Increasing the capacity of a Hopfield network without sacrificing functionality

Amos Storkey

Imperial College

Abstract. Hopfield networks are commonly trained by one of two algorithms. The simplest of these is the Hebb rule, which has a low absolute capacity of $n/(2 \ln n)$, where n is the total number of neurons. This capacity can be increased to n by using the pseudo-inverse rule.
However, capacity is not the only consideration. It is important for rules to be local (the weight of a synapse depends ony on information available to the two neurons it connects), incremental (learning a new pattern can be done knowing only the old weight matrix and not the actual patterns stored) and immediate (the learning process is not a limit process). The Hebbian rule is all of these, but the pseudo-inverse is never incremental, and local only if not immediate. The question addressed by this paper is, 'Can the capacity of the Hebbian rule be increased without losing locality, incrementality or immediacy?'
Here a new algorithm is proposed. This algorithm is local, immediate and incremental. In addition it has an absolute capacity significantly higher than that of the Hebbian method: $n/\sqrt{2 \ln n}$.
In this paper the new learning rule is introduced, and a heuristic calculation of the absolute capacity of the learning algorithm is given. Simulations show that this calculation does indeed provide a good measure of the capacity for finite network sizes. Comparisons are made between the Hebb rule and this new learning rule.

1 Introduction

Attractor networks such as Hopfield [4] networks are used as autoassociative content addressable memories. The aim of such networks is to retrieve a previously learnt pattern from an example which is similar to, or a noisy version of, one of the previously presented patterns. To do this the network associates each element of a pattern with a binary neuron. These neurons are fully connected, and are updated asynchronously and in parallel. They are initialised with an input pattern, and the network activations converge to the closest learnt pattern.

In order to perform in the way described, the network must have a learning algorithm which sets the connection weights between all pairs of neurons so that it can perform this task.

These learning rules can have a number of characteristics. Firstly a rule can be local. If the update of a particular connection depends only on information available to the neurons on either side of the connection, then the rule is said to be local. Locality is important, because it provides a natural parallelism to the learning rule, which, when combined with the local update dynamics, make a Hopfield network a truly parallel machine.

Secondly a rule can be incremental. If the learning process can modify an old network configuration to memorise a new pattern, without needing to refer to any of the previously learnt patterns, then an algorithm is called incremental. Clearly incrementality makes the Hopfield network adaptive, and therefore more suitable for changing environments or real time situations.

Thirdly a rule can either perform an immediate update of the network configuration, or can be a limit process. The former makes for faster learning.

Lastly and most importantly a learning algorithm has a capacity. This is some measure of how many patterns can be stored in a network of a given size. More specifically, in this paper we consider the absolute capacity [6] of the network. This is given by the asymptotic ratio of the number of patterns that can be stored without error to the number of neurons, as the network size tends to infinity.

Capacity is not just important because it allows more efficient information storage. Because the update time is at least proportional to the number of neurons, higher capacity also allows faster processing times.

1.1 Different learning rules

The most common, and indeed the simplest, learning rule for Hopfield networks is called the Hebb rule. The Hebb rule is local, incremental and immediate, but has an absolute capacity of $n/(2\ln n)$ [6].

To increase the capacity of the network, the pseudo-inverse learning rule can be used. This has a capacity of n [5], but does not have the functionality of the Hebb rule. It is not incremental, and in its most common form, is not local either. It is possible to create a pseudo-inverse weight matrix by a local method, but only as a limit process rather than with immediate learning [3, 2].

Another way to increase the capacity to $n/\sqrt{2\ln n}$ is through the use of non-monotonic continuous neurons [7]. However this increases the computational and storage burden because the response function of each neuron, as well as the weight matrix, can change, and neuron activations can take a whole range of values, rather than just binary ones.

It is therefore important to look for new learning methods which increase the capacity of the network from that of Hebbian learning, but without sacrificing important functionality, such as locality, incrementality, or immediacy.

This paper introduces a learning rule which keeps this full functionality, and increases the capacity of the network to $n/\sqrt{2\ln n}$, a significant improvement on that of Hebbian learning.

2 Framework

This section outlines the important concepts used in this paper.

A Hopfield network can be defined by the coupled difference equations

$$\sigma_i(t+1) = sgn\left(\sum_{j=1, j\neq i}^{n} w_{ij}\sigma_j(t)\right) \text{ for } i = 1, 2, \ldots, n$$

where $\sigma_i(t) = \pm 1$ is the state of neuron i after the tth update, and $W = w_{ij}$ is the weight matrix given by the learning rule, and is, in all cases we will consider, symmetric. The network is initialised with a pattern S_i by setting $\sigma_i(0) = S_i$, and the time evolution of the network will quickly settle into an equilibrium.

For a useful learning rule this point will be the learnt pattern which is closest (suitably defined) to the pattern the network was initialised with, and so the aim of a learning rule is to set up the weight matrix so that the patterns to be learnt are attractors of the system.

2.1 Characteristics of a learning rule

The characteristics which have been called locality, incrementality and immediacy have already been mentioned. Here these concepts are made more precise.

In what follows ν counts the patterns (denoted ξ^ν) to be learnt. W^ν denotes the state of the weight matrix after the $1\ldots\nu$th patterns have been learnt but before the $(\nu+1)$th pattern has been introduced.

An incremental learning rule is any where W^ν is a function only of $W^{\nu-1}$ and ξ^ν.

A local learning rule has $w_{ij}(s)$ dependent only on ξ_i^μ, ξ_j^μ, $w_{ik}(s-1)\xi_k^\mu$ and $w_{jk}(s-1)\xi_k^\mu$, where $\mu = 1, 2, \ldots, \nu$, $k = 1, 2, \ldots, n$. Here s counts through the steps in the evolution of the weight matrix W.

An immediate learning process takes a finite number of steps to obtain W^ν. Any process which is not immediate is called a limit process. Note that a workable weight matrix can be obtained by truncating the limit process, but even so, most limit processes are slower at obtaining a weight matrix than immediate ones.

The three learning rules used in this paper are given by

Definition 1 Hebbian learning rule. The Hebbian learning rule is given by

$$w_{ij}^0 = 0 \ \forall i, j \text{ and } w_{ij}^\nu = w_{ij}^{\nu-1} + \frac{1}{n}\xi_i^\nu \xi_j^\nu$$

Definition 2 Pseudo-inverse learning rule. The pseudo inverse is given by

$$w_{ij}^\nu = \frac{1}{n}\sum_{\nu=1}^{m}\sum_{\mu=1}^{m}\xi_i^\nu (Q^{-1})^{\nu\mu}\xi_j^\mu$$

where $Q = \frac{1}{n}\sum_{k=1}^{n}\xi_k^\nu \xi_k^\mu$, and m is the total number of patterns with $m < n$. It can be seen from the form of these definitions that the Hebbian rule is local, incremental and immediate, but that the pseudo-inverse is neither local nor incremental, because it involves calculating an inverse.

Definition 3 The new learning rule. The weight matrix of an attractor neural network is said to follow the new learning rule if it obeys

$$w_{ij}^0 = 0 \ \forall i, j \text{ and } w_{ij}^{\nu} = w_{ij}^{\nu-1} + \frac{1}{n}\xi_i^{\nu}\xi_j^{\nu} - \frac{1}{n}\xi_i^{\nu}h_{ji}^{\nu} - \frac{1}{n}h_{ij}^{\nu}\xi_j^{\nu} \tag{1}$$

where $h_{ij}^{\mu} = \sum_{k=1,k\neq i,j}^{n} w_{ik}^{\mu-1}\xi_k^{\mu}$ is a form of local field at neuron i, and ξ^{μ} is the new pattern to be learnt.

Once again it is clear from the form of the above that the new learning rule is local, incremental and immediate.

3 The capacity: Signal to noise analysis

For a particular pattern ξ^{ν} to be a fixed point of the system, we require that $\sum_{j=1,j\neq i}^{n} \xi_i^{\nu} W_{ij} \xi_j^{\nu} > 0$ for all $i \leq n$. After expanding out the new learning rule this gives the condition that

$$\begin{aligned} 1 + N_i^{\nu} \;=\; & 1 + \sum_{r=1}^{m} \frac{(-1)^{r-1}}{n^r} \sum_{j=1}^{n} \sum_{\mu_r=1,\mu_r\neq\nu}^{m} \left(\prod_{s=1}^{r-1} \sum_{\mu_s=1}^{m} \sum_{k_s=1}^{n} \right) S(\mu_1, \mu_2, \ldots, \mu_r; \nu) \\ & T(i, k_1, \ldots, k_{n-1}, j)\xi_i^{\nu}\xi_i^{\mu_1} \left(\prod_{t=1}^{r-1} \xi_{k_t}^{\mu_t}\xi_{k_t}^{\mu_{t+1}} \right) \xi_j^{\mu_r}\xi_j^{\nu} > 0 \end{aligned} \tag{2}$$

where the $(\prod_{a=1}^{n} \sum_{z_a} \sum_{y_a})$ notation is used to represent $\sum_{z_1}\sum_{y_1}\sum_{z_2}\sum_{y_2}\cdots\sum_{y_n}\sum_{z_n}$, m is the number of patterns stored, $S(a_1, a_2, \ldots, a_r; b) = R(a_1, a_2, \ldots, a_r) - \frac{n-2}{n} R(a_1, a_2, \ldots, a_r, b)$, and

$$R(a_1, a_2, \ldots, a_r) = \begin{cases} 1 \text{ if } \exists p \text{ such that } a_1 > \ldots > a_p < a_{p+1} < \ldots < a_r \\ \quad \text{and } a_i \neq a_j \ \forall i \neq j \\ 0 \text{ otherwise} \end{cases}$$

$$T(a_1, a_2, \ldots, a_r) = \begin{cases} 1 \text{ if } a_i \neq a_j \text{ for all } i, j \leq r \\ 0 \text{ otherwise} \end{cases}$$

The N_i^{ν} is called the noise at neuron i for the recall of memory ν. If $N_i^{\nu} < -1$ then the pattern ξ^{μ} is unstable at neuron i and is therefore recalled incorrectly.

In order to make any estimate of the capacity of this rule something must be known about the patterns that are to be stored. Here the patterns are assumed to be random in the usual way [6] The ξ_k^{μ} are taken to be random variables with $\xi_k^{\mu} = \pm 1$ and $P(\xi_k^{\mu} = +1) = \frac{1}{2}$. The variables are assumed to be mutually independent for all (k, μ).

Definition 4 Absolute Capacity. Suppose $m(n)$ random patterns are stored in the network size n. If in the limit $n \to \infty$ the probability that all the patterns are stable is 1, and $m(n)$ is the largest asymptotic relationship for which this occurs then $m(n)$ is the absolute capacity of the network.

Though the noise term might look very complicated, it is in fact just a sum of many ± 1 variables. Furthermore it can be shown that

$$E|N_i^\nu|^2 < \frac{m^2}{2(n^2 - m^2)}, \ \ E(N_i^\mu) = 0$$

Note that for all $r = 1$, $S = 2/n^2$, effectively eliminating order (m/n) noise.

At this point we make the natural, but unproven assumption that the noise is Gaussian, being a sum of many variables, which, although not independent, have a somewhat 'sparse' dependence. In this case the probability that a particular bit of a particular pattern is unstable is given by

$$P(N_i^\nu < -1) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(-\frac{1}{\sqrt{2}\sigma}\right) \rightarrow \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\right) \tag{3}$$

as $\sigma \rightarrow 0$. Here $\sigma^2 = E|N_i^\nu|^2 < \frac{m^2}{2(n^2-m^2)}$.

Now if we choose $m(n) = n/(\sqrt{2\ln n})$ such that $2(n^2 - m(n)^2)/m(n)^2 = 4\ln n + 1$, then we know from (3) that

$$P(N_i^\nu < -1) < p_{mn} = \frac{e}{2\pi n^2 \sqrt{4 \ln n + 1}}$$

and so $mnP(N_i^\nu < -1) < mnp_{mn} \rightarrow 0$.

Finally, because this is true for all $m \times n$ values of i, ν, the simple lemma

Lemma 5. *Let A_{nk} be an event. Let $A_n = \bigcup_{k=1}^n A_{nk}$. Then $\lim P(A_n) = 0$ if $\limsup_{n\rightarrow\infty} \sum_{k=1}^n P(A_{nk}) \rightarrow 0$*

shows that with $m(n) = n/(\sqrt{2\ln n})$, the probability that any of the bits of any of the patterns are unstable tends to zero as the network gets larger, and hence the capacity is $n/(\sqrt{2\ln n})$.

Simulations of the networks were performed, and the capacity results for the new learning rule were compared with those of the Hebb rule (Figure 1). The graph shows that the theory gives good approximations of the capacity for finite size networks.

4 Other considerations and Conclusions

Little has been said in this paper about basins of attraction and spurious states. These are important issues. Initial work in looking at these indicate that the number of spurious states and the size of the basins of attraction are comparable to that of the Hebb rule. This should not be seen as a surprise. In the new learning rule, the introduction of the attractors into the system still follows the Hebb formulation. The additional terms of the learning rule serve only to remove some of the lower order noise brought about by the interaction of different attractors, and hence increase the number of patterns that can be stored.

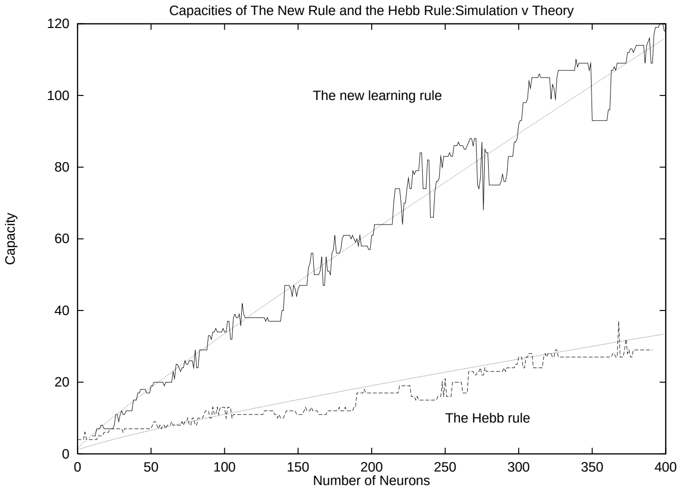

Fig. 1. Capacity of new learning rule v Hebb

This improvement of capacity for the Hopfield network is very significant. At network sizes of a million neurons, the absolute capacity of the new learning rule is five times that of the Hebb rule, increasing both storage and speed.

In this paper we have focussed only on absolute capacity. The Hebb rule has a relative capacity (where imperfect recall is allowed) of $0.14n$ [1]. Although it may be possible to calculate a relative capacity for the new learning rule, it should be noted that even at sizes of a million neurons, the absolute capacity of the new learning rule is greater than the relative capacity of the Hebb.

References

1. Daniel J. Amit, H. Gutfreund, and H. Sompolinsky. Statistical mechanics of neural networks near saturation. *Annals of Physics*, 173(1):30–67, 1987.
2. S. Diederich and M. Opper. Learning of correlated patterns in spin-glass networks by local learning rules. *Physical Review Letters*, 58(9):949–952, 1987.
3. V. S. Dotsenko, N. D. Yarunin, and E. A. Dorotheyev. Statistical mechanics of Hopfield-like neural networks with modified interactions. *Journal of Physics A: Mathematical General*, 24(10):2419–2429, 1991.
4. J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America: Biological Sciences*, 79(8):2554–2558, 1982.
5. I. Kanter and H. Sompolinsky. Associative recall of memory without errors. *Physical Review A- General Physics*, 35(1):380–392, 1987.
6. R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh. The capacity of the hopfield associative memory. *IEEE Transactions on Information Theory*, 33(4):461–482, 1987.
7. H. F. Yanai and S. I. Amari. Autoassociative memory with 2-stage dynamics of nonmonotonic neurons. *IEEE Transactions on Neural Networks*, 7(4):803–815, 1996.