論文内容の要旨

論文題目 Dynamics of Learning and Statistical Mechanical Informatics of Multilayer Perceptrons: Theory and Practice
多層パーセプトロンの学習のダイナミクスと情報統計力学：理論と適用

氏名 Florent Cousseau クーソー・フローラン

We investigate the influence of the intrinsic structural properties of multilayer perceptron (MLP) neural networks on learning problems, and on information processing problems. It is known that MLPs can approximate any non-linear function up to an arbitrary precision, provided the fact that a sufficient number of hidden units are used. Thus, MLPs are a popular kind of neural networks because of their huge potential. However, this thesis shows that despite the promising theoretical potential of MLPs, actually releasing this potential is hard. The special structural properties of MLPs give rise to singular spaces which prevent the standard algorithms from being efficient. This underlines the need for algorithms especially designed to handle these particular spaces.

Introduction

An artificial neural network is a mathematical model based on biological neural networks. It consists of an interconnected group of artificial neurons (or units) and processes information using a connectionist approach to computation. Artificial neural networks are mainly used for classification and pattern recognition tasks. Before being usable, they require a training (or learning) stage. Generally, one is in possession of a data training set of known input-output pairs. Learning takes place by presenting the training data inputs to the network, and changing its parameters until the desired output is obtained. Multilayer perceptrons (MLP) are a kind of neural networks. They are adaptive nonlinear systems that receive input signals and transform them into adequate output signals. In general, MLPs are made of an input layer, one or several hidden layers, and an output layer. Information is processed forwardly and generally each unit receives inputs from all the neurons of its preceding layer, performs some non-linear transformation on the weighted sum of these inputs, and transmits its own output to all the neurons of its next layer. Each input received by a neuron is weighted by a parameter called the weight of the connection. Learning takes place by modifying the connection weights.

In section I we investigate the dynamics of learning in MLPs near special singular points of the parameter space. In section II we discuss potential applications of MLPs in coding theory (lossy compression and error correcting code) and show that despite promising theoretical results, conventional methods fail to provide satisfying performance practically.

I. LEARNING AND SINGULARITIES IN MULTILAYER PERCEPTRONS

A set of multilayer perceptrons can be viewed as a parameter space which forms a geometrical manifold, called neuro-manifold in the case of neural networks. It has already been studied [1] that singularities appear in such a manifold and that they strongly affect the dynamics of learning. Moreover, such a neuro-manifold does not have a Euclidian geometrical structure, but a Riemannian geometrical structure. Consequently, the standard gradient is not the steepest descent anymore. Saad and Solla [2] have shown that, while the learning processes, the standard gradient learning (the backpropagation learning) is usually trapped in the plateau (for a long period of time, the training error remains constant until suddenly it starts to decrease again. This phenomenon is illustrated in figure 1 (left) by the red curve). Amari et al. [1] have shown that such a plateau is due to the attraction of singularities and that the natural gradient learning which takes into account the geometrical structure of the neuro-manifold does not seem to be influenced by these singularities. In this study, we extend the theoretical framework developed in Amari et al. [1] to investigate the dynamics of learning near the singularity which is caused by the symmetry of hidden units. We define a student MLP as (see figure 1, middle left),

$$y = f(\boldsymbol{x}, \boldsymbol{\theta}) + \varepsilon, \quad \text{where} \quad f(\boldsymbol{x}, \boldsymbol{\theta}) = w_1 \varphi(\boldsymbol{J}_1 \cdot \boldsymbol{x}) + w_2 \varphi(\boldsymbol{J}_2 \cdot \boldsymbol{x}), \quad \text{and where} \quad \varphi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt. \tag{1}$$

φ denotes the activation function and ε is a Gaussian white noise. The input $\boldsymbol{x}$ is a n-dimensional Gaussian vector $\boldsymbol{x} \sim N(0, I)$. In this case the parameters are given by $\boldsymbol{\theta} = (\boldsymbol{J}_1, \boldsymbol{J}_2, w_1, w_2)$. Next, we define a teacher perceptron,

$$y = f_0(\boldsymbol{x}) + \varepsilon, \quad f_0(\boldsymbol{x}) = w_0 \varphi(\boldsymbol{T} \cdot \boldsymbol{x}). \tag{2}$$

The student perceptron learns the behavior of the teacher perceptron, based on the training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_M, y_M)$, where M is the number of training data. In this study we investigate the standard gradient learning (SGD), and the natural gradient learning (NGD) algorithm [1]. They are given by

$$\dot{\boldsymbol{\theta}}_{\mathrm{SGD}} = -\eta \left\langle l(\boldsymbol{x}, y, \boldsymbol{\theta}) \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle \text{ (SGD)}, \qquad \dot{\boldsymbol{\theta}}_{\mathrm{NGD}} = -\eta \boldsymbol{G}^{-1}(\boldsymbol{\theta}) \left\langle l(\boldsymbol{x}, y, \boldsymbol{\theta}) \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle \text{ (NGD)}, \tag{3}$$

where $l(\boldsymbol{x}, y, \boldsymbol{\theta}) = \frac{1}{2}(y - f(\boldsymbol{x}, \boldsymbol{\theta}))^2$, and where y is the teacher's output. $\langle \cdots \rangle$ represents the expectation with respect to the teacher's signal. $\boldsymbol{G}(\boldsymbol{\theta}) = [g_{ij}(\boldsymbol{\theta})] = \left[E\left(\frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \theta_i} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \theta_j} \right) \right]$ denotes the Fisher information matrix. E is the expectation with respect to the student's signal. η is a small constant so that the dynamics move along the gradient direction. However, singular regions exist in the parameter space of the student MLP [1]. They are given by,

$$C_1 = \{\boldsymbol{\theta} | \boldsymbol{J}_1 = \boldsymbol{J}_2\}, \quad C_2 = \{\boldsymbol{\theta} | w_1 w_2 = 0\}. \tag{4}$$

C_1 corresponds to an overlapping singularity (two neurons can be summed up in one neuron) while C_2 corresponds to an eliminating singularity (one neuron becomes useless). The Fisher information matrix is singular on these regions. Singularities correspond to the deletion of one neuron. We investigate singularities of type C_1. Note that the teacher is voluntarily written using a single neuron only, thus it lies on the singular region of the student parameter space.

We analytically obtained the trajectories of the dynamics of learning by standard gradient descent (SGD) and natural gradient descent (NGD) when the teacher function lies on the singularity [3]. We plot the dynamic vector fields of the SGD and the NGD in figure 1. The SGD dynamics is slowly attracted by C_1 explaining the plateau phenomenon. On the NGD side however, C_1 is not an equilibrium for the dynamics so that the region C_1 is no more critical.

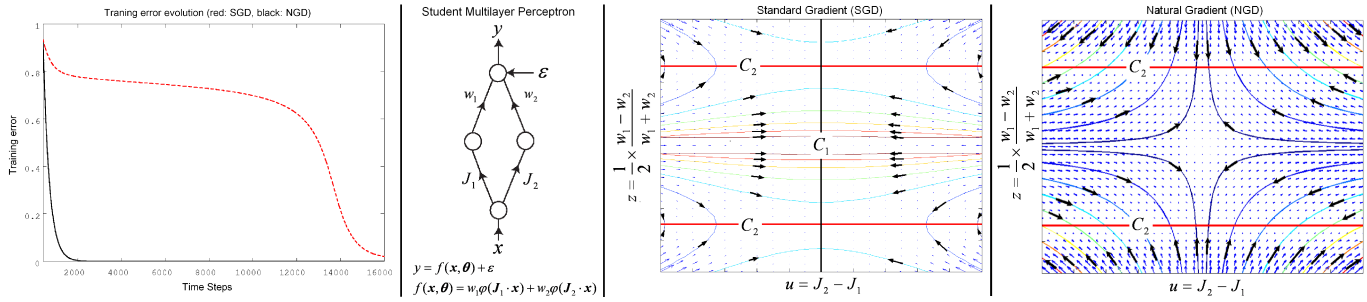

FIG. 1: **Left:** The plateau phenomenon. The red curve shows the SGD dynamics trapped in the plateau. The black curve shows the NGD dynamics unaffected. **Middle left:** Student model network. **Middle right:** Dynamic vector fields with trajectories when the teacher is located on the singularity (SGD). **Right:** Dynamic vector fields with trajectories when the teacher is located on the singularity (NGD).

Finally, we have also investigated the dynamics of learning of the SGD when the teacher is outside the singularity [4]. We found that in this case, the singular region C_1 turns into a Milnor-like attractor composed of some stable and unstable parts. The case of the NGD when the teacher is outside the singularity remains to be investigated.

II. APPLICATIONS OF MULTILAYER PERCEPTRONS IN CODING THEORY

Based on the previous work of Hosaka et al. [5], we investigate a lossy compression scheme using a tree like MLP as decoder. In the same way, based on the previous work of Shinzato et al. [6], we investigate an error correcting code scheme using tree like MLP encoder. We evaluate the typical performance of these schemes at the infinite codeword length limit by using the *Replica Method* of statistical mechanics, and investigate their practical implementations.

A. Lossy Compression

An original Ising message $\boldsymbol{y}$ of size M (i.e.: $\boldsymbol{y} = (y^1, \ldots, y^M)$ where $y^\mu \in \{-1, 1\}$) which is uniformly and independently distributed with a bias p (i.e.: $P[y = 1] = 1 - P[y = -1] = p$) is encoded into an Ising codeword $\boldsymbol{s}$ of size $N < M$ by some non-linear transformation. The code rate of such a scheme is defined by $R = N/M$. The decoded message $\hat{\boldsymbol{y}}$ of size M (i.e.: $\hat{\boldsymbol{y}} = (\hat{y}^1, \ldots, \hat{y}^M)$ where $\hat{y}^\mu \in \{-1, 1\}$) is given by some decoder. Here, we utilize three different tree-like committee machine and tree-like parity machine featuring K hidden units and one output unit as decoder of our scheme. Each network is based the same general architecture and makes use of a non monotonic transfer function f_k, as shown in figure 2. The codeword $\boldsymbol{s}$ is split down into K Ising vectors which are N/K-dimensional, (i.e.:

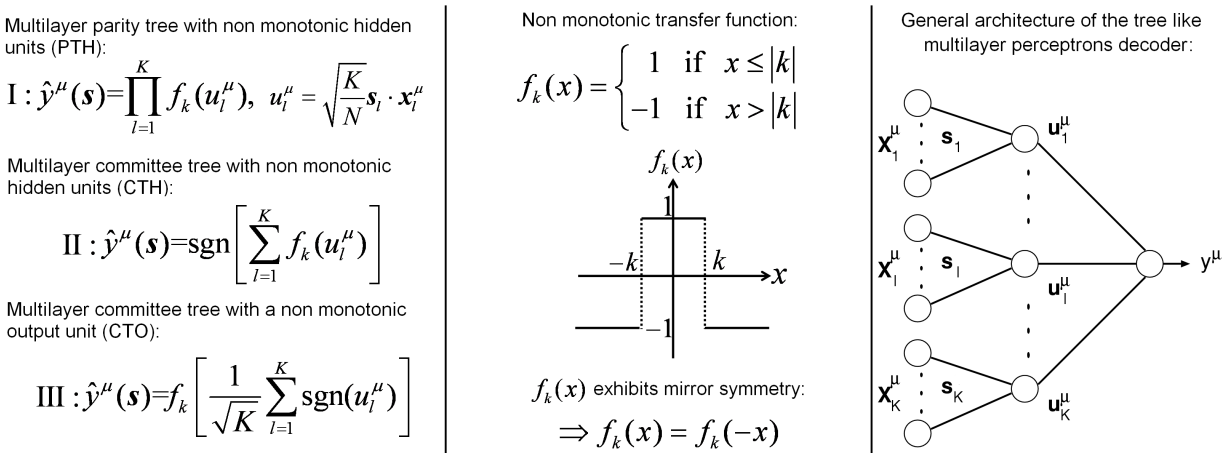

FIG. 2: **Left:** General equation of the three different decoder networks. **Middle:** The non monotonic transfer function f_k. **Right:** General architecture of the three decoder networks.

$\boldsymbol{s} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K)$ where $\boldsymbol{s}_l \in \{-1, 1\}^{N/K}$). The vectors $\boldsymbol{x}_l^{\mu}$ are fixed N/K-dimensional independent vectors uniformly distributed on $\{-1, 1\}$ (i.e.: quenched random variables). The sgn function denotes the sign function taking 1 for $x \geq 0$ and -1 for $x < 0$. Since we investigate the lossy compression case, an amount of distortion D is allowed into the decoded message. This distortion is defined by $D = E[\frac{1}{M}\sum_{\mu=1}^{M} d(y^{\mu}, \hat{y}^{\mu})]$, where E denotes the expectation and d is the Hamming distance ($d(x, y)$ takes the value 0 if $x = y$ and 1 else). In other words, D represents the average error per bit tolerated in the decoded message $\hat{\boldsymbol{y}}$. For a given distortion D, when M and N goes to infinity, the so-called Shannon bound gives the best achievable code rate R a lossy compression scheme can ever achieved. We define the encoder as $\mathcal{F}(\boldsymbol{y}) \equiv \underset{\hat{\boldsymbol{s}}}{\mathrm{argmin}}\, d(\boldsymbol{y}, \hat{\boldsymbol{y}}(\hat{\boldsymbol{s}}))$. Using the *Replica Method*, we were able to show that all the schemes can saturate the Shannon bound under some specific conditions [7].

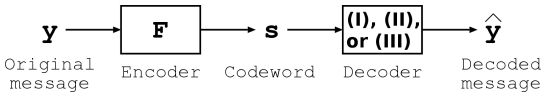

FIG. 3: Layout of the scheme

B. Error Correcting Code

An original Ising message $\boldsymbol{s}^0$ of size N which is uniformly and independently distributed (i.e.: $P[s^0 = 1] = P[s^0 = -1] = 1/2$) is encoded into an Ising codeword $\boldsymbol{y}_0$ of size $M > N$ using the same tree like MLP as introduced in the lossy compression section (i.e.: we use equation (I), (II) or (III) to construct $\boldsymbol{y}_0$. Note that in this case $\boldsymbol{s}$ denotes $\boldsymbol{s}^0$ and $\hat{y}^{\mu}$ denotes y_0^{μ}). The codeword $\boldsymbol{y}_0$ is then fed into a binary asymmetric channel (BAC) where each bit is flipped independently of the others with asymmetric probabilities according to the noise parameters (p, r). The BAC details are shown in figure 4. The corrupted codeword $\boldsymbol{y}$ is finally received at the ouptut of the channel. The code rate of such a scheme is defined by $R = N/M$. The decoded message $\boldsymbol{s}$ of size N is then inferred from $\boldsymbol{y}$ by using some non-linear transformation $\mathcal{G}(\boldsymbol{y}) \equiv \underset{\hat{\boldsymbol{s}}}{\mathrm{argmax}}\, p(\hat{\boldsymbol{s}}|\boldsymbol{y}; \{\boldsymbol{x}\})$. For given parameters (p, r), when M and N goes to infinity the so-called Shannon bound gives the best achievable code rate R an error correcting code scheme can ever achieved. Using the *Replica Method*, we were able to show that all the schemes can saturate the Shannon bound under some specific conditions [8].

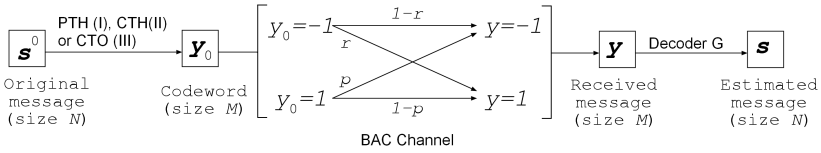

FIG. 4: Layout of the scheme

C. Belief Propagation Algorithm

We have shown that the above schemes can yield optimal performance for both lossy compression and error correcting code tasks. However, in the lossy compression case, we lack a practical encoder (a formal encoder require an amount

of time which grows exponentially with the size of the messages). Respectively for the same reasons, in the error correcting code case we lack a practical decoder.

Thus we need to use some approximation in order to make the encoding/decoding task a tractable one. In this section, we propose to apply the popular belief propagation (BP) algorithm which can be used to compute an approximation of the marginal posterior probabilities and therefore deduce a potential codeword/message in a tractable time. Hosaka et al. used a similar approach for the simple perceptron case [9]. An optimal codeword/message is given by the ground state of the Boltzmann distribution $P(\boldsymbol{s}|\boldsymbol{y},\{\boldsymbol{x}\};\beta) = \exp[-\beta\mathcal{H}(\boldsymbol{s},\boldsymbol{y},\{\boldsymbol{x}\})]/Z$ where $\mathcal{H}$ and Z are the relevant Hamiltonian and partition function respectively. The BP algorithm is used here to compute an approximation of the marginal probabilities $P(s_i|\boldsymbol{y},\{\boldsymbol{x}\};\beta) = \sum_{j\neq i} P(s_j|\boldsymbol{y},\{\boldsymbol{x}\};\beta)$ in a tractable time, assuming that the Boltzmann factor of the previous distribution is factorizable.

Using this assumption, we performed simulations using the BP for both lossy compression and error correcting code case. The results we found are not optimal and far from the theoretical ones. In fact, we found that the BP performance gets poorer as the number of hidden units K increases. Figure 5 shows some results obtained for the CTH case. This results show that using a large number of hidden units perturb the BP dynamics and that the number

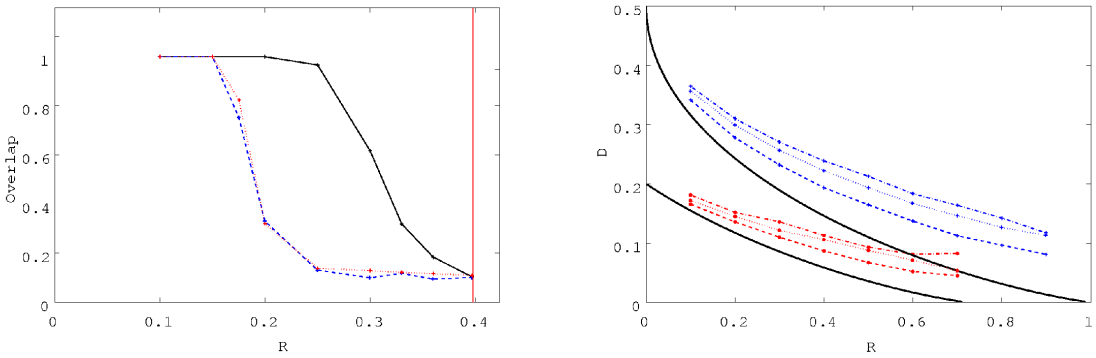

FIG. 5: **Left:** Error correcting code case using the CTH with $K = 1$ (solid), $K = 3$ (dashed) and $K = 5$ (dotted) hidden units. We set $p = 0.1$, $r = 0.2$. We used $N = 1000$. The vertical line represents the Shannon bound. Overlap denotes the quantity $(1/N)\boldsymbol{s}\cdot\boldsymbol{s}^0$ which measure how close the estimated message $\boldsymbol{s}$ is from the original one $\boldsymbol{s}^0$. Overlap of 1 means perfect decoding. **Right** : Lossy compression case using the CTH with $K = 1$ (dashed), $K = 3$ (dotted) and $K = 5$ (dash-dotted). We used $p = 0.5$ (top), and $p = 0.8$ (bottom). We choose $N = 1000$. Solid lines denote the Shannon bound.

of local minima increases with K. It is likely that in the codeword space, singular regions arise and perturb the BP dynamics is some similar way as in section I. The BP dynamics remains to be investigated.

Conclusion

The present thesis investigates the fundamental intrinsic properties of multilayer perceptron neural networks. Since their first discovery, MLPs have shown great potential. From a theoretical point of view, MLPs are universal approximators, capable of coding any non-linear function (provided a sufficient number of hidden units K). Thus, they can solve any pattern recognition task, or yield Shannon optimal performance when used in coding theory.

However practically, the results obtained using conventional methods like the SGD learning in section I or the BP algorithm in section II shows that several problems arise. All these results show that while multilayer perceptrons are very powerful tools theoretically, practical implementation of such systems remain difficult. We need better heuristics to efficiently use such devices. Conventional methods fail to provide satisfying results. Methods especially designed to take care of the particular geometrical structure induced by MLPs appear to be compulsory. On the learning theory side, the natural gradient is an example of such method and works efficiently without being subject to the plateau phenomenon. In the case of coding theory, such an algorithm is still to be found.

[1] S. Amari, H. Park, and T. Ozeki, *Neural Computation*, **18**, 1007-1065, 2006.
[2] D. Saad and S.A. Solla, *Phys. Rev.E*, **52**, 4225-4243, 1995.
[3] F. Cousseau, T. Ozeki, and S. Amari, *IEEE Trans. on Neural Networks*, **19**, 1313-1328, 2008.
[4] H. Wei, J. Zhang, F. Cousseau, T. Ozeki and S. Amari, *Neural Computation*, **20**, 813-843, 2008
[5] T. Hosaka, Y. Kabashima and H. Nishimori, *Phys. Rev. E*, **66**, 066126, 2002.
[6] T. Shinzato and Y. Kabashima, *proceedings of IBIS2006* (in Japanese), Osaka, Japan, 2006.
[7] F. Cousseau, K. Mimura, T. Omori and M. Okada, *Phys. Rev. E*, **78**, 021124, 2008.
[8] F. Cousseau, K. Mimura, and M. Okada, *in press.*
[9] T. Hosaka, Y. Kabashima, *Physica A*, **365**, 113, 2006.