ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 175-180

Precursor networks for training the binary perceptron

Bart Van Rompaey *

Limburgs Universitair Centrum
B-3590 Diepenbeek, Belgium

Abstract. Considering the storage problem for the binary perceptron Penney and Sherrington studied the properties of clipping the maximum stable continuous network and calculated the fraction of weights that correctly predict the binary weights. One can wonder whether there are better continuous networks than the maximum stable one to start from. In this work other starting vectors (precursors) will be presented which perform better on clipping than the maximum stable perceptron. We will show that precursors obtained by minimizing with a cost function in the hypercube instead of on the hypersphere gives better results.

1. Introduction

Perceptrons with synaptic weights which are continuous are typically easier to train than those with discrete–valued weights. There are several learning algorithms for a continuous–weight perceptron. The Hebb rule is able to store a small number of random patterns. The AdaTron [1] algorithm yields the maximum stable network (MSN) for the storage problem. On the other hand for the discrete–valued weights perceptron, and in particular for the binary perceptron, there does not exist an efficient learning algorithm except complete enumeration of all possible states of the synapses. Since computers are limited in processing speed this is only practical for networks with less than 30 synapses. So it would be nice to have a at least less expensive learning strategy.

Here we will reconsider the storage problem for the binary perceptron. There are basically two different learning strategies which have been developed. One kind of strategy operates directly in the set of 2^N binary vectors on the corners of the N-dimensional hypercube. One uses a cost function to find the corner of the hypercube with lowest cost. This problem however is extremely hard since there are many local minima. The other approach [4, 2], and also the one we will consider, uses information from the storage problem with continuous weights. It is based on the assumption that the MSN and max-

Acknowledgement: I would like to acknowledge financial support from the Fund for Scientific Research – Flanders (FWO) as a Research Assistant.

*e–mail: bart@luc.ac.be

imum stable binary (MSB) weights are correlated since both try to maximize the stability.

We will ask whether the MSB can be sufficiently neared by a perceptron with continuous weights which can be obtained by a practical learning algorithm. We call such a continuous network a precursor to the binary one. After having found such a precursor we can try to obtain a good approximation of the MSB by intelligent methods of partly clipping and partly enumerating weights. In the following we will present different precursors and look at their ability to predict the components of the MSB.

2. The MSN as a precursor

An obvious thing to do is to use the MSN as a precursor. Penney and Sherrington [4] have investigated the properties of the MSN/MSB system and calculated the fraction of correctly predicted binary weights on plain clipping. This fraction is large going from 90% for $\alpha = 0.1$ down to 80% near the critical capacity $\alpha = 0.83$ [5]. Further, on the basis of numerical simulations for small systems $N \leq 25$ they 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.

We performed simulations, by using this suggestion, for a perceptron with $N = 50$. The results are shown in figure 1. The full curve displays the theoretical value $\kappa_B(\alpha)$. The data points show the minimum pattern stability as obtained from two different strategies using the MSN as precursor. Each point presents the average over 200 samples. The simplest strategy is plain clipping. It gives a poor lower bound for $\kappa_B(\alpha)$. The next strategy, being the straightforward implementation of the Penney–Sherrington suggestion, consists in clipping the strongest 30 weights and determining the remaining components by enumerating all 2^{20} possibilities. This strategy is fast and yields an excellent estimate for $\kappa_B(\alpha)$ at low α, although the estimate deteriorates at higher values of α where a small fraction of clipped components will have the wrong sign. One notices that for small α the data points are slightly above the theoretical curve. We did simulations for other smaller values of N and it turned out to be a finite size effect.

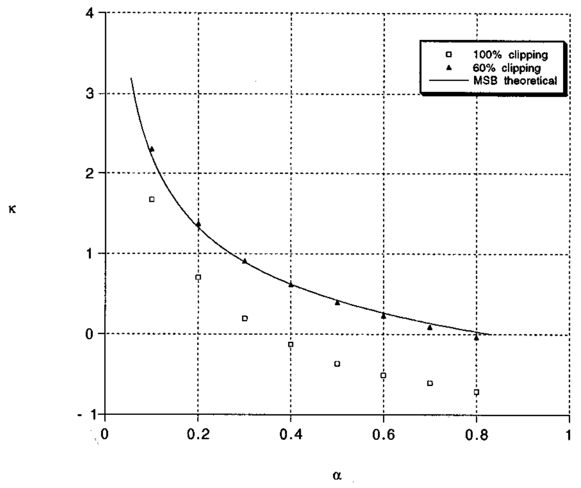

Figure 1: Minimum pattern stability κ as a function of α as determined from numerical simulations for a perceptron with $N = 50$ following different strategies described in the text. The MSN is used as precursor.

3. Towards better precursors

3.1. The optimal potential

Naturally the question arises if there are better precursors on the hypersphere than the MSN. In order to answer this question we consider a set of precursors in which the MSN is contained. The precursors we will consider here are obtained by a class of learning algorithms defined by means of a cost function with a unique minimum on the hypersphere $\mathbf{J}^2 = N$. More specifically we will consider cost functions of the form $E(\mathbf{J}) = \sum_\mu V(\lambda_\mu)$ with λ_μ the stability of pattern μ. Common learning rules like Hebb and MSN are contained in this class of algorithms.

Our calculations [3] show that there exists an optimal potential, within the class of algorithms. Optimal in the sense that the precursor on clipping, correctly predicts the largest number of MSB components. Although we were not able to derive this potential analytically we succeeded in obtaining a very good approximation of the optimal potential. This potential is given by:

$$V(\lambda) = \begin{cases} \frac{1}{\lambda - \kappa_B} & \text{if} \quad \lambda \geq \kappa_B \\ \infty & \text{if} \quad \lambda < \kappa_B \end{cases} \tag{1}$$

When one uses this potential to obtain the precursor on the hypersphere, one

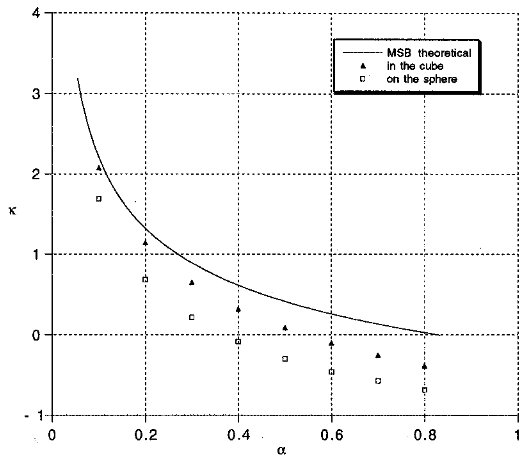

Figure 2: Comparison between the hypercube and hypersphere. Minimum pattern stability κ is shown as a function of α as obtained from numerical simulations for a perceptron with $N = 50$. The data points are for a precursor which is fully clipped. The lowest data points are for the hypersphere–case, the others are for the hypecube.

expects a larger fraction of correctly predicted binary weights by clipping than with any other potential. Indeed, when one compares that fraction for the MSN and the optimal precursor, for instance, the difference between the latter and the former is approximately 2% at $\alpha = 0.83$ in favor of the optimal precursor.

Simulations with the precursor obtained from (1) are not included here since, for $N = 50$, the increase in pattern stability is very small so that they would result in the same curves as in figure 1. From our theoretical results, we expect the difference to grow with N.

3.2. The hypercube

There is still a way to improve the previous results. We tried to change the domain on which we minimize the cost function. Instead of minimizing on the hypersphere we tried to minimize the cost function *in* the hypercube $|J_i| \leq 1$. A simple intuitive reasoning shows that this might improve the above results. Since one tries to maximize the stabilities of all the patterns, weight vectors which are longer are favoured against shorter weight–vectors. When we minimize the cost function with a starting vector in the hypercube this vector will always evolve, during the minimization, towards the borders of the hypercube

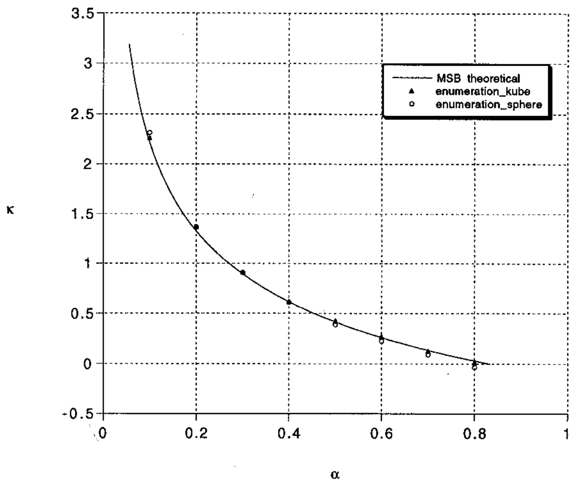

Figure 3: Comparison between the hypercube and the hypersphere. Minimum pattern stability κ is shown as a function of α as obtained from numerical simulations for a perceptron with $N = 50$. The data points are for a precursor of which 30 weights are clipped and the others enumerated. The lowest data points are for the hypersphere–case, the others are for the hypercube.

and preferably towards a corner, since there the vector is the longest. However a corner will only be reached for very small values of α. This is because, for larger values of α, some weights which give difficulty in obtaining stability for all the patterns will stay much smaller than 1 in absolute value.

Instead of searching for the optimal potential in this case, we investigated the behaviour of potential (1). It turns out that this behaviour is almost optimal. So we decided also to work with this potential in this case.

We calculated the fraction of correctly predicted binary weights by clipping with this potential. When one compares that fraction with the one obtained with the same potential on the hypersphere, the difference between the latter and the former is approximately 2% at $\alpha = 0.83$ in favor of the hypercube. After minimizing with potential (1) in the hypercube there will be about 50% of the components ± 1 for $\alpha = 0.83$. This number increases for smaller values of α as expected intuitively. We performed simulations again for a perceptron with $N = 50$ in order to compare with the hypersphere. The results are shown in figure 2 and 3. The full curve displays the theoretical value $\kappa_B(\alpha)$. The data points show the minimum pattern stability using the precursor obtained by (1). Each point presents the average over 200 samples. The lower data points in figure 2 are the results obtained by using the hypersphere as minimization domain, and the precursor obtained is then fully clipped. The next set of data points in figure 2 are simulations where the hypercube is used and again where the precursor is fully clipped. These two simulations really indicate that the hypercube is superior over the hypersphere. The last strategy is one where we minimize again on the hypercube but we clip the 30 largest components (of which many are already ± 1, if not all, as is very probable for small α) and enumerate the other 2^{20} possibilities. These results are shown in figure 3 together with the results for the hypersphere using the same strategy as mentioned above. For small α the two strategies are almost identical. The strategy which uses the hypercube is slightly better for α close to the critical capacity.

4. Outlook

We presented simulations where we obtained satisfying results for the storage problem for a perceptron with $N = 50$. We showed that using the hypercube as minimization domain provides a way of obtaining better results than with the hypersphere. It is obvious that one can obtain better or the same results than on the hypersphere by using the hypercube and less expensive methods. Of course if we take N larger, say $N = 100$, then the method of clipping and enumerating is not practical anymore since we may at most clip 60% of the components which still leaves 40 components to be enumerated. This is simply not possible. But with the results obtained with the hypercube we have hope that we could develop a more complicated strategy that will work for $N > 50$. For $N > 100$ the problem remains difficult and even our least expensive strategy will be hard to implement.

References

[1] J. K. Anlauf and M. Biehl. The AdaTron: an adaptive perceptron algorithm. *Europhys.Lett.*, 10:687, 1989.

[2] J. Schietse, M. Bouten and C. Van den Broeck . Training binary perceptrons by clipping. *Europhys.Lett.*, 32:279, 1995.

[3] L. Reimers ,M. Bouten and B. Van Rompaey. Learning strategy for the binary perceptron. *J.Phys.A:Math.Gen.*, 29:6247, 1996.

[4] Penney and Sherrington. The weight space of the binary perceptron. *J.Phys.A:Math.Gen.*, 26:6173, 1993.

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