Duality between learning machines: a bridge between supervised and unsupervised learning

J.-P. Nadal
Laboratoire de Physique Statistique[*]
Ecole Normale Supérieure,
24, rue Lhomond, F-75231 Paris Cedex 05, France.

N. Parga
Departamento de Física Teórica,
Universidad Autónoma de Madrid,
Canto Blanco, 28049 Madrid, Spain.

Abstract

We exhibit a duality between two perceptrons which allows us to compare the theoretical analysis of supervised and unsupervised learning tasks. The first perceptron has one output and is asked to learn a classification of p patterns. The second (dual) perceptron has p outputs and is asked to transmit as much information as possible on a distribution of inputs. We show in particular that the maximum information that can be stored in the couplings for the supervised learning task is equal to the maximum information that can be transmitted by the dual perceptron.

This paper has been published in Neural Computation Vol. 6, Issue 3 (May 1994, pages 491-508), published by The MIT Press.

[*] Laboratoire associé au C.N.R.S. (U.R.A. 1306), à l'E.N.S. et aux Universités Paris VI et Paris VII.

1 Introduction

Supervised and unsupervised learning are the two main research themes in the study of formal neural networks. In the first case, one is given a set of input-output pairs which have to be learned by a neural network (usually of a given architecture). One may be interested in the performance of the network as an associative memory, or one may be interested in the ability of the network to generalize: a rule is assumed to be hidden behind the examples (the input-output pairs to be learned), and one asks whether the net will give a correct output for a new input. In the case of an associative memory, the emphasis is usually put on the fact that the memory is *distributed*: the memory is distributed among the synapses, but also the ouput patterns (or attractors for an auto-associative memory) are made of features distributed among the neurons (the best studied case is the one of random patterns)[1] [2]. It is generally considered that such encoding should facilitate associative recall with a high noise tolerance.

In the second case, no desired output is given, and one is asking the network to classify the data (input patterns). Typically one would like two patterns to be put in the same class if they are nearby in input space. Such a constraint is either implicit in the heuristic chosen for modifying the couplings, or explicit in the choice of a cost function. One of the most famous algorithms is the Kohonen maps algorithm [3], where a topology is introduced in the output space. In some approaches one puts the emphasis on *discriminating* between patterns rather than on clustering. For example, it has been shown that unsupervised Hebbian learning with a single linear output neuron leads to a principal component analysis [4]. For a Gaussian input distribution this is equivalent to maximizing the amount of information that the output gives on the input. In fact, a particular strategy is to define a cost function based on information theoretic criteria [5] [6] [7] [8], the justification being general considerations of what type of neural representations (or "codes") of the environment should be useful for the brain.

Unsupervised learning often leads to "grand-mother" type cells: each neuron tends to respond specifically to a given type of stimuli, or one particular feature. For exemple, with some unsupervised algorithms based on Hebbian learning [4] each output unit become specific to one principal component; in clustering algorithms one gets cluster specific cells. One is thus confronted by two completely opposite approaches, differing not only in the type of issues that they address but also in the type of neural codes that they use or construct. What we propose in this paper is a framework which might allow a better understanding of the differences between a supervised and an unsupervised learning task. We will show that one can establish a relationship between the questions which are relevant for each task. This will be done *via* a duality between two neural architectures. Moreover this duality is interesting in itself: in the context of supervised learning, the Bayesian approach tells one how to derive the parameters from the data by relating the probability of the parameters (the model) knowing the data to the probability of the data knowing the parameters. The duality that we introduce is nothing but an explicit implementation of this exchange between model and data.

The paper is organized as follows. In section 2 we present the duality between two perceptrons, and show how this allows us to relate the study of a supervised learning task to that of an unsupervised learning task. In particular we show the identity between various *capacities* which have been defined in each context. In section 3 we put emphasis on the differences between the two tasks, showing however the deep relationship between the two problems. We show in particular how the statistical mechanics approach to learning is related to the study of the quantity of information that is relevant in the context of unsupervised learning. We show also that the first perceptron can be thought of as a *decoder* if one considers the second one as a neural *encoder*. Perspectives are given in the Conclusion, and a generalization to other learning machines (other than the simple perceptron) is given in the Appendix.

2 From Supervised To Unsupervised Learning

2.1 The Dual Perceptrons

Let us consider a simple perceptron, with one binary output (whose state σ takes, say, the values 0 or 1), N inputs neurons and couplings $\vec{J} = \{J_1, ..., J_N\}$. We consider continuous inputs unless otherwise specified. In a supervised learning task, one is given a set Ξ of p input patterns,

$$\Xi = \{\vec{\xi}^\mu, \mu = 1, ..., p\} \tag{1}$$

and the set of the desired outputs,

$$\vec{\tau} = (\tau^\mu = 0, 1, \ \mu = 1, ..., p)$$

which have to be learned by the perceptron. For a given choice of the couplings, the output σ^μ when the μth pattern is presented is given by:

$$\sigma^\mu = A(\vec{J}, \vec{\xi^\mu}) \equiv \Theta(\sum_{j=1}^{N} J_j \xi_j^\mu) \tag{2}$$

where $\Theta(h)$ is 1 for $h > 0$ and 0 otherwise. For simplicity we assume zero threshold, and we will consider only the above deterministic rule (no synaptic noise).

Now one can interpret this formula (2) in two ways. One is, as above, that we have p input-ouput pairs realized by a perceptron with a single output unit, whose couplings are the J's.

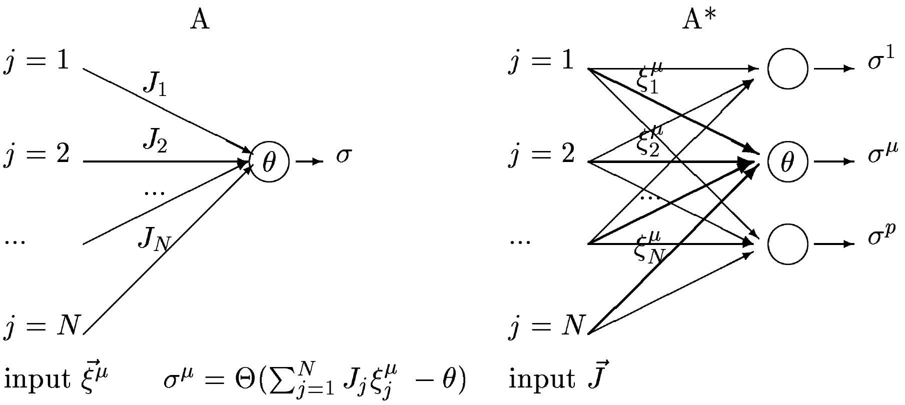

Figure 1: The dual perceptrons

But one can as well say that we have a perceptron with p output units, where $\vec{J}$ is now an input pattern, and the $\vec{\xi}^{\mu}, \mu = 1, ..., p$ are the p coupling vectors (figure [1]). Let us call our initial perceptron with a unique output $\mathcal{A}$, and the dual perceptron, with p output units as just explained, $\mathcal{A}^*$. In the following we show how useful this duality can be, in particular for the comparison between supervised and unsupervised learning. To avoid confusions when considering one of the dual perceptrons, we will append a "*" to each ambiguous word whenever we are considering $\mathcal{A}^*$: in particular we will write "pattern*" and "couplings*", the * being a reminder that for $\mathcal{A}^*$ these denominations refer to $\vec{J}$ and to the $\vec{\xi}^{\mu}$, respectively.

2.2 The number of domains

Let us recall some important results concerning the supervised learning task for $\mathcal{A}$. Of particular interest for what follows is the geometrical approach [9] to the computation of the maximal storage capacity: one considers the space of couplings ($\vec{J} = \{J_j, j = 1, ..., N\}$ being considered as a point in an N dimensional space). Then each pattern μ defines a hyperplane, and the output σ^{μ} is 1 or 0 depending on which side of the hyperplane the point $\vec{J}$ lies. Hence the p hyperplanes divide the space of couplings into domains (figure [2]), each domain being associated with one specific set $\vec{\sigma} = \{\sigma^1, ..., \sigma^p\}$ of outputs. Let us call $\Delta(\Xi)$ the number of domains

$$\Delta(\Xi) = \textit{number of domains} \tag{3}$$

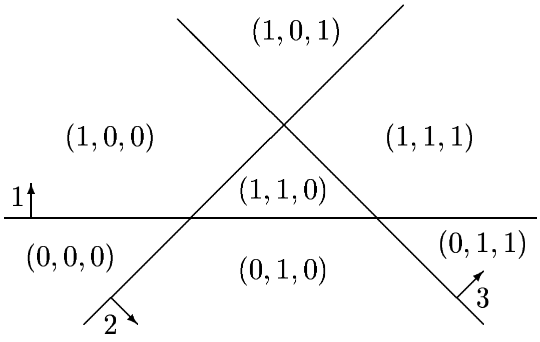

Figure 2: Partition of $\vec{J}$ space in domains

Since each σ^{μ} is either 0 or 1, there are at most 2^p different output configurations $\vec{\sigma}$, that is

$$\Delta(\Xi) \leq 2^p \tag{4}$$

If the patterns are "in a general position", then $\Delta(\Xi)$ is in fact independent of Ξ and a function only of p and N. One has the basic result [9]:

$$\Delta(\Xi) = \Delta(N, p) \equiv \sum_{k=0}^{\min N,p} C_p^k \tag{5}$$

where $C_p^k = \frac{p!}{k!\,(p-k)!}$. In particular

$$\Delta(N,p) = \begin{cases} 2^p & \text{if } p \leq N \\ < 2^p & \text{if } p > N \end{cases} \tag{6}$$

This means that N is the "Vapnik-Chervonenkis dimension" [10] [11] of the perceptron (that is $N+1$ is the first value of p for which Δ is smaller than 2^p):

$$d_{VC} = N \tag{7}$$

If the task is to learn a rule from examples, the VC dimension plays a crucial role: generalization will occur if the number of examples p is large compared to d_{VC} [10]. Another important parameter is the asymptotic capacity. In the large N limit, for a fixed ratio

$$\alpha \equiv \frac{p}{N} \tag{8}$$

the fraction of output configurations which are not realized remains vanishingly small for α greater than 1, up to the "critical storage capacity" ([9], [12]) α_c,

$$\alpha_c = 2. \tag{9}$$

2.3 The number of domains: the dual point of view

Now let us reconsider the geometrical argument from the point of view of the dual perceptron $\mathcal{A}^*$ as defined in 2.1. What we have just said is that, for a given choice of the couplings*, Ξ, one explores all the possible different output states $\vec{\sigma}$ that can be obtained when the input pattern* $\vec{J}$ varies. If $\vec{J}$ represents, say, the light intensities on a retina, $\vec{\sigma}$ is the first neural representation of a visual scene in the visual pathway. Since all visual scenes falling into a same domain are encoded with the same neural representation, $\Delta(\Xi)$ is the maximal number of visual scenes that can be distinguished. This can be said in term of coding of information: to specify one domain out of $\Delta(\Xi)$ represents $\ln\Delta(\Xi)$ bits of information. Hence the maximum amount of information, or "information capacity" C, that $\vec{\sigma}$ can convey on the inputs* is

$$C(\Xi) = \ln\Delta(\Xi). \tag{10}$$

In what sense is $C(\Xi)$ the *maximal* amount of information that can be gained? Let us consider again the retina analogy. Each visual scene is a vector in a N dimensional space, but not every vector of that space may correspond to a possible visual scene. Hence some of the domains might be empty (no stimulus ever falls inside these domains), so that some of the output codes may not be used. More generally, the statistics of visual scenes will typically be such that the input* domains are not visited with equal frequency. The amount of information I actually transmitted is thus smaller (and at best equal to) C (we will come back to the study of I in section 3).

In the language of information theory, C is the channel capacity of the perceptron $\mathcal{A}^*$ if used as a memoryless channel in a communication system [13]. In that case the input alphabet is the set of all possible J's, and the output alphabet the 2^p possible output configurations.

If the Ξ are in general position, the capacity is only a function of p and N, and is given by (5) - note that otherwise the capacity is smaller than (or equal to) $\ln\Delta(N,p)$. From (5) one sees that up to $p = N$ each output neuron gives one bit of information ($C = p$), and for

$p > N$ one gains less and less information by adding new units*. More precisely, one has the asymptotic behavior

$$\lim_{N\to\infty} C/N \equiv c(\alpha) = \begin{cases} \alpha & \text{if } \alpha \leq 2 \equiv \alpha_c \\ \alpha S(1/\alpha) & \text{if } \alpha > 2 \end{cases} \tag{11}$$

Here (and throughout this paper) logarithms are expressed in base 2, and $S(x)$ is the entropy function (measured in bits):

$$S(x) = -[x \ln x + (1-x)\ln(1-x)]. \tag{12}$$

The information capacity $c(\alpha)$ is shown on figure [3].

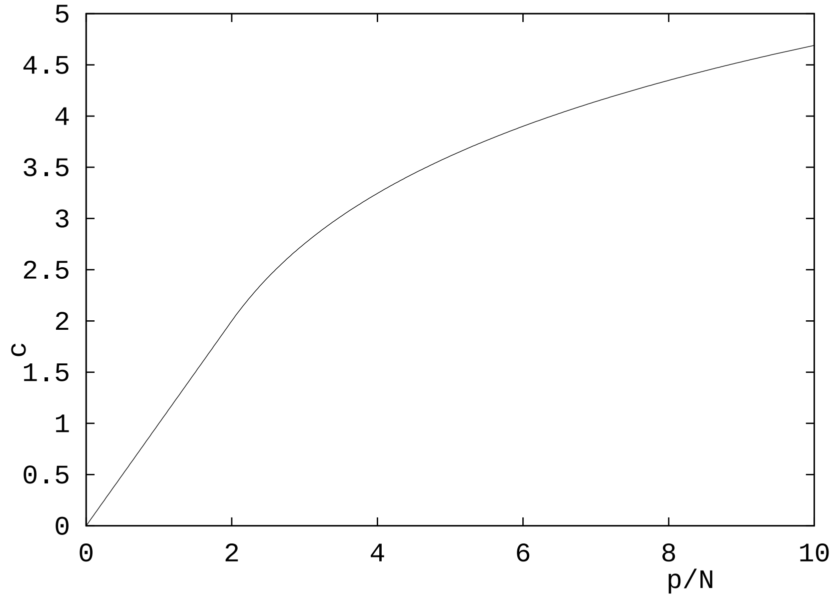

Figure 3: The asymptotic information capacity/content c of the perceptron $\mathcal{A}^*/\mathcal{A}$ (in bits per input* neuron/coupling) as a function of $\alpha = p/N$.

We are thus led to consider the dual perceptron as what we will call a "neural encoder", a device which associates a neural representation (or codeword) with each input* signal, for which the performance is evaluated with tools coming from information theory. This point of view corresponds to an approach developed recently in particular for modeling the sensory pathways in the brain ([7] [8]). In that context one wants the system to perform an efficient coding, according to some cost function derived from information theory concepts and general considerations on what type of coding might be useful for the brain [5] [6]. The algorithmic counterpart, that is the modification of the couplings* in order to minimize such a cost function, results in unsupervised learning schemes: the cost function specifies an average quality of the code, but not a desired output for a given input* (we will come back to this later on). The duality between the two perceptrons is thus a bridge between the study of supervised and unsupervised learning tasks.

2.4 The information content

We have seen that $\ln \Delta$ ($c(\alpha)$ in the large N limit) is an information quantity relevant for $\mathcal{A}^*$. What is its meaning for $\mathcal{A}$? Since it is the number of bits needed for specifying one domain out of Δ, it is the amount of information stored in the couplings when learning an association $(\Xi, \vec{\tau})$ whenever this particular configuration $\vec{\tau}$ corresponds to an existing domain. This gives the obvious result that below α_c the amount of information stored (in bits per synapse) is equal to α. But for $\alpha > \alpha_c$ with probability one (in the large N limit) no domain exists for a configuration $\vec{\tau}$ chosen at random, and errors will result. However, it has been shown by G. Toulouse [14] that even above α_c, $c(\alpha)$, as given by (11), remains the maximal amount of information that can be stored in the synapse. Hence we can use the term "information capacity" with its dual meaning of information content or of capacity for transmitting information. The rest of this paper will detail the comparison between the study of $\mathcal{A}$ for a supervised learning task and of $\mathcal{A}^*$ as a neural encoder (as defined above).

3 Statistical Mechanics and the Mutual Information

Although the information capacity of the perceptron* is equal to the information storage capacity of the perceptron as an associative memory, there are important differences between the analysis of the two tasks. To see this, we have to be more specific about the relevant questions for each perceptron.

3.1 Supervised Learning

We start with the supervised learning task for $\mathcal{A}$. The statistical physics (or Bayesian) approach to supervised learning ([12], [15], [16]) forces us to study a *statistical ensemble* of machines, the couplings being taken from some *prior* distribution $\rho(\vec{J})$. For example, if one looks for discrete couplings, $\rho(\vec{J})$ may give equal weight to every possible choice of couplings. Another example, the best studied case, is the one of spherical couplings:

$$\sum_{j=1}^{N} J_j^2 = N \tag{13}$$

with $\rho(\vec{J})$ being the uniform measure on the sphere. One is interested in the probability that a *given* set of outputs $\vec{\sigma}$, chosen at random, is realizable. According to the deterministic rule (2), the probability for having $\vec{\sigma}$ has the expression

$$P_\sigma = \int d\vec{J} \rho(\vec{J}) \prod_{\mu=1}^{p} \Theta(\sigma^\mu \vec{J}.\vec{\xi}^\mu). \tag{14}$$

In other words P_σ is the fractional volume of the couplings which implement the particular set of associations $(\Xi, \vec{\sigma})$. After $\Delta(\Xi)$, P_σ is the most important quantity relevant to our discussion. The typical probability that a random $\vec{\sigma}$ is learnable has been computed [12] for patterns drawn from a statistical ensemble. In principle one has to compute P_σ for a given choice of the patterns. However, in the large N limit the log-probability L,

$$L(\sigma) \equiv \ln P_\sigma \tag{15}$$

is "self-averaging": the limit $l \equiv L/N$ when N goes to infinity exists and is only a function of the distribution ρ^*. It is then also given by the limit of the averaged value of L:

$$l \equiv lim_{N\to\infty} L/N = lim_{N\to\infty} << \ln P_\sigma >> /N \tag{16}$$

where $<< . >>$ means the average over the patterns:

$$<< f >>= \int \prod_{\mu=1}^{p} d\vec{\xi}^\mu \; \rho^*(\Xi) f(\Xi) \tag{17}$$

Statistical mechanics tools such as the "replica technique" or the "cavity method" [17] have made possible the computation of l for various choices of the patterns distributions (uncorrelated, with and without bias, and very recently correlated patterns[18]), and of the space of couplings (continuous or discrete). For each case one gets in particular the critical asymptotic capacity α_c.

3.2 Unsupervised Learning

We turn now to the dual perceptron. Having specified the distribution $\rho(\vec{J})$ for $\mathcal{A}$, we have thus to consider that at each instant the dual perceptron receives a new input* $\vec{J}$, a particular pattern* $\vec{J}$ occurring with probabilty $\rho(\vec{J})$. To be more concrete we will talk of $\vec{J}$ as an "image", for which the neural encoder has to give a neural representation, or code, $\vec{\sigma}$. As much as possible different images should have different neural representations: as already mentioned, one is interested in having the largest variety of available outputs. This variety is measured by the entropy of the output:

$$H(P_\sigma) = -\sum_{\vec{\sigma}} P_\sigma \ln P_\sigma. \tag{18}$$

Since we are considering a deterministic system, this entropy is equal to the *mutual information* $I(\sigma, J)$ between the input* and the output:

$$I(\sigma, J) = H(P_\sigma) \tag{19}$$

Indeed, when a configuration $\vec{\sigma}$ is observed, the gain of information is equal to $-\ln P_\sigma = -L(\sigma)$, and $I(\sigma, J) = H(P_\sigma)$ is the average gain of information. The mutual information is the main quantity of interest for the study of $\mathcal{A}^*$. Its study, that is of the entropy (18), is to be contrasted with that of L for a randomly chosen $\vec{\sigma}$. Note that I is a function of the distribution ρ and of the couplings* Ξ: $I = I(\rho, \Xi)$. What is its relationship with the capacity C ? If ρ gives the same weight to every domain (in $\vec{J}$ space), then for any $\vec{\sigma}$ which corresponds to a domain, $P_\sigma = \frac{1}{\Delta(\Xi)}$, and the entropy is at its maximal possible value, $\ln \Delta(\Xi)$. Hence one has

$$I(\sigma, J) \leq C = \max_\rho I(\sigma, J) \tag{20}$$

and $C - I(\sigma, J)$ is the *redundancy* of the code $\vec{\sigma}$. Now for a given distribution ρ, one would like to optimize the performance of the network by a proper choice of the parameters* Ξ. A simple idea is that the network should extract as much information as possible from the environment, which means maximizing the mutual information:

$$\text{Optimization principle: find } \Xi^* \text{ which realizes } \max_\Xi I(\sigma, J) \tag{21}$$

This strategy has been used by Linsker [7] for modeling the first stages in the visual pathway, and is related to other strategies such as the minimization of redundancy ([8], [5]). In this framework most of the analytical studies have been done for networks with linear neurons. Here we limit our study to the extreme opposite case of binary neurons. We will not detail here the various proposed strategies (for a general discussion see [5], [6], [8], and for the case of the perceptron see [19] [20] [21]), and consider here only the above optimization principle.

What would be the meaning of this principle in the context of $\mathcal{A}$? Given that the couplings would be taken from some prior distribution, the optimal patterns Ξ^* are those which are the easiest to learn in the following sense: in the ideal case, that is if $I = C$, all the learnable set of associations Ξ^*,$\vec{\tau}$ are equiprobable. This might be useful if we are free to choose the patterns as "random" addresses, the perceptron being used as a random access memory for storing p-bit strings (the $\vec{\tau}$). In the context of learning a rule by example, the network is fully unbiased for this particular choice of input patterns, every learnable rule has the same weight. Conversely, is there any *typical* quantity of interest in the vein of (16) ? In fact it is indeed interesting to consider the typical mutual information $\overline{\imath}$ that results if the couplings* are taken from some statistical ensemble $\rho^*(\Xi)$:

$$\overline{\imath} \equiv lim_{N\to\infty} I/N = lim_{N\to\infty} << H(P_\sigma) >> /N \tag{22}$$

where $<< . >>$ denotes the average over the distribution $\rho^*(\Xi)$ as in (17). The motivation is the following. One considers a given input* distribution $\rho(\vec{J})$. First, one would like to know what is the information that is transmitted in the absence of any optimization: this can be obtained by considering couplings* taken at random, with each component being an independent unbiased random variable. This tells us how much can be gained by optimization. Furthermore, instead of trying to find *the* optimal couplings*, one may consider a statistical ensemble of coupling* vectors characterized by the correlations Γ in their components. Then one looks for the correlations which maximize $\overline{\imath}$. We have carried out this program for the perceptron, and we present the details in a separate paper [20] [21]. The main result is that, for Gaussian inputs* with correlation matrix G, the optimal correlation matrix Γ is equal to the inverse of G. This result is very similar to the one obtained for linear units by Linsker, but with two main differences. First as explained above we are computing the optimal statistical properties of the couplings* instead of the exact optimal couplings*. Second we are not considering translational invariant couplings*, that is

$$\xi_j^\mu = \xi(x_\mu - x_j) \tag{23}$$

where x_μ and x_j are the locations of the output* neuron μ and of the input* unit j, as is the case in the works of Linsker and Atick et al. In fact the study of binary units with the restriction (23) is much more difficult (see Bialek and Zee [19] for the study of a binary perceptron* used for a discrimination task under the condition (23)). In particular, one does not know the capacity in that case (all we can say is that it is smaller than the one we computed without restricting the couplings* by (23)). However, in our statistical approach we have a statistical invariance under translation : the correlations within the couplings* do not depend on the index μ.

3.3 Decoding and Learning

When considering the perceptron as an neural encoder, it is natural to ask for the possibility of *decoding*, that is of reconstructing the input image that generated a particular codeword.

This aspect may not be of any biological relevance at all, first because it is likely that the system does not need to perform such a reconstruction, and second because the way we consider the decoding process as explained below has no reason to be biologically plausible. However it is a legitimate question from the information processing point of view. Of course one cannot obtain exactly the input image, since many different inputs give the same output configuration - and indeed we have at our disposal only a finite amount of information about $\vec{J}$ knowing the codeword $\vec{\sigma}$, this information being precisely equal to $-\ln P_{\vec{\sigma}}$. However one can produce one input configuration among those which produce this same codeword: a prototype of all the images considered as identical by the coding system. This can be done by considering precisely the perceptron $\mathcal{A}$ as we explain it now. From the knowledge of the codeword $\vec{\sigma}$ and of the couplings* Ξ, we want to generate a pattern* $\vec{J}$ that satisfies the p equations (2). Hence one can conveniently come back to the first interpretation of these equations, by saying that we are looking for couplings realizing the p associations $(\vec{\xi}^{\mu},\ \sigma^{\mu}),\ i = 1, ..., p$. This problem can be solved algorithmically by the use of perceptron type algorithms [22]. Such algorithms are known to converge whenever a solution exists, which is the case here since the true input pattern is of course one particular solution. One may look for the coupling vector (that is the input pattern*) the most likely to have produced the codeword. This is the standard strategy in the task of learning a rule by example ([15],[16]). Hence, one sees that maximum likelihood decoding for $\mathcal{A}^*$ (with $\rho(\vec{J})$ as input* distribution) is equivalent to Bayesian learning for $\mathcal{A}$ (with $\rho(\vec{J})$ as prior distribution).

3.4 Source Entropy and Storage Resources

The last point of comparison that we will make now bears on another aspect of the efficiency of the perceptrons. In the context of supervised learning, one is interested in comparing the amount of information stored to the number of bits used for storing the couplings. It is thus convenient to assume here a finite (possibly very large) number of bits per synapse, K. There are thus NK bits available, but since the couplings are taken from the prior distribution ρ, the total number of bits that is effectively available may be smaller, being given by

$$I_0 = \sum_{\vec{J}} -\ \rho(\vec{J}) \ln \rho(\vec{J}). \tag{24}$$

Clearly the amount of information I that can be stored in the couplings cannot be larger:

$$I \leq I_0 \tag{25}$$

In the language of the dual perceptron, (25) states that the mutual information cannot be larger than the information content of the source, I_0. How does this relates to the information capacity ? The information capacity that we have computed was for continuous couplings (K infinite). If one limits the number of bits to K, then there are only 2^{NK} different coupling vectors. In the partition of the J space induced by the patterns, some of the domains may be empty. If we call $\Delta_K(\Xi)$ the number of non empty domains, the capacity is now

$$C_K(\Xi) = \ln \Delta_K(\Xi) \leq C \tag{26}$$

and this capacity is clearly bounded above by NK (more generally I_0 is the upper bound of the capacity if restricted to distributions of maximum entropy I_0). Few analytical results are known for the case of discrete couplings, apart from the critical storage capacities α_c for various choices of discrete couplings ([23], [24]). For example, if one takes binary couplings ($J_j = +/-1$) (which corresponds to binary inputs*), the asymptotic information capacity c is equal to α up to $\alpha_c \simeq 0.83$[23] (hence for binary inputs* the capacity is equal to α only up to $\alpha_c \simeq 0.83$).

4 Conclusion

We have shown that the existence of a duality property between two perceptrons allows comparison of the theoretical analysis of supervised and unsupervised learning tasks. The questions that are relevant in one case are intimately related to those relevant for the other perceptron. In particular the information capacity has a nice dual meaning. But there are important differences, expressed mainly in the fact that in a supervised learning task one is interested in the performance of one given choice of outputs, whereas in the unsupervised task it is the average properties over all possible outputs that matter. We have shown also that statistical physics tools can be used for studying the typical properties of the dual perceptron $\mathcal{A}^*$. We present elsewhere[20][21] the detailed analysis.

We have considered the simplest neural architecture. However we stress that the duality can be extended to a general learning machine as shown in details in the Appendix. In the introduction we mentioned that the duality can be viewed as an implementation of the exchange between model and data that appears in the Bayesian approach to learning: this is made explicit in the Appendix. Considering the extension to a general learning machine is useful in particular in order to identify the specific role of the number of couplings, the VC dimension and the number of inputs, which are all equal in the case of the perceptron. The main result is that essentially all that we have said for the perceptron remains valid in the general case, provided one interpretes α as the ratio of p to the VC dimension (instead of the ratio of p to the number of couplings). This result is based on an upper bound for the number of domains which is given by Vapnik in his book[10], and we point out in the Appendix that this bound is optimal. In fact it appears that the perceptron plays a special role: among all learning machines having the same VC dimension, $d_{VC} = d$, the perceptron (with $N = d$) is the one which has the largest information capacity $C = \ln \Delta(\Xi)$ for p larger than d.

Finally we note that we have restricted our study to a deterministic perceptron. We are presently working on noisy systems: most of what we have said remains valid, although noise introduces additional (and interesting) differences between the two types of learning tasks.

Acknowledgements

We would like to thank M. Mézard, R. Monasson, N. Sourlas and G. Toulouse for fruitful discussions. We also thank G. Toulouse for a critical reading of the manuscript. We thank S. Verdu for discussions on information theory. We thank T. Watkin for checking English correctness.

One of us (NP) would like to warmly thank the hospitality received at the Laboratoire de Physique Statistique of the Ecole Normale Supérieure (Paris), and the Laboratorio di Fisica and the INFN of the Istituto Superiore di Sanita (Rome).

This work was partly supported by the programme Cognisciences of C.N.R.S..

A Appendix: Dual Learning Machines

A.1 Statement of the problem

Results obtained for the perceptron may be misleading: the number of couplings, the number of inputs and the VC dimension are all equal. Moreover, the number of domains is independent of the choice of the patterns (if they are in general position or if chosen at random). It is thus useful to consider the case of a general learning machine. We will see that all that is valid for the perceptron remains nearly valid in the general case, provided one identifies the specific role of the various parameters.

Let us thus consider a machine defined by a given architecture $\mathcal{A}$, with a set of N adjustable parameters (couplings) $\vec{J} = \{J_j, j = 1, ..., N\}$. If M is the dimension of the input space, the machine $\mathcal{A}$ associates with each input $\vec{\xi} = \{\xi_i, i = 1, ..., M\}$ a binary output σ (figure [4]). One wants to study the hetero-associative task, where for p input patterns

$$\Xi = \{\vec{\xi}^{\mu}, \mu = 1, ..., p\} \tag{27}$$

one is given the set of the desired outputs,

$$\vec{\tau} = (\tau^{\mu} = 0, 1, \ \mu = 1, ..., p) \tag{28}$$

For a given choice of couplings the outputs are $\vec{\sigma} = \{\sigma^1, ..., \sigma^p\}$ with $\sigma^{\mu} = \mathbf{A}(\vec{J}, \vec{\xi}^{\mu})$. The input patterns are chosen from some distribution ρ^*, and the desired ouputs are chosen at random.

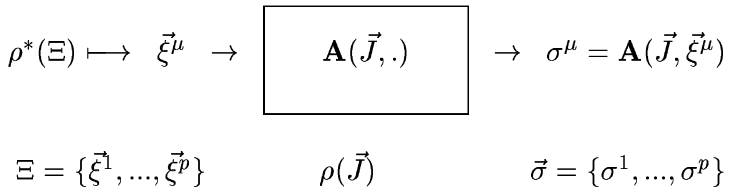
$\rho^*(\Xi) \longmapsto \ \vec{\xi}^{\mu} \ \rightarrow$
$\mathbf{A}(\vec{J}, .)$
$\rightarrow \ \sigma^{\mu} = \mathbf{A}(\vec{J}, \vec{\xi}^{\mu})$
$\Xi = \{\vec{\xi}^1, ..., \vec{\xi}^p\}$
$\rho(\vec{J})$
$\vec{\sigma} = \{\sigma^1, ..., \sigma^p\}$

Figure 4: The learning machine **A**

The main questions that one asks are: What is the storage capacity (the maximal number of input-ouput pairs that can be learned)? What is the maximal amount of information that can be stored in the couplings ? And if the task is to learn a rule by example, what is the probability of making an error on a new pattern as a function of the fraction of errors done on the training set $(\Xi, \vec{\tau})$?

The statistical physics (or Bayesian) approach to learning ([12], [15], [16]) forces us to study a *statistical ensemble* of machines, the couplings being taken from some prior distribution $\rho(\vec{J})$. For example, if one looks for discrete couplings, $\rho(\vec{J})$ may give equal weight to every possible choice of couplings. Having specified the machine $\mathcal{A}$, the distributions ρ^* and ρ, we can now introduce the dual *statistical ensemble* of learning machines as shown on figure [5].

The inputs to $\mathcal{A}^*$ are N dimensional patterns* $\vec{J}$, occurring with probability $\rho(\vec{J})$. We want to study $\mathcal{A}^*$ as a neural encoder, or as a module in a communication system, as briefly explained above. Here the main questions are: for a given distribution $\rho(\vec{J})$, what is the information that the output conveys about the input* (the mutual information between $\vec{\sigma}$

$$\begin{tikzcd}
\rho(\vec{J}) \arrow[r, mapsto] & \vec{J} \arrow[r] & \boxed{\mathbf{A}^*(\Xi,.) = \begin{cases} \mathbf{A}(.,\vec{\xi}^1) \\ \vdots \\ \mathbf{A}(.,\vec{\xi}^p) \end{cases}} \arrow[r] & \vec{\sigma} = \{\sigma^1, ..., \sigma^p\} = \mathbf{A}^*(\Xi, \vec{J}) \\
& & \rho^*(\Xi) &
\end{tikzcd}$$

Figure 5: The dual machine **A***

and $\vec{J}$) ? What is the maximal amount of information that can be conveyed (irrespective of $\rho(\vec{J})$) [or what is the channel capacity of $\mathcal{A}^*$ if used as a channel] ? What is the choice of the parameters* Ξ (or the choice of $\rho^*(\Xi)$) which optimizes the performance of $\mathcal{A}^*$ (according to some cost function to be specified)?

A.2 The number of domains

One can relate the questions listed for the study of $\mathcal{A}$ to those for $\mathcal{A}^*$ exactly as for the case of the perceptron. The first step is made by considering the number of domains $\Delta(\Xi)$. For a general machine, this number will depend on the data Ξ. Note also that one domain, which is the set of points in $\vec{J}$ space associated with one given configuration $\vec{\sigma}$, need not be connected. The information capacity for a given Ξ is thus

$$C(\Xi) = \ln \Delta(\Xi) \tag{29}$$

For a large system, we may expect $C(\Xi)$ to be a self averaging quantity, so that one is interested in the average value

$$\overline{C} \equiv < \ln \Delta(\Xi) > \tag{30}$$

where $< . >$ means the average with respect to $\rho^*(\Xi)$. $\overline{C}$ is thus the typical amount of information that can be stored in the couplings, or the typical information capacity of the neural encoder. In the context of learning a rule by example, it has been shown by Vapnik [10] that generalization is guaranteed (that is the probability of making an error on a new input pattern will tend towards the fraction of errors made on the training set) if

$$lim_{p \to \infty} \frac{\overline{C}}{p} = 0 \tag{31}$$

One may define a "typical" VC dimension d_t as the first value of p for which $\overline{C}$ is smaller than p, and the critical storage capacity as the maximal value of $\alpha_t \equiv p/d_t$ for which $lim_{d_t \to \infty} \frac{\overline{C}}{p} = 1$. This storage capacity should correspond to what is computed by statistical mechanics tools. A sufficient condition for (31) to be true is that the VC dimension is finite. The VC dimension is defined relative to the worst case (or the best case, it depends on the point of view): one considers the maximal value of the number of domains:

$$\Delta_m \equiv \max_{\Xi} \Delta(\Xi) \tag{32}$$

In [10] [11] Δ_m is called the *growth function*. It depends on the number of patterns, p, and on the architecture $\mathcal{A}$. Again Δ_m is at most equal to 2^p, and the VC dimension d_{VC} is equal to the first value of p for which $\Delta_m < 2^p$. In 1968 Vapnik and Chervonenkis [10] [11] showed the remarkable result that when d_{VC} is finite,

$$\Delta_m \leq \Delta(d_{VC}, p) \tag{33}$$

where $\Delta(d_{VC}, p)$ is the number of domains for a perceptron with $N = d_{VC}$ inputs and p patterns in general position, that is (see (5)):

$$\Delta(d_{VC}, p) = \sum_{k=0}^{d_{VC}} C_p^k \quad \text{for } p > d_{VC} \tag{34}$$

where $C_p^k = \frac{p!}{k!\,(p-k)!}$. We have thus an upper bound $C(d_{VC}, p) \equiv \ln \Delta(d_{VC}, p)$ for the information capacity $\overline{C}$. Note that this bound is optimal: the bound is valid for all learning machines having a same value d of the VC dimension, and the bound is saturated for at least one of these machines, the perceptron for which $N = d$.

For a large network, d_{VC} is large. It is then convenient to define α by

$$\alpha \equiv \frac{p}{d_{VC}} \tag{35}$$

and the curve shown on figure [3] appears as a universal curve. One has in the large size limit ($d_{VC} \to \infty$ for a given ratio α):

$$\lim_{d_{VC} \to \infty} \overline{C}/d_{VC} \equiv \overline{c}(\alpha) \;\leq\; \lim_{d_{VC} \to \infty} C(d_{VC}, p)/d_{VC} \equiv c(\alpha) \tag{36}$$

$$c(\alpha) = \begin{cases} \alpha & \text{if } \alpha \leq \alpha_c = 2 \\ \alpha S(1/\alpha) & \text{if } \alpha > 2 \\ \sim \ln \alpha & \text{for } \alpha \text{ large} \end{cases} \tag{37}$$

It is also interesting to note the meaning of C as an information content in the context of learning a rule: generalization occurs when adding a new pattern does not bring much new information. To conclude this section, one sees that one can extrapolate the results obtained for the perceptron to more general learning machines, provided one interpretes α as the ratio of the number of patterns to the VC dimension, and one interprets $C = \ln \Delta(N, p)$ as an upper bound for the information capacity if N is replaced by d_{VC}.

References

[1] Hopfield J.J. Neural networks as physical systems with emergent computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–58, 1982.

[2] Peretto P. *An introduction to the modeling of neural networks.* Cambridge University Press, 1992.

[3] Kohonen T.O. *Self-organization and associative memory.* Springer, Berlin, 1984.

[4] Hertz J., Krogh A., and Palmer R. G. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Cambridge MA, 1990.

[5] Barlow H. B. Possible principles underlying the transformation of sensory messages. In Rosenblith W., editor, *Sensory Communication*, page 217. M.I.T. Press, Cambridge MA, 1961.

[6] Barlow H. B. Unsupervised learning. *Neural Comp.*, 1:295–311, 1989.

[7] Linsker R. Self-organization in a perceptual network. *Computer*, 21:105–17, 1988.

[8] Atick J. J. Could information theory provide an ecological theory of sensory processing. *NETWORK*, 3:213–251, 1992.

[9] Cover T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electron. Comput.*, 14:326, 1965.

[10] Vapnik V. *Estimation of Dependences Based on Empirical Data.* Springer Series in Statistics. Springer, New-York, 1982.

[11] Vapnik V. N. and Chervonenkis A. YA. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[12] Gardner E. The space of interactions in neural networks models. *J. Phys. A: Math. and Gen.*, 21:257, 1988.

[13] Blahut R. E. *Principles and Practice of Information Theory.* Addison-Wesley, Cambridge MA, 1988.

[14] Brunel N., Nadal J.-P., and Toulouse G. Information capacity of a perceptron. *J. Phys. A: Math. and Gen.*, 25:5017–5037, 1992.

[15] Levin E., Tishby N., and Solla S. A. A statistical approach to learning and generalization in layered networks. In *1989 workshop on computational learning theory, COLT'89*, 1990.

[16] Grassberger P. and Nadal J.-P., editors. *From Statistical Physics To Statistical Inference and Back.* Kluwer Acad. Pub., Dordrecht, 1993.

[17] Mézard M., Parisi G., and Virasoro M. *Spin Glass Theory and Beyond.* World Scientific Pub., Singapore, 1987.

[18] Monasson R. Properties of neural networks storing spatially correlated patterns. *J. Phys. A: Math. and Gen.*, 25:3701–3720, 1992.

[19] Bialek W. and Zee A. Understanding the efficiency of human perception. *Phys. Rev. Lett.*, 61:1512–1515, 1988.

[20] Nadal J.-P. and Parga N. Information processing by a perceptron in an unsupervised learning task. *LPSENS preprint, to appear in NETWORK.*

[21] Nadal J.-P. and Parga N. Information processing by a perceptron. In *Neural networks from biology to high energy physics, Elba 1992.* Int. Journ. of Neur. Syst., 1992.

[22] Minsky M.L. and Papert S.A. *Perceptrons.* M.I.T. Press, Cambridge MA, 1988.

[23] Krauth W. and Mézard M. Storage capacity of memory networks with binary couplings. *J. Physique (France)*, 50:3057, 1989.

[24] Gutfreund H. and Stein Y. Capacity of neural networks with discrete synaptic couplings. *J. Phys. A: Math. and Gen.*, 23:2613, 1990.