < \alpha \leq 1$

The intensity of the “anesthesia” is $1 - \alpha$.

3 Results

At a given point in the training time (Figure 2) we study the IR that occur when a “movie” is presented to the NN. We start analyzing the fully trained network. Several types of movies can be presented, see Table 2. First, we present a movie of M_S type for intervals comprising ~ 500 frames random examples of one of the ten categories is shown, then for another ~ 500 frames, examples from another category are presented. Nothing interesting happens except for a small region in time starting at the transition with large and persistent fluctuations, see Figure 4. In Figure 5 we show results from a typical run with several input class changes. The NN is working satisfactorily as shown in 5.A, since the output cell with the dominant $U^{(3)}$ is the correct one.

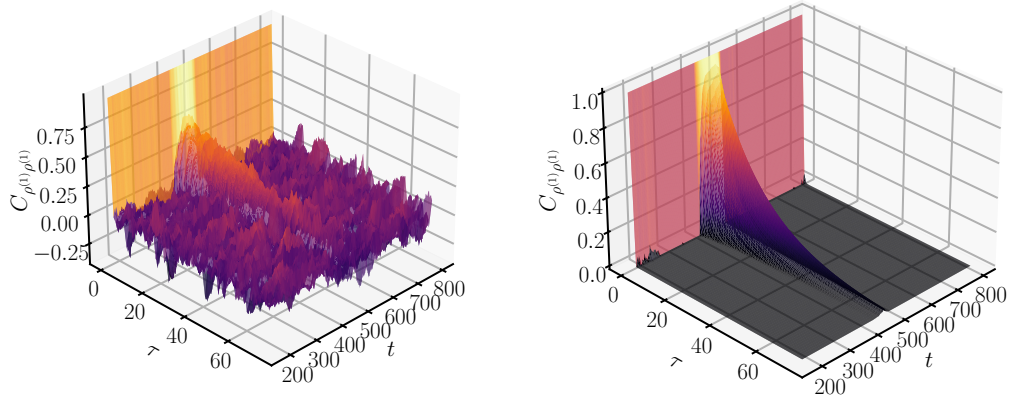


Figure 4: IR fluctuations temporal correlation as a function of τ measured on a sliding window centered at time t for the single image change movie. Left: the raw measured correlations. Right: For each t , the best fit of an exponential decay $e^{-\frac{\tau}{\xi_\tau}}$, equation 10. At the region of the transition of images, the decay time ξ_τ increases a few orders of magnitude.

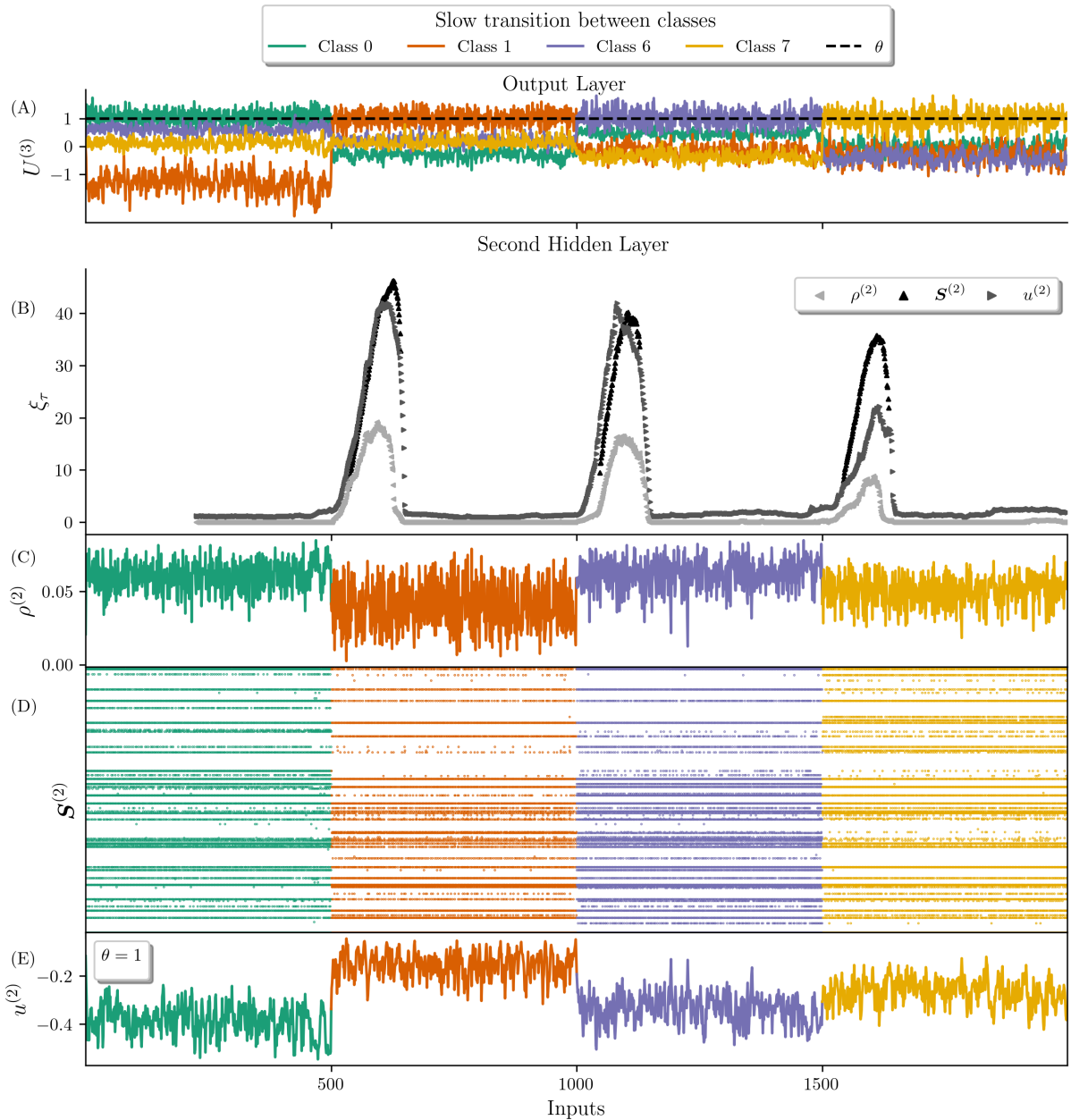


Figure 5: The M_S movie input (no anesthesia $\alpha = 1$) contains a sequence of 500 random images of label “0” (green), then 500 of label “1” (orange), 500 of label “6” (purple), and 500 of “7” (yellow) from the MNIST database. (A) The membrane potential of output neurons responsible for classification of labels during the movie, (B) The characteristic time of decay of correlations of the fluctuations of the average activity, of spikes and average membrane potentials of the second hidden layer. (C) Average activity. (D) Spike train: raster plot of a few neurons of hidden layer 2. (E) Average membrane potential.

Figure 5.B shows results for the second hidden layer IR. A very large increase in ξ_τ following the input label transitions. A similar behavior, with smaller peaks, is seen for the IR in the first hidden layer. While the NN moves rapidly toward the new classification, IR show persistent fluctuations, around the transitions. Is this peak in time correlation length the signature of criticality? We argue in section 4 that it is not.

A second numerical experiment investigates the results of partial training. We show results in Figure 6 for a training time of 25, when as Figure 3 shows, only category “0” is partially recognizable in the output layer. It may be expected that since the category attractors are not yet fully developed, persistent fluctuations should not appear. However, despite not achieving correct classification over all data labels, such persistence occurs during the transition of input categories. This is observed in distinct cases, first during the transition of a correctly classified input to one not yet learned (“0” \rightarrow “1”). Second, when the neural network still has not learned both input labels (“1” \rightarrow “6” \rightarrow “7”). This is evidence that during the learning process, attractors of the dynamics are partially present in the hidden layer, while still sub-threshold in the output layer.

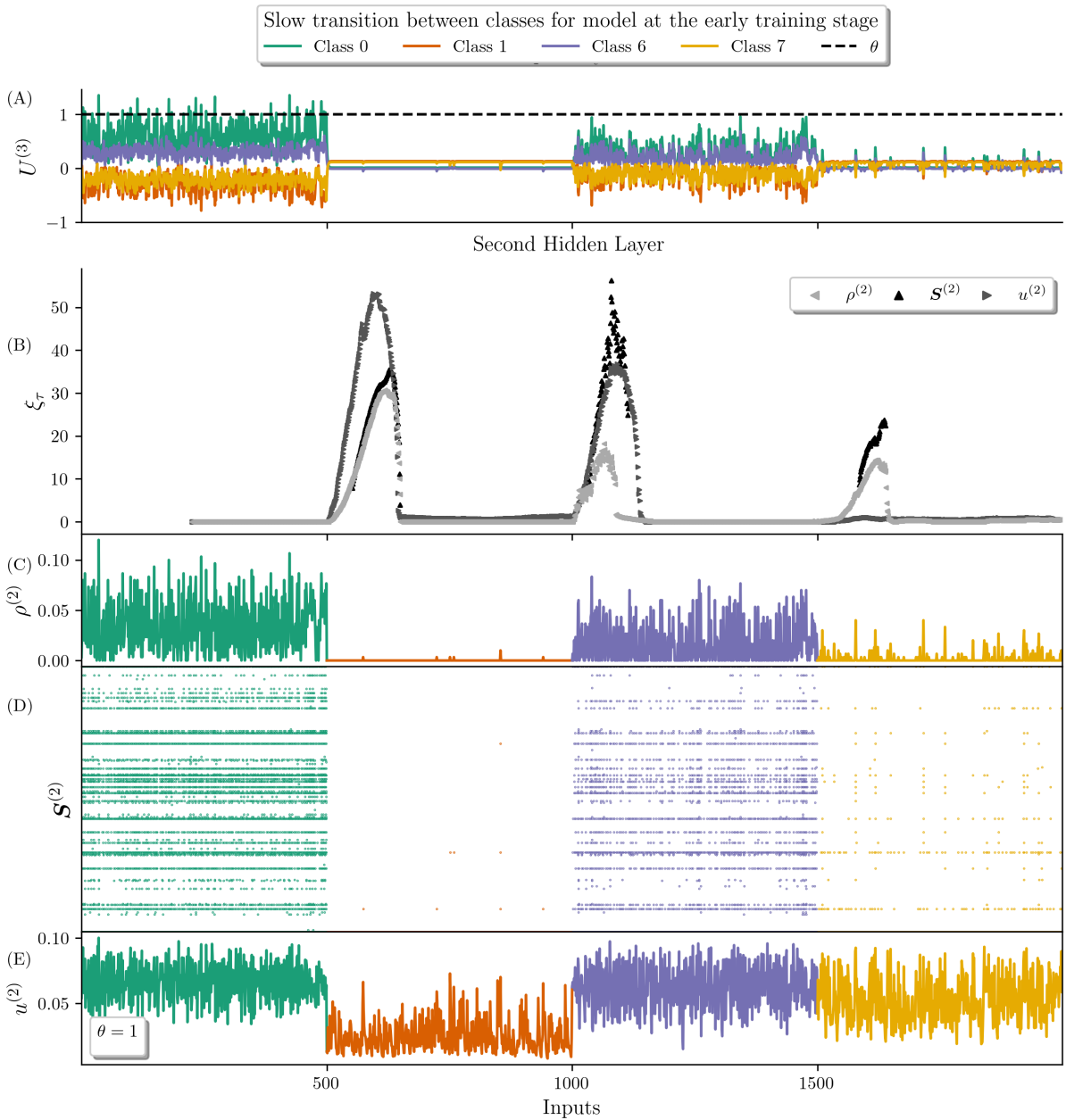


Figure 6: Same as Figure 5 but for a partially trained NN, where only “0” is recognizable. Transition regions show persistent fluctuations, independently of output recognition.

In a third experiment, the fully trained network, used in figure 5, is presented to a M_S movie with a dimmed input, every pixel input is multiplied by $\alpha < 1$, which we call a model for anesthesia. Figure 7 shows again that for inputs on which the output is not correct due to diminished input sensibility, the sub-threshold images can elicit persistent correlations with large characteristic times. Finally, we notice there is no peak presence in correlation time when movie M_0 is being exposed. These results, which only serves as a control test, are not shown.

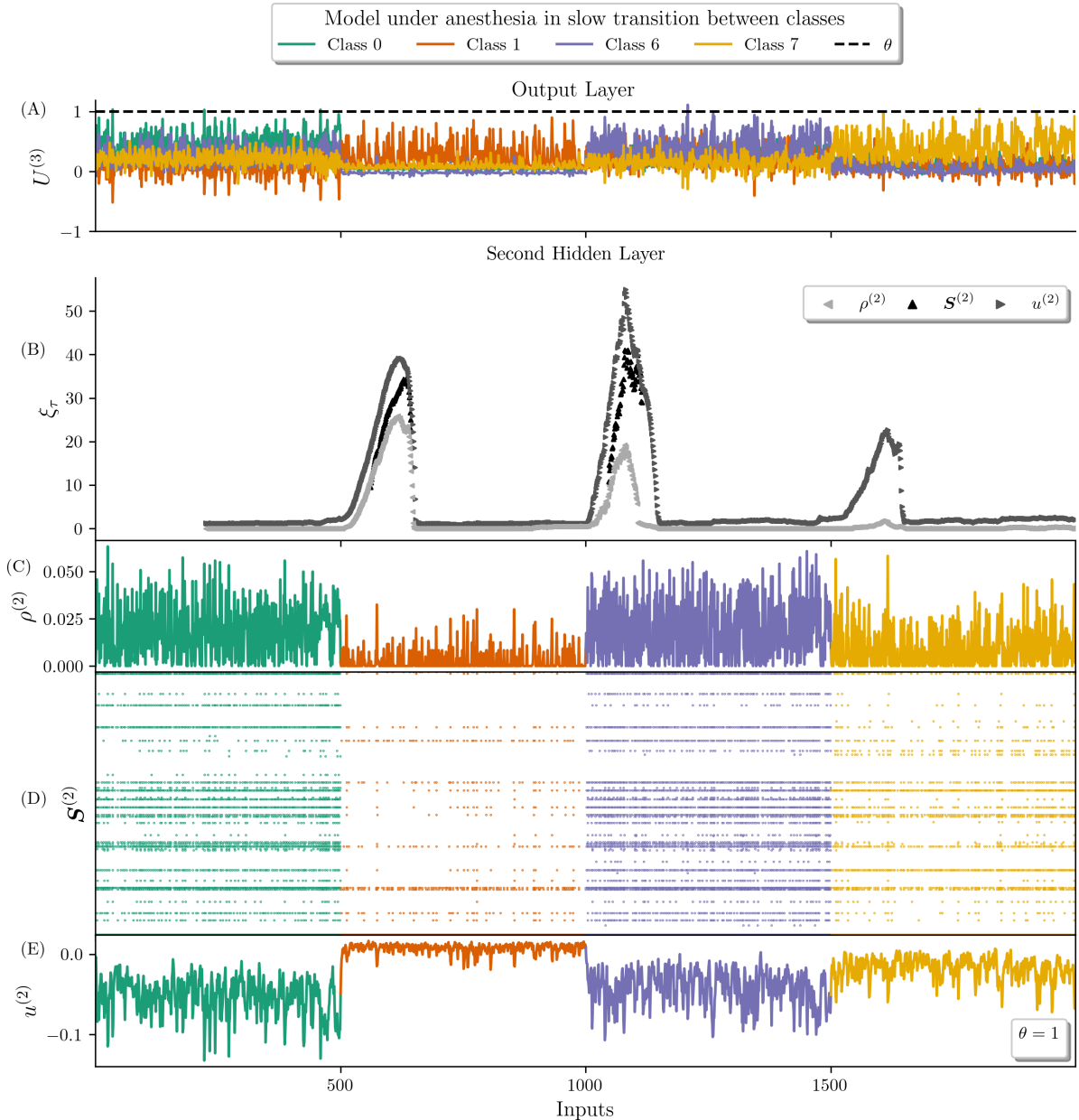


Figure 7: Same as Figure 5 but with neural network under effect of anesthesia, $\alpha = .4$, with a M_S input. (A) The network is not able to correctly categorize the inputs, which generate sub threshold activity at the output layer: there is no communication of the computation to the outside world. (B) Nevertheless, the transition of inputs generates persistent fluctuation correlation. (C, D) show the activity in the second hidden layer, but (E) $u^{(2)}$ fails to exceed the threshold.

We use the M_F movie to generate several fast input transitions so that the final output alternates between different categories, to simulate a free moving agent in a rich environment. This input generates a large amount of variability in the IR and avalanches can be measured. Histograms are shown in Figure 8.A and B. While we kept the number neurons in the hidden layers equal $N^{(l)} = N$ for all hidden layers, we ran simulations with different values of N (50 and from 100, 200, \dots to 1200 .)

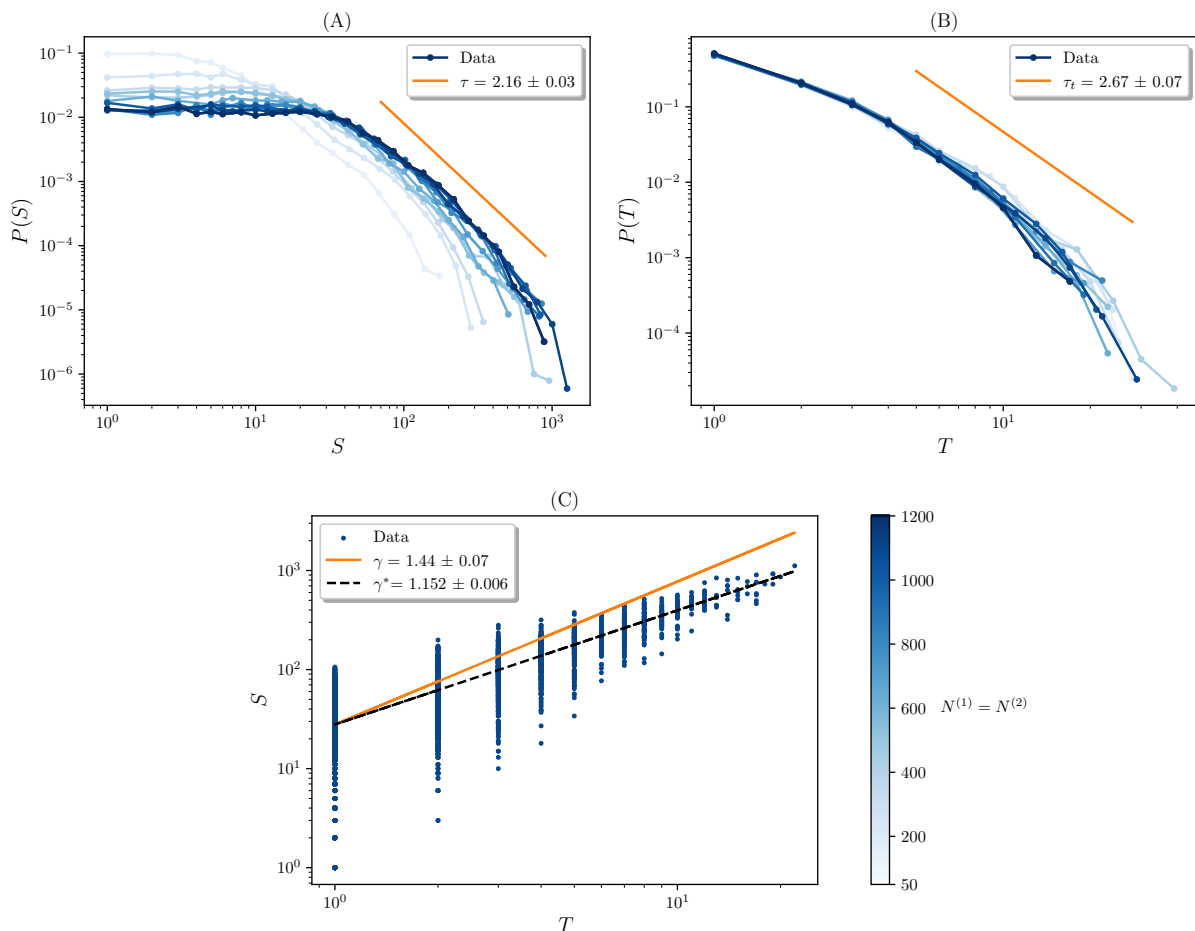


Figure 8: (A) Size and (B) duration avalanches. Power law fit using [38] are very sensitive to the choices of the minimum and maximum values.

Each point in Figure 8.C shows the duration and size of an avalanche. For the characterization of power laws we use a standard procedure [38]. The size and duration exponents τ, τ_t give a crackling exponent [31]

$$\gamma = \frac{\tau_t - 1}{\tau - 1} \approx 1.44. \quad (11)$$

A direct fit of a power law in Figure 8.C yields a different $\gamma^* \approx 1.15$. The uncertainty of the fits, from [38] are too optimistic, being rather of the order of .2, not 0.03. It is not possible to affirm that power laws are present in this case. Note that the procedures that yield these exponents are very sensitive, see e.g. [39], not only to choices of maximum and minimum values, but also to sub-sampling and to the noise process that is used to model the fluctuations in the data. Since we are not defending this to be a signature

of criticality we don't delve into the details of the fit. If however the reader finds this a compelling argument for criticality, which we don't, it is clearly not in the class of directed percolation.

4 Broad tails without criticality

4.1 Langevin dynamics on a time dependent potential

We now present a toy problem which is not critical in any sense and yet, it yields broad tail distribution in the style of what was found in the neural network of the previous section.

With $\mathbf{r}, \mathbf{r}_0 \in R^2$, let $V(\mathbf{r}, \pm\mathbf{r}_0)$ be potential wells with a single minimum at $\pm\mathbf{r}_0$. We construct a time dependent potential

$$U(\mathbf{r}, t) = h_1(t)V(\mathbf{r}, \mathbf{r}_0) + h_2(t)V(\mathbf{r}, -\mathbf{r}_0), \quad (12)$$

where for example we choose $h_1 = |\cos(|\omega t|)|^k$ and $h_2 = |\sin(|\omega t|)|^k$ and use as a typical example $k = 6$. The potential changes from one well to another in periodic fashion, located at $\pm\mathbf{r}_0$. The process described by the discrete time Langevin dynamics

$$\mathbf{r}_{t+\Delta t} = \mathbf{r}_t - \eta \nabla U(\mathbf{r}, t) + \mathbf{W}, \quad (13)$$

with W a two dimensional normal random process. \mathbf{r} has a similar behavior to that of the IR of the previous section. It evolves from being around a minimum, drifting to the other as it becomes prominent, Figure 9. The dynamics obviously is not critical, nevertheless it has long time persistence of the correlation decay when there is no clear minimum present, as shown in Figure 10.

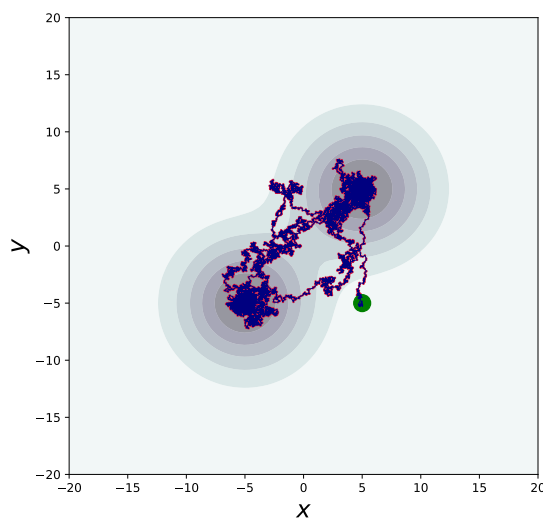


Figure 9: A trajectory of the two dimensional model equation 13 starting from the green dot. It spends time in one well until it becomes very shallow and then wanders towards the other minimum that is becoming dominant. The contours show the time average of the potential. Minima at $\pm\mathbf{r}_0 = \pm(5, 5)$, $\eta = 0.15$, $\sigma_W = 0.1$.

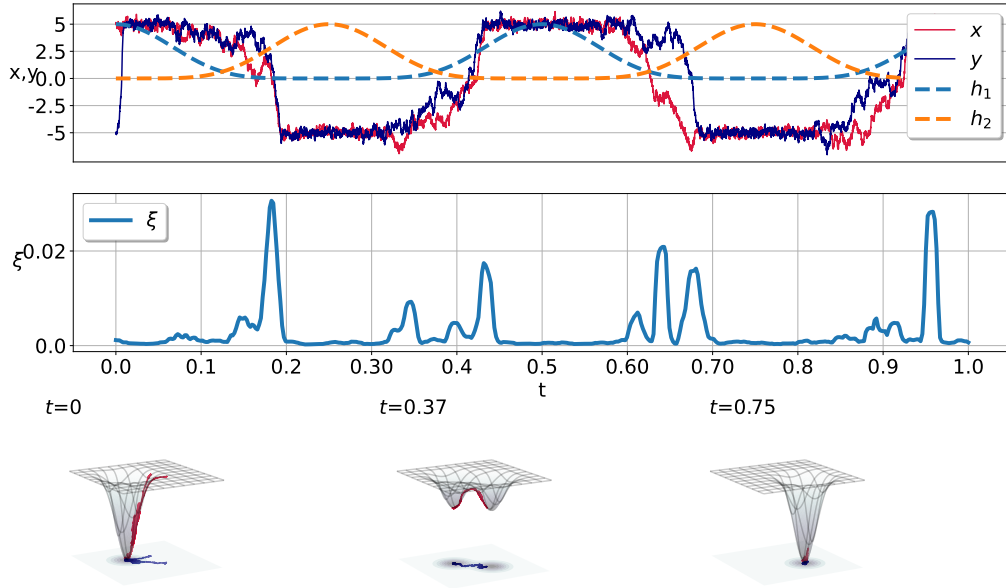


Figure 10: Top: Coordinates x and y of a particle that evolves under the dynamics of equation 13. $h_{1,2}$ are proportional to the depths of the wells. Center: The continuous line shows a moving average of the characteristic time ξ of decay of the auto-correlation of the position of the particle, showing large peaks during the transitions. Bottom: The 3d potentials drawn at the time coordinate. Moving average over window size = 7 ticks = .5ms)

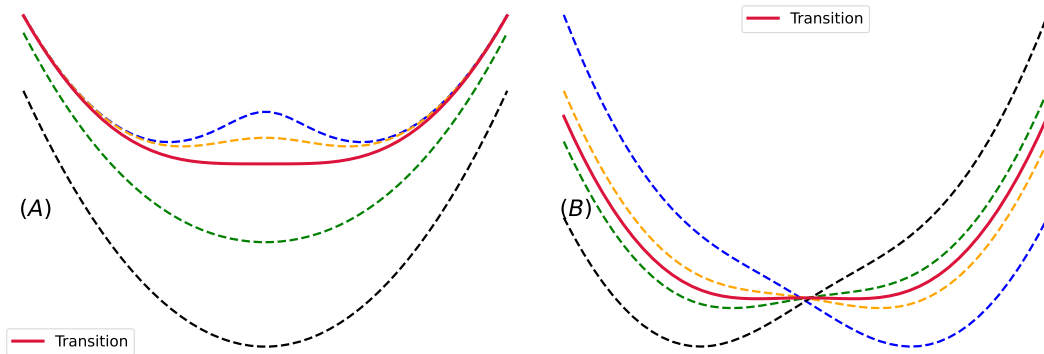


Figure 11: (A) Free energy functional of an infinite range Ising model at zero external field, undergoing a critical second order transition between the disordered phase (black, green dashed curves) and the broken symmetry phase (orange, blue dashed lines). (B) Same model at temperature above critical. As the external field changes, the expected value of the magnetization transitions between two different ordered states ($h > 0$ orange and blue dashed lines, $h < 0$ black and green dashed lines). For both graphs, transition points represented by the continuous red line.

A second analogy is based on the infinite range Ising spin system in an external field,

which plays the role of the input pattern to the neural network of section 2.3,

$$H = -\frac{1}{2}\left(\sum_{i=1}^N s_i\right)^2 + \sum_i h_i s_i \quad (14)$$

At each site i the magnetic field changes very slowly between the value ξ_i^1 and ξ_i^2 . If the change is sufficiently slow, we can solve for the thermodynamics of the model for an external field fixed in time, and at each site the magnetization is obtained from

$$m_i(t) = \tanh(\beta(m_i(t) + h_i(t))) \quad (15)$$

If $h_i = 0$, the exact solution is a minimum of the free-energy functional shown in Figure 11.A, with the typical second order transition occurring at the critical value $\beta = 1$. Now we consider a case where $\beta < 1$, so the system is not critical and h_i changes slowly in time. Then m_i follows the minimum of the curve shown in 11.B. In both cases, fluctuations measured in a time scale faster than the slow change of the external field, show broad tails independent of being critical or not.

4.2 Monte Carlo Renormalization Group: convolutional classifier of the thermodynamic state

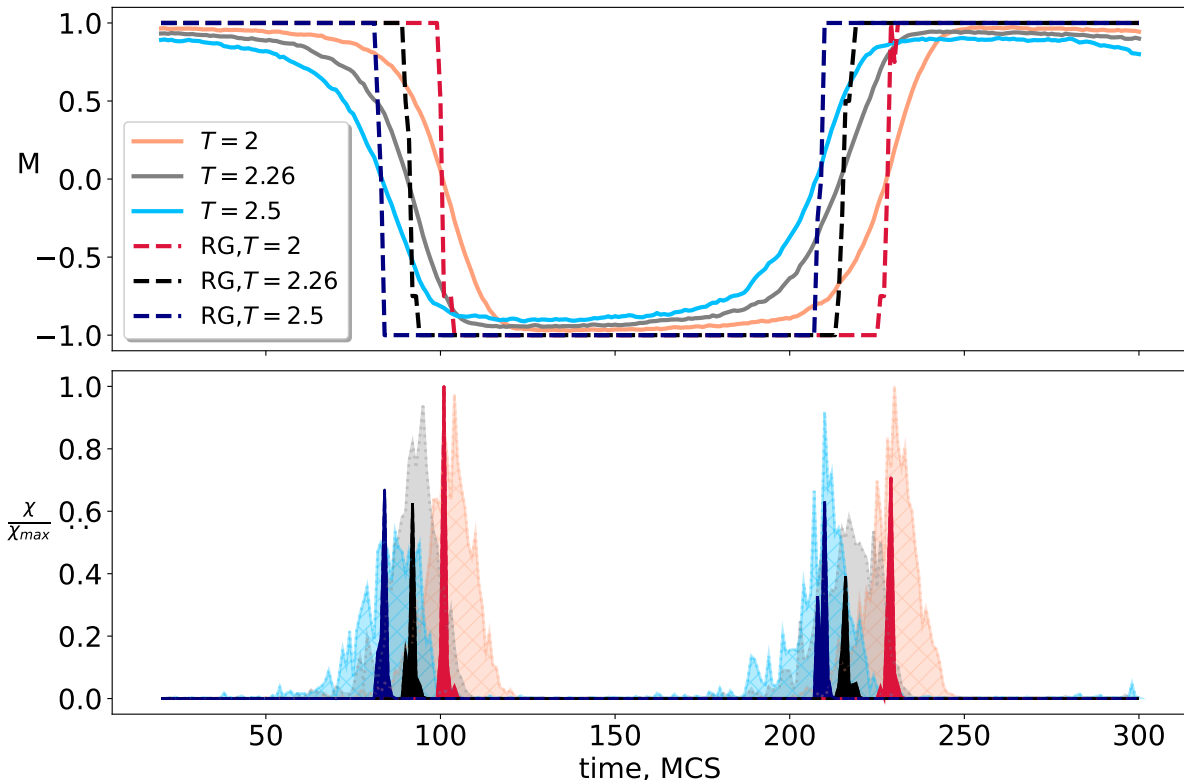


Figure 12: Monte Carlo Metropolis simulation of 2d Ising model with a time dependent external field for temperatures below critical, critical and above critical: $T_B = 2 < T_c \approx 2.26 < T_A = 2.5$. Short time averages (equations 16, $r = .9$) for top: Magnetization. Continuous lines for original Ising (128×128 lattice, nearest neighbor interaction), dashed lines for system renormalized by a factor $b = 2$, six times. Bottom: Susceptibility, divided by its maximum value. The susceptibilities of the renormalized system are much larger (≈ 200 times) than those of the unrenormalized, and are non zero during a much shorter time, since the transition to a new classification is much faster.

We present a third analogy of a non critical system with large fluctuations. We use the MCRG algorithm as developed by Swendsen [29], for the nearest-neighbor model in 2d in the presence of a slowly changing external uniform field. We use the majority rule with a scale renormalization parameter $b = 2$. Figure 12 shows the results of the simulation as a function of time, measured in Monte Carlo steps, for temperatures above, below and at the critical temperature. Since we are not interested in the well known equilibrium properties of the model, our order parameters are the magnetization $M(t)$ and susceptibility and $\chi(t)$ estimated by a moving average with relaxation:

$$\begin{aligned}
 M(t+1) &= (1-r)M(t) + rm(t), \\
 \mu_2(t+1) &= (1-r)\mu_2(t) + rm(t)^2, \\
 \chi(t+1) &= \mu_2(t+1) - M(t+1)^2,
 \end{aligned} \tag{16}$$

where $m(t) = \sum_k s_k / N_{RG}$, for a renormalized configuration with N_{RG} spins.

In Figure 12 (top) we show the magnetization as the external field changes for three temperatures. The continuous lines are for the unrenormalized model on a 128×128 square lattice and the dashed lines for the majority rule MCRG, scale parameter $b = 2$, 6 times renormalized. Two relevant things can be seen; first, the transitions are qualitatively the same, independent of temperature. Second, the magnetizations characterize the macroscopic state that the system would be if the field had been kept constant. Notice that the renormalized magnetization has a much sharper transition of classification than the unrenormalized magnetization. The MCRG magnetization can be seen as the result of a classifier analogous to a convolutional Neural Network, where regions of size $b \times b$ are used to define the “activity” or Kadanoff block spin, in the next layer, i.e the next renormalization stage. The “movie” presented to this network is generated by the Monte Carlo procedure. In Figure 12 (bottom) we show, with the same color code, the susceptibilities. The peak at the transitions show that the Monte Carlo dynamics is exploring larger regions of configurations than when the field has a clear sign. The renormalized susceptibilities are larger in magnitude (~ 200 times) but present much sharper peaks, consistent with the abrupt change in magnetization. the indicator of the classification of microscopic states into macroscopic state. This MCRG uses a predetermined convolutional filter, the simple majority rule. A tunable version [40, 41], with probabilistic filters shows, for the Ising 3d nearest neighbor model, a much faster approach to the critical point than with the majority rule. In the language of neural networks this means that adjustments in the weights of the filters may lead to the possibility of the reduction of the number of hidden layers need to converge to a classification.

5 Conclusions

We have presented results from simulations of spiking neural networks trained as a classifier. The change in the environmental stimulus induces broad tail distributions of the truncated correlation of fluctuations of activity in the internal representations in the NN. Before and after this transient period the network settles into a state where these correlations show very short characteristic time or reduced fluctuations variability. This is similar to results in [42], where the “stimulus onset quenches neural variability”, which they claim to be a rather general property of the cortex. These broad distributions after input transitions, are found even if we disturb the performance of the classifier by incomplete training or decreased intensity of the input, a simple model of an animal under the effect of anesthesia. We have defined avalanches and tried to fit power laws to the empirical distributions of size and duration, for the partially and fully trained NN as well as for the anesthetized model. Crackling noise scaling relations show that, even if there is a critical state, it is not in the directed percolation universality class. But the large increase of ξ_r is not a criticality signature. Instead we argue that the role of the external stimulus, analogous to a magnetic field, is to lead the dynamics from a disappearing basin of attraction to a new one. The crossover from one attractor to another leads to large fluctuations without criticality. To illustrate cases where this occurs, we discussed, first a toy model where a particle moves in two dimensions under the influence of an oscillating potential and normal noise. Second, an infinite range Ising model in an oscillating external field, exactly solvable under the separation of time scales of fast spins and slowly evolving field. Third, we present a MCRG study of an Ising model in 2 dimensions. The interesting thing is that the renormalization of the configurations is

mathematically equivalent to a feed-forward convolutional neural network, acting as a classifier. The increases in susceptibility driven by a changing field, are present below, at and above the critical temperature. For the renormalized system the duration of the susceptibility decreases and its height dramatically increases. These are indications that the renormalization procedure, that filters high frequency spatial Fourier components, is selecting the correct degrees of freedom to classify the thermodynamic state. The experiments with the spiking neural network show that in a changing environment, where the external world stimulus has to drive the dynamics of the network toward a meaningful interpretation, persistent fluctuations of the internal representation will appear as a result of the search for the new basin of attraction associated to a concept that represents the external world. Feed forward neural networks are certainly not models of the brain, but, certainly the cortex of freely behaving and even anesthetized animals, are constantly stimulated by changing inputs, which have to be processed by extracting relevant information to drive behavioral decisions. This can induce changes in the basin of attraction of the different concepts being identified and this search for the new attractor, generate persistent fluctuations of neuronal activity.

Acknowledgments

JHS was supported by a CAPES-Brasil fellowship. This work received support from CNAIPS-USP.

References

- [1] John M. Beggs and Dietmar Plenz. “Neuronal Avalanches in Neocortical Circuits”. In: *Journal of Neuroscience* 23.35 (2003), pp. 11167–11177. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.23-35-11167.2003. eprint: <https://www.jneurosci.org/content/23/35/11167.full.pdf>. URL: <https://www.jneurosci.org/content/23/35/11167> (cit. on p. 2).
- [2] Osame Kinouchi and Mauro Copelli. “Optimal dynamical range of excitable networks at criticality”. In: *Nature Physics* 2.5 (May 2006), pp. 348–351. ISSN: 1745-2481. DOI: 10.1038/nphys289. URL: <https://doi.org/10.1038/nphys289> (cit. on p. 2).
- [3] Heinz Georg Schuster. *Criticality in neural systems*. John Wiley & Sons, 2014. ISBN: 9783527651009. DOI: 10.1002/9783527651009 (cit. on p. 2).
- [4] Mauricio Girardi-Schappo. “Brain criticality beyond avalanches: open problems and how to approach them”. In: *Journal of Physics: Complexity* 2.3 (Sept. 2021), p. 031003. DOI: 10.1088/2632-072X/ac2071. URL: <https://dx.doi.org/10.1088/2632-072X/ac2071> (cit. on p. 2).
- [5] Viola Priesemann, Matthias HJ Munk, and Michael Wibral. “Subsampling effects in neuronal avalanche distributions recorded in vivo”. In: *BMC Neuroscience* 10.1 (Apr. 2009), p. 40. ISSN: 1471-2202. DOI: 10.1186/1471-2202-10-40. URL: <https://doi.org/10.1186/1471-2202-10-40> (cit. on p. 2).

- [6] Miguel A. Muñoz. “Colloquium: Criticality and dynamical scaling in living systems”. In: *Rev. Mod. Phys.* 90 (3 July 2018), p. 031001. DOI: 10.1103/RevModPhys.90.031001. URL: <https://link.aps.org/doi/10.1103/RevModPhys.90.031001> (cit. on p. 2).
- [7] Antonio J. Fontenele et al. “Criticality between Cortical States”. In: *Phys. Rev. Lett.* 122 (20 May 2019), p. 208101. DOI: 10.1103/PhysRevLett.122.208101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.122.208101> (cit. on p. 2).
- [8] Bruno Del Papa, Viola Priesemann, and Jochen Triesch. “Criticality meets learning: Criticality signatures in a self-organizing recurrent neural network”. In: *PLOS ONE* 12.5 (May 2017), pp. 1–21. DOI: 10.1371/journal.pone.0178683. URL: <https://doi.org/10.1371/journal.pone.0178683> (cit. on p. 2).
- [9] T. GISIGER. “Scale invariance in biology: coincidence or footprint of a universal mechanism?” In: *Biological Reviews* 76.2 (2001), pp. 161–209. DOI: <https://doi.org/10.1017/S1464793101005607>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1017/S1464793101005607>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1017/S1464793101005607> (cit. on p. 2).
- [10] C. Bédard, H. Kröger, and A. Destexhe. “Does the $1/f$ Frequency Scaling of Brain Signals Reflect Self-Organized Critical States?” In: *Phys. Rev. Lett.* 97 (11 Sept. 2006), p. 118102. DOI: 10.1103/PhysRevLett.97.118102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.97.118102> (cit. on p. 2).
- [11] Jonathan Touboul and Alain Destexhe. “Power-law statistics and universal scaling in the absence of criticality”. In: *Phys. Rev. E* 95 (1 Jan. 2017), p. 012413. DOI: 10.1103/PhysRevE.95.012413. URL: <https://link.aps.org/doi/10.1103/PhysRevE.95.012413> (cit. on p. 2).
- [12] J J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554> (cit. on p. 2).
- [13] Daniel J Amit, Hanoach Gutfreund, and H Sompolinsky. “Statistical mechanics of neural networks near saturation”. In: *Annals of Physics* 173.1 (1987), pp. 30–67. ISSN: 0003-4916. DOI: [https://doi.org/10.1016/0003-4916\(87\)90092-3](https://doi.org/10.1016/0003-4916(87)90092-3). URL: <http://www.sciencedirect.com/science/article/pii/0003491687900923> (cit. on p. 2).
- [14] E. Gardner. “The space of interactions in neural network models”. In: *Journal of Physics A: Mathematical and General* 21.1 (Jan. 1988), p. 257. DOI: 10.1088/0305-4470/21/1/030. URL: <https://dx.doi.org/10.1088/0305-4470/21/1/030> (cit. on p. 2).
- [15] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001. DOI: 10.1017/CB09781139164542 (cit. on p. 2).
- [16] Michael Biehl. *The Shallow and the Deep: A biased introduction to neural networks and old school machine learning*. University of Groningen, 2023. ISBN: 9789403430270. DOI: <https://doi.org/10.21827/648c59c1a467> (cit. on p. 2).

- [17] Douglas R Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, Inc. Division of HarperCollins, 1979 (cit. on p. 2).
- [18] Franz J. Wegner and Anthony Houghton. “Renormalization Group Equation for Critical Phenomena”. In: *Phys. Rev. A* 8 (1 July 1973), pp. 401–412. DOI: 10.1103/PhysRevA.8.401. URL: <https://link.aps.org/doi/10.1103/PhysRevA.8.401> (cit. on p. 3).
- [19] Pedro Pessoa and Ariel Caticha. “Exact Renormalization Groups As a Form of Entropic Dynamics”. In: *Entropy* 20.1 (2018). ISSN: 1099-4300. DOI: 10.3390/e20010025. URL: <https://www.mdpi.com/1099-4300/20/1/25> (cit. on p. 3).
- [20] Ariel Caticha. “Entropic Dynamics”. In: *Entropy* 17 (9 Sept. 2015), pp. 6110–6128. DOI: doi:10.3390/e17096110. URL: <http://www.mdpi.com/1099-4300/17/9/6110> (cit. on p. 3).
- [21] Pankaj Mehta and David J. Schwab. *An exact mapping between the Variational Renormalization Group and Deep Learning*. 2014. arXiv: 1410.3831 [stat.ML] (cit. on p. 3).
- [22] Maciej Koch-Janusz and Zohar Ringel. “Mutual information, neural networks and the renormalization group”. In: *Nature Physics* 14.6 (2018), pp. 578–582. ISSN: 1745-2481. DOI: 10.1038/s41567-018-0081-4. URL: <https://doi.org/10.1038/s41567-018-0081-4> (cit. on p. 3).
- [23] Shuo-Hui Li and Lei Wang. “Neural Network Renormalization Group”. In: *Phys. Rev. Lett.* 121 (26 Dec. 2018), p. 260601. DOI: 10.1103/PhysRevLett.121.260601. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.121.260601> (cit. on p. 3).
- [24] Jui-Hui Chung and Ying-Jer Kao. “Neural Monte Carlo renormalization group”. In: *Phys. Rev. Res.* 3 (2 June 2021), p. 023230. DOI: 10.1103/PhysRevResearch.3.023230. URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.3.023230> (cit. on p. 3).
- [25] Nestor Caticha. “Entropic Dynamics in Neural Networks, the Renormalization Group and the Hamilton-Jacobi-Bellman Equation”. In: *Entropy* 22.5 (2020). ISSN: 1099-4300. DOI: 10.3390/e22050587. URL: <https://www.mdpi.com/1099-4300/22/5/587> (cit. on p. 3).
- [26] L.P. Kadanoff. “Scaling laws for Ising models near $T(c)$ ”. In: *Physics Physique Fizika* 2 (1966), pp. 263–272. DOI: 10.1103/PhysicsPhysiqueFizika.2.263 (cit. on p. 3).
- [27] Kenneth G. Wilson and J. Kogut. “The renormalization group and the ϵ expansion”. In: *Physics Reports* 12.2 (1974), pp. 75–199. ISSN: 0370-1573. DOI: [https://doi.org/10.1016/0370-1573\(74\)90023-4](https://doi.org/10.1016/0370-1573(74)90023-4). URL: <http://www.sciencedirect.com/science/article/pii/0370157374900234> (cit. on p. 3).
- [28] Shang-keng Ma. “Renormalization Group by Monte Carlo Methods”. In: *Phys. Rev. Lett.* 37 (8 Aug. 1976), pp. 461–464. DOI: 10.1103/PhysRevLett.37.461. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.37.461> (cit. on p. 3).
- [29] Robert H. Swendsen. “Monte Carlo Renormalization Group”. In: *Phys. Rev. Lett.* 42 (14 Apr. 1979), pp. 859–861. DOI: 10.1103/PhysRevLett.42.859. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.42.859> (cit. on pp. 3, 18).

- [30] O Perkovic, K Dahmen, and J P Sethna. “Avalanches, Barkhausen noise, and plain old criticality”. In: *Phys Rev Lett* 75(24) (1995), pp. 4528–4531. DOI: doi:10.1103/PhysRevLett.75.4528. PMID:10059931. (cit. on p. 3).
- [31] J P Sethna, K A Dahmen, and C R Myers. “Crackling noise”. In: *Nature* 410.6825 (Mar. 2001), pp. 242–250. ISSN: 1476-4687. DOI: 10.1038/35065675. URL: <https://doi.org/10.1038/35065675> (cit. on pp. 3, 14).
- [32] L. Lapique. “Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarisation”. In: *Journal of Physiol Pathol Générale* 9 (1907), pp. 620–635. (Cit. on p. 3).
- [33] L.F Abbott. “Lapicque’s introduction of the integrate-and-fire model neuron (1907)”. In: *Brain Research Bulletin* 50 (1999), pp. 303–304. URL: <https://api.semanticscholar.org/CorpusID:46170924> (cit. on p. 3).
- [34] Peter Dayan and Laurence F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2001. ISBN: 9780262041997. URL: <https://mitpress.mit.edu/9780262041997/> (cit. on p. 3).
- [35] Paul J. Werbos. “Generalization of backpropagation with application to a recurrent gas market model”. In: *Neural Networks* 1.4 (1988), pp. 339–356. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(88\)90007-X](https://doi.org/10.1016/0893-6080(88)90007-X). URL: <https://www.sciencedirect.com/science/article/pii/089360808890007X> (cit. on p. 4).
- [36] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63. DOI: 10.1109/MSP.2019.2931595 (cit. on p. 4).
- [37] Jason K. Eshraghian et al. “Training Spiking Neural Networks Using Lessons From Deep Learning”. In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054. DOI: 10.1109/JPROC.2023.3308088 (cit. on p. 4).
- [38] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. “powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions”. In: *PLOS ONE* 9.1 (Jan. 2014), pp. 1–11. DOI: 10.1371/journal.pone.0085777. URL: <https://doi.org/10.1371/journal.pone.0085777> (cit. on p. 14).
- [39] Leonardo Dalla Porta and Mauro Copelli. “Modeling neuronal avalanches and long-range temporal correlations at the emergence of collective oscillations: Continuously varying exponents mimic M/EEG results”. In: *PLOS Computational Biology* 15.4 (Apr. 2019), pp. 1–26. DOI: 10.1371/journal.pcbi.1006924. URL: <https://doi.org/10.1371/journal.pcbi.1006924> (cit. on p. 14).
- [40] H. W. J. Blöte et al. “Monte Carlo Renormalization of the 3D Ising Model: Analyticity and Convergence”. In: *Phys. Rev. Lett.* 76 (15 Apr. 1996), pp. 2613–2616. DOI: 10.1103/PhysRevLett.76.2613. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.76.2613> (cit. on p. 19).
- [41] Dorit Ron, Achi Brandt, and Robert H. Swendsen. “Surprising convergence of the Monte Carlo renormalization group for the three-dimensional Ising model”. In: *Phys. Rev. E* 95 (5 May 2017), p. 053305. DOI: 10.1103/PhysRevE.95.053305. URL: <https://link.aps.org/doi/10.1103/PhysRevE.95.053305> (cit. on p. 19).

- [42] Mark M. Churchland et al. “Stimulus onset quenches neural variability: a widespread cortical phenomenon”. In: *Nature Neuroscience* 13.3 (Mar. 2010), pp. 369–378. ISSN: 1546-1726. DOI: 10.1038/nn.2501. URL: <https://doi.org/10.1038/nn.2501> (cit. on p. 19).