An Incremental Bayesian Learning Rule for Palimpsest Memory

Anders Sandberg   Örjan Ekeberg
Anders Lansner
SANS, Royal Institute of Technology
Stockholm, Sweden

Abstract

A continuously learning system needs to gradually forget old information in order to avoid catastrophic forgetting. This can be achieved by allowing new information to overwrite old, a so-called palimpsest memory. This paper describes an incremental learning rule based on the Bayesian confidence propagation network that has palimpsest properties when applied to a Hopfield-like neural network.

1 Introduction

A learning system interacting with the real world has to adapt to changing conditions and new data as they appear. It can not afford to ignore them, but at the same time the continuing inflow of data threatens its stability. In order to avoid information overload a forgetting mechanism is necessary.

The Hopfield model has been widely studied as a general autoassociative memory, but it suffers from a limited capacity of only $0.138N$ uncorrelated patterns ξ^n that can be stored at the same time with only minor interference [1]. As the number of stored patterns increases above this limit all the stored patterns become unstable and the network becomes useless. This breakdown is unfortunate, since most learning systems need to continually learn, i.e. to store new information for future use as the information becomes available, and can not ignore the information as it arrives.

To ameliorate this palimpsest memories have been developed [2, 3] where new patterns are stored on top of older ones which are gradually overwritten and become inaccessible. This system retains the capacity to learn, at the price of forgetfulness.

Several palimpsest models have been developed. A marginalist learning paradigm has been proposed by Nadal, Toulouse and Changeaux [3], where the acquisition intensity is tuned to the present noise level in the weight matrix. Parisi has described learning within bounds, where the synaptic weights are bounded within an interval [2]. Srivastava, Vipin and Granato proposed a variant of learning within bounds where the weight is reset to zero when reaching the bound [4].

This paper presents an incremental learning rule based on Bayesian confidence propagation networks that results in a palimpsest memory. It is a Hebbian rule that reinforces connections between simultaneously active units and weakens or makes connections inhibitory between anti-correlated units. The synaptic strengths will be estimates of the conditional probabilities of the units firing together, based on incremental learning over time.

2 Bayesian learning

The activity of each unit in a Bayesian neural network can be said to correspond to the confidence in a certain property or membership in a class [5]. Each unit computes its support as the sum of the confidence of all other active units in the net times the conditional probabilities that they imply the unit in question, plus a bias based on its a priori probability of being active. The support is a measure of how desirable it is that that particular unit is active given the context formed by the other units. The activities of the units can then be updated using a relaxation scheme (for example by sequentially changing the units with the largest discrepancies between their activity and their support) or by continuously spreading activation, moving it towards a more consistent state.

The learning rule can be derived from Bayes rule[6]:

$$P(j|A) = P(j)\prod_{i\in A}\frac{P(j|i)}{P(j)} = P(j)\prod_{i\in A}\frac{P(i,j)}{P(i)P(j)} \tag{1}$$

where A is a set of independent events and $P(j|A)$ is the conditional probability that event j will happen given this information. If we take the logarithm of this formula we get:

$$\begin{aligned}\log(P(j|A)) &= \log(P(j)) + \sum_{i\in A}\log\left(\frac{P(i,j)}{P(i)P(j)}\right) &(2)\\ &= \log(P(j)) + \sum_{i}\log\left(\frac{P(i,j)}{P(i)P(j)}\right)o_i &(3)\end{aligned}$$

where $o_i = 1$ if $x \in A$ and zero otherwise. The last equation is similar to the usual form of neural network action, with o_i as the output of unit i, $\log(P(j))$ as a bias term, and $\log\left(\frac{P(i,j)}{P(i)P(j)}\right)$ as weights. If we want the unit output to correspond to the posterior probability of q we can use an exponential transfer function.

Although this is based on the assumption that the events in A are independent, in practice the network works well even when there are correlations between them. Extended variants of the network have been developed that can deal with correlated events [7].

2.1 Incremental Bayesian Learning

To derive the connection weights estimates of the probabilities $P(i)$, $P(j)$, $P(i,j)$ have to be made. If the training data is already present they can be calculated by counting the number of occurrences of events i, j and ij in the training data. If the network is intended to learn during operation, it has to continuously derive estimates and update the weights correspondingly. It is also undesirable to use the counting approach, since if the input is non-stationary old states are best forgotten.

The incremental Bayesian learning rule [8] approximates the the probabilities given moving averages of unit activities. The probability $P(j)$ that unit j is active can be estimated from the current unit activity $S_n(j)$ at time-step n as

$$\widetilde{P}_{n+1}(j) = \widetilde{P}_n(j) + \frac{S_n(j) - \widetilde{P}_n(j)}{\tau} \tag{4}$$

where τ is a time constant. An estimate $\widetilde{P}_n(i)$ of $P(i)$ can be calculated correspondingly.

The probability of coactivation $P(ij)$ can be similarly estimated

$$\widetilde{P}_{n+1}(ij) = \widetilde{P}_n(ij) + \frac{S_n(i)S_n(j) - \widetilde{P}_n(ij)}{\tau} \tag{5}$$

These estimates can be combined into connection weights which are updated for each time-step.

$$w_{nij} = \log\left(\frac{\widetilde{P}_n(ij)}{\widetilde{P}_n(i)\widetilde{P}_n(j)}\right) \tag{6}$$

Since the weights of the network depend more on recent data than on old data, it appears likely that a Hopfield-like network with the above learning rule would exhibit palimpsest properties. In the extreme case of a very short time constant only the latest pattern would be retained.

2.2 An Attractor Network with Incremental Bayesian Learning

Unit states x_i are updated asynchronously according to

$$x_i(n+1) = \operatorname{sgn}\left(\beta_i + \sum_{j=1}^{N} w_{ij}x_j(n)\right) \tag{7}$$

where the weights w_{ij} and bias β_i are calculated using a discrete incremental Bayesian learning rule:

$$P_i(n+1) = \alpha((1-P_0)x_i + P_0) + (1-\alpha)P_i(n+1) \tag{8}$$

$$P_j(n+1) = \alpha((1-P_0)x_j + P_0) + (1-\alpha)P_j(n+1) \tag{9}$$

$$P_{ij}(n+1) = 2\alpha((1-P_0^2)x_ix_j + P_0^2) + (1-2\alpha)P_{ij}(n+1) \tag{10}$$

$$w_{ij} = \log\left(\frac{P_{ij}(n)}{P_i(n)P_j(n)}\right) \tag{11}$$

$$\beta_i = \log(P_i(n)) \tag{12}$$

α is the inverse time constant, regulating the speed of learning and unlearning. Note that $P_{ij}(n)$ has a faster adaption than the other two state variables. By setting α temporarily to zero the network activity can change with no corresponding weight changes. P_0 corresponds to a weak background level of activity in the network and prevents P_i, P_j and P_{ij} from underflow which would produce an infinite negative weight. The logarithm in equation 11 acts as a soft upper bound on the weights.

If the activities x_i and x_j have probability $P(i)$, $P(j)$ and covariance C_{ij} the expectation of the weight becomes

$$E(w_{ij}) \approx \log\left(\frac{C_{ij}}{P(i)P(j)} + 1\right)$$

for small values of P_0.

A version with a slightly modified update rule for sparse patterns was also implemented.

3 Results

Figure 1 shows a comparison between a traditional sparse Hopfield network and a network using the incremental Bayesian learning rule.

The performance for a given pattern ξ^i was measured as the percentage of perturbed patterns $\xi^i + \delta$ (where δ is a normally distributed vector with variance 0.3) which were correctly recalled after relaxation (to a tolerance of 1.5%); this measure is shown for the first learned pattern ξ^1 and the latest ξ^n.

As can be seen the Bayesian network avoids catastrophic forgetting by gradually forgetting the oldest pattern. Figure 2 shows this in more detail. The forgetting does not occur immediately, the pattern is stored quite well until a certain number of interfering patterns have been stored and then gradually fades.

The capacity of the network (measured as the numbers of pattern that can be stored before recall performance of the first pattern degrades to 50%) scales roughly as as α^{-1} down to $\alpha \approx 10^{-4}$, i.e directly proportional to the time constant of the network (figure 3). For longer time constants the maximal capacity for this network is reached, roughly $N/\log(N)$. Fast learning overwrites memories quickly, while slow learning allows old memories to be retained even as more are added.

4 Conclusion

In this paper, I have demonstrated a incremental Bayesian learning rule that produces palimpsest memory properties in a simple Hopfield-like attractor network. It is able to learn new patterns while retaining old patterns fairly clearly until they reach a certain age, deals well with sparse patterns and has a capacity proportional to the time constant of learning.

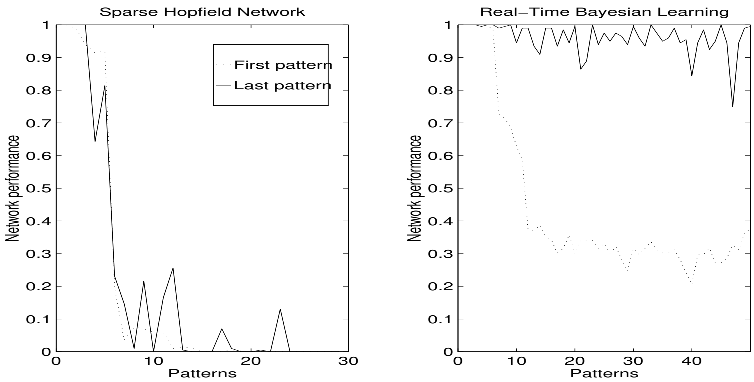

Figure 1: Comparison of the sparse Hopfield learning rule and the Bayesian learning rule for sparse random patterns ($\rho = 0.05$). For the Bayesian learning the input pattern was injected into the network for 20 time-steps followed by 20 time-steps of "rest" before the next pattern presentation. $N = 100$, $\alpha = 0.01$.

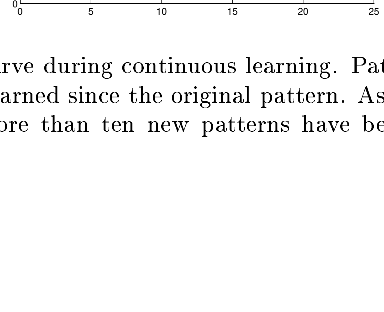

Figure 2: Forgetting curve during continuous learning. Pattern recall as a function of number of patterns learned since the original pattern. As can be seen the pattern is recalled well until more than ten new patterns have been learned, and then is gradually forgotten.

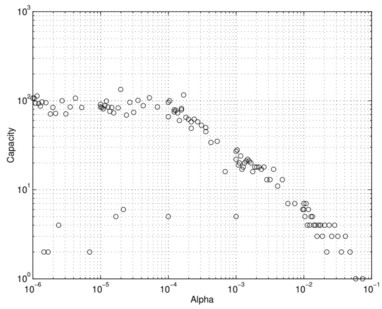

Figure 3: Network capacity as a function of α. 100 units, ten randomly active units in each pattern.

Future work includes developing a two-step learning rule better able to handle spurious correlations in the data and temporal association, and analysis of the relationship between learning rate, storage capacity and stability.

References

[1] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation.* Number 1 in Lecture Notes. Addison-Wesley, Santa Fe Institute for studies in the sciences of complexity, 1991.

[2] Giorgio Parisi. A memory which forgets. *J. Phys. A: Math. Gen*, 19:L617–620, 1986.

[3] J.P Nadal. Networks of formal neurons and memory palimpsests. *Europhysics Letters*, 1(10):535–542, 1986.

[4] Vipin Srivastava, Meena Vipin, and Enzo Granato. Recall of old and recent information. *Network*, 9:159–166, 1998.

[5] Anders Lansner and Örjan Ekeberg. A one-layer feedback artificial neural network with a bayesian learning rule. 1(1):77–87, 1989.

[6] Anders Lansner and Örjan Ekeberg. An associative network solving the "4-Bit ADDER problem". In Maureen Caudill and Charles Butler, editors, *IEEE First International Conference on Neural Networks*, pages II–549, San Diego, CA, June 21–24 1987.

[7] Anders Holst. *The Use of a Bayesian Neural Network Model for Classification Tasks.* PhD thesis, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, September 1997. TRITA-NA-P9708.

[8] Örjan Ekeberg. An incremental learning rule for artificial neural networks and its implementation with low precision fixed point arithmetics. Draft, March 1991.