Palimpsest memories: a new high-capacity forgetful learning rule for Hopfield networks

Palimpsest memories: a new high-capacity forgetful learning rule for Hopfield networks

Amos Storkey

Neural Systems Group, Dept Electrical Engineering

Imperial College, London Sw7 2AZ, UK

Preprint

Abstract

Palimpsest or forgetful learning rules for attractor neural networks do not suffer from catastrophic forgetting. Instead they selectively forget older memories in order to store new patterns. Standard palimpsest learning algorithms have a capacity of up to $0.05n$, where n is the size of the network. Here a new learning rule is introduced. This rule is local and incremental. It is shown that it has palimpsest properties, and it has a palimpsest capacity of about $0.25n$, much higher than the capacity of standard palimpsest schemes. It is shown that the algorithm acts as an iterated function sequence on the space of matrices, and this is used to illustrate the performance of the learning rule.

1 Introduction

Attractor networks such as Hopfield [1] 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. One problem with many such learning rules is that they suffer from catastrophic forgetting. As patterns are presented to the network it stores them for future retrieval. This continues until the network reaches capacity. Then if patterns continue to be presented the network is quickly unable to retrieve *any* of the stored patterns. In other words at storage levels above capacity the network forgets all it has learnt.

A number of learning rules were developed to get round this problem. They are called palimpsest or forgetful learning rules [2]. These work in a different way. Patterns are stored until capacity is reached. Then as patterns continue to be presented the network forgets the older patterns, preferring to remember the newer ones. The network will continue to remember the most recent p patterns where p is the palimpsest capacity of the network.

The problem with palimpsest memories is that they tend to have very low capacities: less than $0.05n$, compared with $0.14n$ for the Hebb rule [3, 4]. Here

we introduce a new learning rule which has a palimpsest capacity of estimated to be roughly about $0.25n$. Furthermore this rule has all the important characteristics of a Hopfield learning scheme, including locality and incrementality.

2 Learning scheme characteristics

Hopfield 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 (including weighted information these neurons receive from elsewhere), 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.

Fourthly a learning algorithm has a capacity. This is some measure of how many patterns can be stored in a network of a given size. The palimpsest capacity is the number of recent patterns which are retrievable after many patterns are stored in the network.

3 Hopfield learning rules

The update rule for Hopfield networks is given by

$$x_i(t+1) = \text{sgn}\left(\sum_{j \neq i}^{n} w_{ij} x_j(t)\right)$$

Usually Hopfield networks are trained by the Hebb rule

$$w_{ij}^0 = 0 \ \forall i, j$$

$$w_{ij}^{\mu} = w_{ij}^{\mu-1} + \frac{1}{n}\xi_i^{\mu}\xi_j^{\mu}$$

which has a capacity of about $0.14n$. It is both local and incremental. However it suffers from catastrophic forgetting. The pseudo-inverse rule is not incremental, and can never store more that n memories. It is of little use in situations where adaptive update occurs.

In [5] a new learning rule was introduced:

$$w_{ij}^0 = 0 \ \forall i, j \in \{1, 2, ..., n\}$$

$$w_{ij}^m = w_{ij}^{m-1} + \frac{1}{n}\xi_i^m\xi_j^m - \frac{1}{n}\xi_i^m h_{ji}^m - \frac{1}{n}h_{ij}^m\xi_j^m \tag{1}$$

where

$$h_{ij}^m = \sum_{k=1, k\neq i,j}^{n} w_{ik}^{m-1}\xi_k^m$$

It has a higher capacity than that of the Hebb rule, and is local and incremental. It deals well with correlated patterns [6], and has larger, more even basins of attraction than those obtained by the Hebbian scheme [7]. As it stands it also suffers from a catastrophic loss of memory after capacity is reached.

3.1 Palimpsest schemes

There are a number of different palimpsest learning rules, but they have all been based on the same principle: keeping the size of the weight matrix ele-

ments bounded. Hopfield suggested a forgetful scheme in his original paper [1]. Others have developed this framework [8, 2, 4, 9, 3, 10, 11]. Palimpsest storage prescriptions are given generally by the local rule

$$w_{ij}^m = \frac{1}{n}\phi(nw_{ij}^{m-1} + \epsilon\xi_i^m\xi_j^m)$$

where $\boldsymbol{\xi}^m$ is the pattern to be stored, ϕ is some function, and n is the size of the network. $w_{ij}^m = w_{ji}^m$ is the weight matrix after the mth pattern is stored.

Parisi formalised Hopfield's original proposal by choosing the function $\phi(x) = \mathrm{sgn}(x)\min(1, |x|)$. Nadal, Toulouse, Changeux and Dehaene [2] examined a number of learning methods including what they called the marginalist scheme ($\phi(x) = \lambda_N x$) and the smooth scheme ($\phi(x) = \tanh(x)$). The largest palimpsest capacity obtained for such schemes is about $0.05n$.

Here we modify the learning rule of [5] very slightly to obtain a palimpsest rule with a capacity greater than $0.25n$, many times that of other palimpsest schemes, and larger than the (non-palimpsest) capacity of the Hebb rule:

$$w_{ij}^0 = 0 \ \forall i, j \in \{1, 2, ..., n\}$$

$$w_{ij}^m = \begin{cases} w_{ij}^{m-1} + \frac{1}{n}\xi_i^m\xi_j^m - \frac{1}{n}\xi_i^m h_j^m - \frac{1}{n}h_i^m\xi_j^m & \text{for } i \neq j \\ 0 \text{ for } i = j \end{cases} \tag{2}$$

where

$$h_i^m = \sum_{k=1}^{n} w_{ik}^{m-1}\xi_k^m$$

4 Results

Simulations of this network were performed. Random independent zero mean ± 1 patterns were presented to a network of 400 neurons. The palimpsest storage is plotted against number of patterns presented

Definition 1 (Relative/absolute palimpsest storage) *Store m patterns in the network. For each pattern, starting with the most recent and moving back through time, check whether it is recalled with an error within a given tolerance. Stop when one pattern fails this test. The total number of pattern recalled within the tolerance is called the palimpsest storage for m patterns. For absolute palimpsest storage, this tolerance level is zero. For relative palimpsest storage, the tolerance level is small, but non-zero.*

Here we make one assumption to speed up the computation of the relative capacity. We assume that if the stored pattern has only a small number of unstable bits, then it is within the direct attraction basin of some stable point. This assumption is often fair. It fails when flipping one bit of the pattern induces instability in a whole number of other bits which were previously stable. But the affect of a single bit change is usually relatively small [7].

The benefit of this assumption is that we can test the number of unstable neurons when the stored pattern is presented, rather than searching for network fixed points. We take a tolerance level of 5 percent, So 95 percent of the bits of the nearest fixed point must be correct.

Figure 1 gives the palimpsest storage at different memory loadings. The network recalls all patterns until capacity is reached. The storage level then decreases, tending to the palimpsest capacity of the network.

In order to find the palimpsest capacity, the palimpsest storage is averaged for a number of high memory loadings for different network sizes. Figure 2 plots the palimpsest capacity as the number of neurons in the network increases. The straight line on the graph corresponds to $0.25n$, and gives a lower bound to the capacity scaling of the network. The dotted line gives the $0.05n$ capacity of the

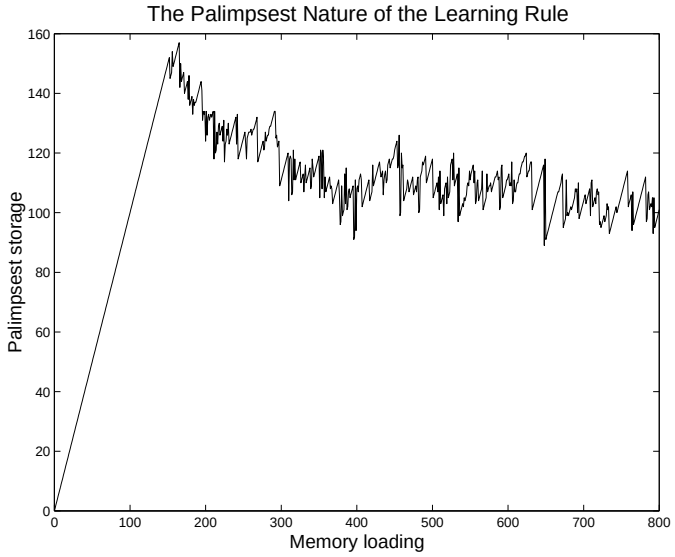

Figure 1: The palimpsest storage at different memory loadings

standard palimpsest rules.

Because of the assumptions made in calculating the relative capacities, a plot of the absolute palimpsest capacity is also given on the same graph. This gives a definite lower bound for the relative capacity, as well as being interesting in its own right.

5 How the learning rule works

The recursive nature of the learning rule makes it hard to analyse, but it is possible to illustrate why the learning rule acts as a palimpsest. We demonstrate that it acts as an iterated function sequence (IFS) with probabilities.

Definition 2 (IFS with probabilities) *An IFS with probabilities, denoted by* $IFS(X, d; f_1, f_2, \ldots, f_r; p_1, p_2, \ldots, p_r)$*, consists of a complete metric space (X,d) together with a finite set of contraction mappings* $\{f_1, f_2, \ldots, f_r\}$ *and a set of*

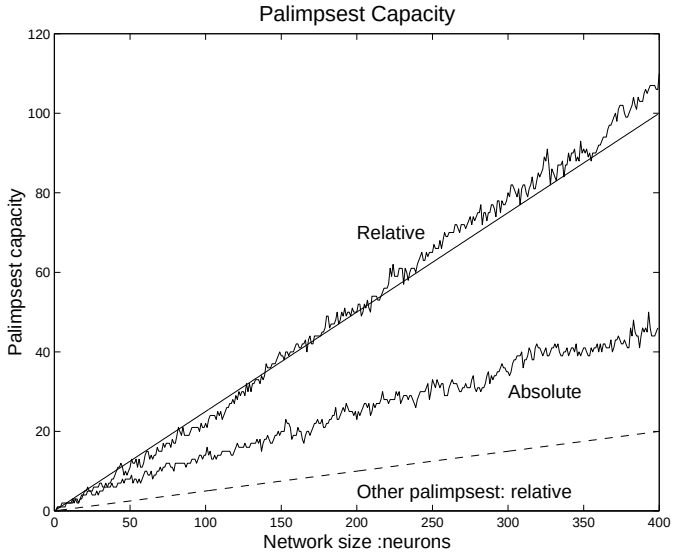

Figure 2: The palimpsest storage at different memory loadings

probabilities $\{p_1, p_2, \ldots, p_r\}$ *where* p_i *is the probability of choosing transformation* f_i.

IFSs are important because they have a unique invariant fractal measure. The importance of an invariant fractal distribution for palimpsest rules is elaborated in [3] and [11].

The learning scheme (2) is now reformulated in IFS terminology. Let X be the space of weight (i.e. zero diagonal symmetric) matrices, with the Frobenius metric

$$d(W^1, W^2) = \sum_{i,j \neq i}^{n} (w_{ij}^2 - w_{ij}^1)^2 \tag{3}$$

where we make the fact that the weight matrix has zero diagonal explicit in the summation.

Define $f_{\boldsymbol{\xi},\boldsymbol{\xi}'}(W)$ by the action of the update rule (2) on W as two distinct patterns $\boldsymbol{\xi}, \boldsymbol{\xi}'$ are stored consecutively. Let $p(\boldsymbol{\xi}, \boldsymbol{\xi}') = 1/(2^n(2^n - 1))$ for all

possible distinct $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$.

Proposition 1 *$(X, d, \{f(\boldsymbol{\xi}, \boldsymbol{\xi}')\}; \{p(\boldsymbol{\xi}, \boldsymbol{\xi}')\})$ is an IFS with probabilities, and defines the learning rule (2) where identical pattern pairs are disallowed.*

Proof Consider two possible weight matrices W^1 and W^2. The Frobenius distance between these two matrices is given by (3).

Now consider what happens to this distance as the two matrices are updated on the arrival of pattern $\boldsymbol{\xi}$. Let W^{1new} and W^{2new} be these updated weight matrices. Then

$$\sum_{i,j\neq i}^{n}(w_{ij}^{2new} - w_{ij}^{1new})^2 = \sum_{i,j\neq i}^{n}(w_{ij}^2 - w_{ij}^1)^2 \tag{4}$$

$$+ \quad \frac{1}{n^2}\sum_{i,j\neq i}^{n}\left[\sum_{k\neq i}^{n}(w_{ik}^2 - w_{ik}^1)\xi_k\xi_j + \sum_{k\neq i}^{n}\xi_i\xi_k(w_{ik}^2 - w_{ik}^1)\right]^2 \tag{5}$$

$$- \quad \frac{2}{n}\sum_{i,j\neq i}^{n}(w_{ij}^2 - w_{ij}^1)\left[\sum_{k\neq i}^{n}(w_{ik}^2 - w_{ik}^1)\xi_k\xi_j + \sum_{k\neq j}\xi_i\xi_k(w_{kj}^2 - w_{kj}^1)\right] \tag{6}$$

The part of this equation labelled (5) simplifies to

$$\frac{2(n-1)}{n^2}\sum_{i}^{n}\left[\sum_{k\neq i}^{n}(w_{ik}^2 - w_{ik}^1)\xi^k\right]^2 + \frac{2}{n^2}\left[\sum_{i,k\neq i}^{n}\xi_i(w_{ik}^2 - w_{ik}^1)\xi_k\right]^2 \tag{7}$$

where we use $\sum_{j\neq i}\xi_j\xi_j = n-1$. Clearly the second part of the RHS of (7) is smaller than the first part by the triangle inequality, and both parts are positive (they are sums of square terms). Hence (5) is less than

$$\frac{4(n-1)}{n^2}\sum_{i}^{n}\left[\sum_{k\neq i}^{n}(w_{ik}^2 - w_{ik}^1)\xi^k\right]^2$$

On the other hand the part-equation labelled (6) simplifies to

$$-\frac{4}{n}\sum_{i}^{n}\left[\sum_{k\neq i}^{n}(w_{ik}^2 - w_{ik}^1)\xi_k\right]^2$$

Putting all this together we get

$$\sum_{i,j\neq i}^{n}(w_{ij}^{2new}-w_{ij}^{1new})^2<\sum_{i,j\neq i}^{n}(w_{ij}^{2}-w_{ij}^{1})^2-\frac{2}{n^2}\sum_{i}^{n}\left[\sum_{k\neq i}^{n}(w_{ik}^{2}-w_{ik}^{1})\xi_k\right]^2$$

Now we see that on the whole the distance between two weight matrices is reduced by putting them through the update equation. The only exception to this is if each row of $w_{ij}^2 - w_{ij}^1$ lies in a hyperplane perpendicular to the new pattern ξ. For now, let us constrain $w_{ij}^2 - w_{ij}^1$ to have some row a for which

$$\sum_{j}(w_{aj}^{2}-w_{aj}^{1})\xi_j\geq\alpha\sum_{i,j\neq i}^{n}(w_{ij}^{2}-w_{ij}^{1})^2 \tag{8}$$

for α small and positive. This excludes matrices for which all row vectors are within a small angle of the aforementioned hyperplane. Then

$$\sum_{i,j\neq i}^{n}(w_{ij}^{2new}-w_{ij}^{1new})^2\leq(1-\alpha)\sum_{i,j\neq i}^{n}(w_{ij}^{2}-w_{ij}^{1})^2 \tag{9}$$

If on the other hand the weight matrix difference does not satisfy the constraint (8) for this ξ, then it will undergo only a small perturbation when updated by (2), and will certainly satisfy the constraint for a new and different pattern $\boldsymbol{\xi}'$ which arrives next. Hence after the weight matrices have been updated by two different incoming patterns, equation 9 is satisfied for all w_{ij}^1 and all w_{ij}^2. Therefore we have a contraction mapping on the space of weight matrices. This enables us to define an IFS with probabilities.

It is easiest if we restrict the incoming patterns so that each pair of incoming patterns cannot be identical (or opposite). This constraint is negligible for large network sizes. Then for each pattern pair $s = (\boldsymbol{\xi}, \boldsymbol{\xi}')$, we have a contraction mapping, denoted ϕ_s, and a probability $p_s = 1/(2^n(2^n-1))$, and so we have an IFS with probabilities $(\phi_1, \phi_2, \ldots; p_1, p_2, \ldots)$ on the space of matrices with a Frobenius metric.

The contraction mapping is given by the application of the learning rule, and the probabilities are the probabilities of choosing a given pattern pair for zero mean ± 1 patterns chosen without replacement. Hence this IFS is isomorphic to the learning rule where identical pattern pairs are disallowed. $\Box$

The constraint on pattern pairs is negligible for large networks, and in fact an identical second pattern will serve only to reinforce that same pattern. Hence we have shown that the learning rule acts as an IFS.

Consider two points W^1, W^2 in weight space chosen with respect to the invariant measure of the learning rule. This corresponds to choosing two points with different histories. The distance between these two weight matrices measures the difference in the recall performance of the Hopfield networks corresponding to each weight matrix.

Now the learning rule for any given pattern acts as a contraction mapping in the space of weight matrices. Hence as each weight matrix is updated with incoming patterns, W^1 and W^2 are mapped to points closer to one another. In other words the effect of the different histories is reduced as new training patterns arrive: the learning rule enables gradual forgetting of the past, favouring more recent patterns. This is an indication of the palimpsest effect of the learning rule.

The above result also helps elucidate why the learning rule has such a high capacity. The decay rate for each row of the weight matrix depends on its dot product with the training pattern. Hence each row vector is pushed towards the hyperplane perpendicular to the training vector before the Hebbian term is added. This reduces the interference of old patterns with the recall of the new

pattern (recall depends on the same dot product). One consequence of this is that there is no reduction in the information stored in the components of the matrix rows which are perpendicular to the training pattern. This contrasts with standard palimpsest rules where exponential decay occurs to all elements of the weight matrix indiscriminately.

6 Conclusion

The new learning rule introduced in this paper provides a new approach to producing palimpsest learning schemes. Instead of forcing the weight matrix to exhibit exponential decay, the patterns are stored in the weight matrix in the normal way. However at each stage the components of the weight matrix which interfere with the recall of that pattern are significantly reduced. Hence old memories are not forced to decay if it is not necessary.

The result of this is a palimpsest memories with capacities larger than standard non-palimpsest learning schemes such as the Hebb rule. This improvement is at least a factor of five improvement over previous palimpsest rules.

6.1 Acknowledgements

The author would like to thank Romain Valabregue and Philippe DeWilde for helpful discussions.

References

[1] 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.

[2] J. P. Nadal, G. Toulouse, J. P. Changeux, and S. Dehaene. Networks of formal neurons and memory palimpsests. *Europhysics Letters*, 1:535–542, 1986.

[3] U. Behn, J. L. van Hemmen, R. Kuhn, A. Lange, and V. A. Zagebnov. Multifractality in forgetful memories. *Physica D*, 68:401–415, 1993.

[4] M. Mezard, J. P. Nadal, and G. Toulouse. Solvable models of working memories. *Journal de Physique*, 47:1457–1462, 1986.

[5] A. J. Storkey. Increasing the capacity of the Hopfield network without sacrificing functionality. In W. Gerstner, A. Germond, M. Hastler, and J Nicoud, editors, *ICANN97: Lecture Notes in Computer Science 1327*, pages 451–456. Springer-Verlag, 1997.

[6] A. J. Storkey and R. Valabregue. A new Hopfield learning rule with high capacity storage of correlated patterns. *Electronics Letters*, 33(21):1803–1804, 1997.

[7] A. J. Storkey and R. Valabregue. The basins of attraction of a new Hopfield learning rule. Preprint, 1998.

[8] G. Parisi. A memory which forgets. *Journal of Physics A: Mathematical and General*, 7:L617–L620, 1986.

[9] J. L. van Hemmen, G. Keller, and R. Kuhn. Forgetful memories. *Europhysics Letters*, 5(7):663–668, 1988.

[10] D. J. Amit and S. Fusi. Learning in neural networks with material synapses. *Neural Computation*, 6:957–982, 1994.

[11] P. J. Potts. The storage capacity of forgetful neural networks. Master's thesis, Department of Computing, Imperial College, 1995.