Learning strategy for the binary perceptron

1996 J. Phys. A: Math. Gen. 29 6247

(http://iopscience.iop.org/0305-4470/29/19/010)

J. Phys. A: Math. Gen. **29** (1996) 6247–6252. Printed in the UK

Learning strategy for the binary perceptron

L Reimers, M Bouten and B Van Rompaey
Limburgs Universitair Centrum, B-3590 Diepenbeek, Belgium

Received 11 April 1996

Abstract. Pursuing the work of Penney and Sherrington, we determine the *optimal* continuous-weight perceptron which, on clipping, correctly predicts the largest number of weights for the binary perceptron with maximum stability. We calculate the fraction of correctly predicted binary weights when only the continuous weights stronger than a certain threshold are clipped. We finally carry out simulations for a perceptron with 50 weights to test the practicability of different learning strategies.

Neural networks must learn before they can perform. During the learning phase, the synaptic strengths are adjusted so as to reproduce a given set of training examples, called patterns. Information about these patterns is thereby stored in the synapses of the network. If the learning process has been successful, the network will be able to perform well on new input examples too. For a network that functions as a memory device [1], it should be able to recognize one of the stored patterns when a noisy version of it is presented. If the classification of the training examples is governed by an underlying rule [2], the network should be able to generalize, i.e. implement the rule on a new input. In the following we will examine the learning problem for a simple perceptron network that functions as a memory device but the discussion could easily be extended to the generalization problem.

Several good learning algorithms exist for a perceptron whose synapses may take a continuum of values. The simple Hebb rule splendidly stores a small number of random patterns. The more sophisticated AdaTron algorithm [3] yields the maximum stable network (MSN) [4], provided exact storage of the patterns is possible. By contrast, for networks with discrete synapses and in particular for its simplest example, the binary perceptron, no reliable algorithm exists except complete enumeration of all possible states of the synapses. As the number of these states grows exponentially fast with the number of synapses, this method is practically limited to networks with fewer than 30 synapses.

In this paper, we reconsider the learning problem for the binary perceptron. We call N the number of input units and $p = \alpha N$ the number of patterns $\{\boldsymbol{\xi}^\mu, \zeta^\mu\}$ $(\mu = 1, \ldots, p)$ to be stored. The N-dimensional input vectors $\boldsymbol{\xi}^\mu$ are randomly chosen on the surface of the hypersphere $\boldsymbol{\xi} \cdot \boldsymbol{\xi} = N$ and the outputs ζ^μ are random ± 1. Learning involves finding N synaptic strengths or weights $B_i = \pm 1$ $(i = 1 \ldots N)$ which make the perceptron produce the correct output ζ^μ for each of the p inputs $\boldsymbol{\xi}^\mu$. This is equivalent to demanding that the p aligning fields $\Lambda_\mu = \zeta^\mu \sum_i B_i \xi_i^\mu / \sqrt{N}$ all be positive. The stronger condition $\Lambda_\mu \geqslant K_B$ $(\mu = 1 \ldots p)$ with the largest possible value for the stability bound K_B defines the maximum stable binary network (MSB).

Many characteristics of the MSB have been calculated [5] in the thermodynamic limit $N \to \infty$. Unlike the continuous-weight vector MSN which is unique for all values of

0305-4470/96/196247+06$19.50

the storage ratio α, the MSB weights are not unique [6] except for α very close to zero. For larger values of α, many different binary vectors exist, all with the same maximum stability $K_B(\alpha)$. Near the saturation limit $\alpha = 0.83$, these binary vectors differ in as much as 20% of their components. When in the following we refer to the MSB in our theoretical calculations, we always mean the *average* over the set of these binary vectors. By contrast, in numerical simulations for finite values of N, there exists a single binary vector with maximum stability. The degeneracy occurring in the limit $N \to \infty$, disappears for finite N but we expect many different binary vectors with stabilities spread over a narrow range of values.

A number of learning schemes have been proposed for the binary perceptron. They can conveniently be grouped in two classes. One kind of approach operates directly in the set of 2^N binary vectors which are the corners of the N-dimensional hypercube. Using a cost function that penalizes small or negative values of Λ_μ, one tries to find the corner of the hypercube with lowest cost using general optimization techniques like simulated annealing [7, 8] or genetic algorithms [8, 9]. The problem is extremely hard, due to the huge number of local minima in which the minimization procedure may get trapped. Nevertheless, in its most sophisticated form [8], the method has met with some success for networks as large as $N = 100$, but the results deteriorate with increasing N. The alternative and more appealing approach [6, 9, 11] tries to draw some advantage from the fact that efficient algorithms exist for the learning problem with continuous weights. It is based on the reasonable assumption that the MSN and MSB weights are correlated since both try to maximize the stability. It should therefore be possible to extract useful information about the MSB weights from the knowledge of the MSN.

The simplest way to construct a binary vector from a vector with continuous components is by clipping. Although this generates the binary vector nearest to the original one, their moderate typical overlap of $\sqrt{2/\pi}$ gives a hint that clipping is not a gentle operation. The destructive effect of clipping becomes manifest by looking at the stability-field distribution of the clipped MSN [6]. This finding, however, need not imply that the clipped MSN vector is of no use. Penney and Sherrington [6] have calculated the fraction of components in the clipped MSN that correctly predict the corresponding component in the MSB. The fraction is large, going from 90% for $\alpha = 0.1$ down to 80% near the saturation limit $\alpha = 0.83$. These numbers demonstrate that, even though clipping the MSN does not directly provide a good approximation to the MSB, it does supply a good initial vector from which a supplementary training process may get started. The additional training, moreover, can be confined to the exploration of the neighbouring vectors of order up to $0.2N$.

A further reduction of the problem could be achieved if one were able to identify the correct components (or part of them). On the basis of numerical simulations for small systems $N \leqslant 25$, Penney and Sherrington [6] make the interesting suggestion that the large-size components of the MSN are very likely to give the correct prediction for the MSB. This means that the 20% wrong signs in the clipped MSN must primarily be sought among the 40% weakest weights of the MSN. This suggestion, if correct for general N, would drastically reduce the effective size of the original problem as 60% of the MSB components could directly be obtained from the MSN. The remaining components would then be determined by complete enumeration or by general optimization techniques.

The first problem we address in this paper is the question whether the MSN is the best choice among all continuous-weight vectors for use as a starting vector in a search for the MSB. For this purpose, we consider a general class of learning algorithms, defined by means of a cost function with a unique minimum on the hypersphere $\boldsymbol{J}^2 = N$. More specifically, we consider cost functions of the form $E(\boldsymbol{J}) = \sum_\mu V(\lambda_\mu)$ with $\lambda_\mu = \zeta^\mu \boldsymbol{J}\cdot\boldsymbol{\xi}^\mu/\sqrt{N}$ [12, 13].

Common learning rules like Hebb or MSN are contained in this class of algorithms [14]. Using standard techniques [6, 15], it is possible to derive a general equation for the overlap $r = \boldsymbol{J} \cdot \boldsymbol{B}/N$ of the continuous-weight vector $\boldsymbol{J}$ which minimizes $E(\boldsymbol{J})$ and the MSB vector $\boldsymbol{B}$. We write down the result and refer to [6] for details of a similar calculation :

$$r = \alpha \int \mathrm{D}t\,(\lambda_0 - t)\; g_r(t). \tag{1}$$

As usual, $\mathrm{D}t = \mathrm{d}t\, \exp(-t^2/2)/\sqrt{2\pi}$. The first factor $(\lambda_0 - t)$ in the integrand depends on the choice of $V(\lambda)$ while the second factor $g_r(t)$ is largely determined by the MSB. The function $\lambda_0(t, x)$ is the value of λ which minimizes $V(\lambda) + (\lambda - t)^2/2x$ where x must satisfy the saddle-point equation:

$$1 = \alpha \int \mathrm{D}t\,(\lambda_0(t, x) - t)^2. \tag{2}$$

The function $g_r(t)$ is expressed as

$$g_r(t) = \frac{\sqrt{1-q}}{\sqrt{2\pi}} \int \mathrm{d}s\, \frac{\mathrm{e}^{-\frac{(s-\gamma t)^2}{2(1-\gamma^2)}}}{\sqrt{2\pi(1-\gamma^2)}} \frac{\mathrm{e}^{-\frac{(K_B - \sqrt{q}s)^2}{2(1-q)}}}{H\left(\frac{K_B - \sqrt{q}s}{\sqrt{1-q}}\right)} \tag{3}$$

where $H(x) = \int_x^\infty Dt$. The function $g_r(t)$ is determined by the order parameter q and the stability bound K_B of the binary perceptron [5]. Its dependence on the overlap r comes via γ, a shorthand for $r/\sqrt{q}$. For a given choice of $V(\lambda)$, one has to solve equation (1) for $r(\alpha)$.

We now want to determine the optimal potential which yields the largest possible overlap with the MSB. To this end, we combine equations (1) and (2) to obtain [16, 17]

$$r^2 = \alpha \frac{[\int \mathrm{D}t\,(\lambda_0 - t) g_r(t)]^2}{\int Dt\,(\lambda_0 - t)^2}. \tag{4}$$

Using the Schwartz inequality yields an upperbound for the r.h.s. of (4). Thus $r^2 \leqslant \alpha \int Dt\,(g_r(t))^2$ with the equality sign holding iff $\lambda_0 - t = C g_r(t)$ where C is an arbitrary constant. The equality sign defines an upperbound for r because $r^2/\int Dt\,(g_r(t))^2$ is independent of the choice of $V(\lambda)$ and a strictly increasing function of r. The maximum overlap obtainable within the considered class of algorithms is the solution of the equation

$$r^2 = \alpha \int \mathrm{D}t\,(g_r(t))^2. \tag{5}$$

On clipping a continuous vector $\boldsymbol{J}$ that has overlap $r(\alpha)$ with the MSB, the fraction of components that agree with the MSB is given by [6]

$$f = \frac{1}{2} + \int \mathrm{D}z \tanh\left(\sqrt{\hat{q}}z\right) H\left(-\frac{\hat{\gamma}}{\sqrt{1-\hat{\gamma}^2}}z\right) \tag{6}$$

where $\hat{q}$ is the order parameter conjugate to q [5] and $\hat{\gamma}$ is a shorthand for $r/\sqrt{\hat{q}}(1-q)$. The sole dependence on $V(\lambda)$ is through the overlap r hidden in $\hat{\gamma}$. The maximum possible value for f is obtained by using the largest value for r i.e. the solution of (5).

Figure 1(a) shows the fraction of binary weights that are correctly predicted by clipping the optimal continuous-weight vector and by clipping the MSN. The difference only shows up at large values of α where it is about 2%. For comparison, we also show the upperbound $f = (1 + \sqrt{q})/2$ which follows from the fact that any two MSB vectors have overlap q [18].

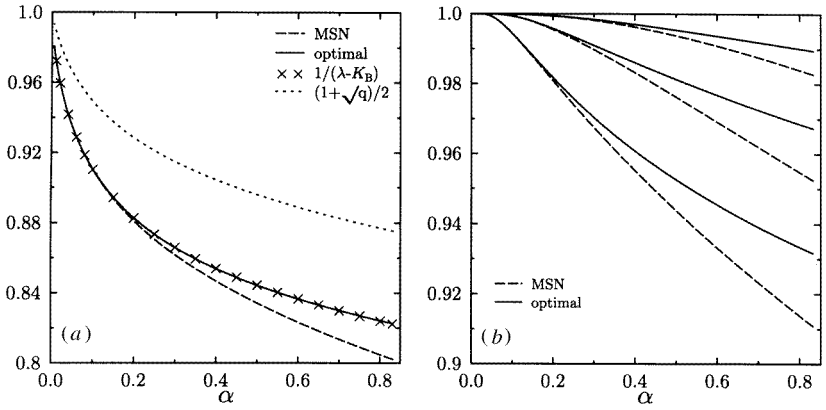

Figure 1. (*a*) The fraction of binary weights in the MSB that are correctly predicted by clipping the continuous weights of the MSN and of the networks defined by the optimal potential and its substitute $1/(\lambda - K_B)$. The dotted curve shows a simple upperbound. (*b*) The same fraction for the MSN and the optimal potential when only the strongest 20% (upper curves), 40% (middle curves) or 60% (lower curves) weights are clipped.

With the aim of better understanding the role of the optimal potential we have focused on the centre-of-mass of the continuous Gibbs ensemble with stability K_B and have calculated its overlap with the MSB. This turns out to be almost equal to the optimal overlap. As in the case of supervised learning [13], we therefore use a simple substitute for the optimal potential which is equal to infinity when $\lambda < K_B$ and given by $1/(\lambda - K_B)$ when $\lambda > K_B$. As seen in figure 1(*a*), the value of f for this substitute coincides almost perfectly with the result for the optimal potential.

In a second problem, we want to assess the validity of the Penney–Sherrington suggestion for large values of N. For this purpose, we focus on the components of the continuous-weight vector $\boldsymbol{J}$ that are larger than a threshold J_0 and calculate the probability that any of these components corresponds to $+1$ in MSB. This probability rapidly increases with increasing J_0, thus substantiating the Penney–Sherrington suggestion. A derived quantitative measure for its validity is the fraction of these large components that correctly predict the MSB component. It is given by a direct extension of (6)

$$f(J_0) = \frac{1}{2} + \frac{1}{H(J_0)} \int \mathrm{D}z \tanh\left(\sqrt{\hat{q}}z\right) H\left(\frac{J_0 - \hat{\gamma}z}{\sqrt{1-\hat{\gamma}^2}}\right). \tag{7}$$

This fraction is shown in figure 1(*b*) for the optimal potential and for the MSN for three values of J_0 corresponding to clipping the 20%, 40% and 60% strongest weights.

We have tested the practicability of different learning strategies by performing simulations for a perceptron with $N = 50$. The results are shown in figure 2. The full curve displays the theoretical value $K_B(\alpha)$ [5]. The data points show the minimum pattern stability as obtained from three different strategies. Each point represents the average over 100 random samples. In all cases, we start by determining the MSN using AdaTron. We then follow three straightforward strategies using the MSN as starting vector. Results of similar calculations in which the optimal potential is used are not presented here because, for $N = 50$, the increase in pattern stability is very small. From our theoretical results, we expect the difference to grow with N.

The simplest strategy is plain clipping. As expected, it gives a poor lower bound for $K_B(\alpha)$. The next strategy, being the straightforward implementation of the Penney–

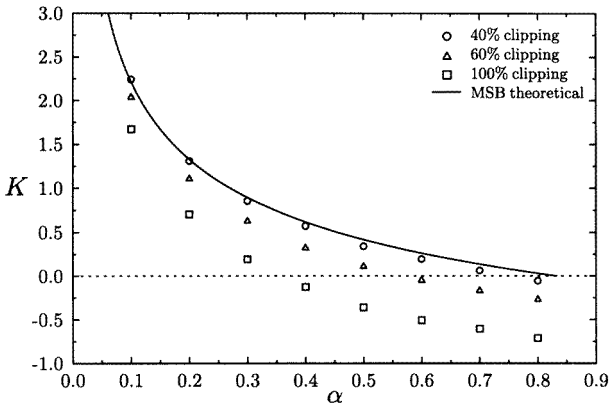

Figure 2. Minimum pattern stability K as a function of α as determined from numerical simulations for a perceptron with $N = 50$ following different strategies described in the text.

Sherrington suggestion, consists in clipping the strongest 30 weights and determining the remaining components by enumerating all 2^{20} possibilities. This second strategy is fast and yields a good estimate for $K_B(\alpha)$ at low α. Not surprisingly, the estimate deteriorates at higher values of α where a growing fraction of clipped components will have the wrong sign. Our third and more expensive strategy consists in clipping the strongest 20 weights only, leaving 30 components to be determined by further training. Rather than enumerating all 2^{30} possibilities, we confine the search for these 30 components to the vector obtained by clipping the remainder of the MSN and all its neighbours up to tenth order. This restriction entails a reduction of the number of explored vectors by a factor 20. Nevertheless, the agreement with the theoretical curve becomes excellent for small α and, at larger values of α, the discrepancy is small. It is clear that many more sophisticated strategies can be designed that may remove this discrepancy. Further strategies will be presented elsewhere together with numerical results for larger values of N.

Acknowledgments

We thank the Inter-University Attraction Poles of the Belgian Government for financial support. BVR also acknowledges support from the NFWO Belgium.

References

[1] Amit D J 1989 *Modelling Brain Function* (Cambridge: Cambridge University Press)
[2] Watkin T L M, Rau A and Biehl M 1993 *Rev. Mod. Phys.* **65** 499
[3] Anlauf J K and Biehl M 1989 *Europhys. Lett.* **10** 687
[4] Gardner E J 199 *J. Phys. A: Math. Gen.* **21** 257
[5] Krauth W and Mézard M 1989 *J. Physique* **50** 3057
[6] Penney R W and Sherrington D 1993 *J. Phys. A: Math. Gen.* **26** 6173
[7] Horner H 1992 *Z. Phys.* B **86** 291
[8] Köhler H M 1990 *J. Phys. A: Math. Gen.* **23** L1265
[9] Fontanari J F and Meir R 1991 *Network* **2** 353
[10] Perez Vicente C J, Carrabina J and Valderrana E 1992 *Network* **3** 165
[11] Schietse J, Bouten M and Van den Broeck C 1995 *Europhys. Lett.* **32** 279
[12] Wong K Y M and Sherrington D 1990 *J. Phys. A: Math. Gen.* **23** 4659
[13] Bouten M, Schietse J and Van den Broeck C 1995 *Phys. Rev.* E **52** 1958

[14] Griniasti M and Gutfreund H 1991 *J. Phys. A: Math. Gen.* **24** 715
[15] Wong K Y M, Rau A and Sherrington D 1992 *Europhys. Lett.* **19** 559
[16] Kinouchi O and Caticha N 1996 Learning Algorithm which gives the Bayes generalization limit for perceptrons *Preprint* Sao Paulo, Brazil
[17] Van den Broeck C and Reimann P 1996 *Phys. Rev. Lett.* **76** 2188
[18] Watkin T L H 1993 *Europhys. Lett.* **21** 871