An incremental Bayesian learning rule NADA Technical report TRITA-NA-P9908

A. Sandberg†, A. Lansner†, K. M. Petersson‡and Ö. Ekeberg†

† Department of Numerical Analysis and Computing Science, Royal Institute of Technology, 100 44 Stockholm, Sweden
‡ PET/Cognitive Neurophysiology Research Group, Karolinska Hospital, Stockholm, Sweden

Abstract. A realtime online 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, as in a so-called palimpsest memory. This paper describes an incremental learning rule based on the Bayesian confidence propagation neural network that has palimpsest properties when employed in an attractor neural network. The network does not exhibit catastrophic forgetting, has a capacity dependent on the learning time constant and exhibits decreasing convergence speed for older patterns.

1. Introduction

In contrast to the artificial learning situation often faced by artificial neural networks (ANN), real world learning presents a next to unlimited number of learning examples, potentially surpassing the storage capacity of the learner by orders of magnitude. An additional complication is that the world may be non-stationary, partly due to the changing behavior of the system and its interaction with the environment. In general, it is critical to give priority to the retention of recent information of a kind that is relevant to the operation for a capacity limited real world learning system. This may occur at several different time-scales, as in for example short-term and long-term memory. These are fundamental characteristics of existing biological learning and memory systems which are also critical for advanced artificial learning systems.

Auto-associative attractor ANNs, like for example early binary associative memories and the more recent Hopfield net, have been proposed as models for biological associative memory [30, 16, 1]. They can be regarded as formalizations of Donald Hebb's original ideas of synaptic plasticity and emerging cell assemblies [12]. A number of psychological memory and Gestalt perception phenomena have been modelled based on such network models [7, 25]. Simulations have indicated that a network of cortical pyramidal and basket cells can operate as an attractor network and its connectivity structure captures many aspects of cortical functional architecture [10, 9, 6, 5].

The standard correlation based learning rule for attractor ANN suffer from catastrophic forgetting, that is, all memories are lost as the system gets overloaded. To cope with this situation Nadal, Toulouse and Changeaux [22] proposed a so called "marginalist" learning paradigm where the acquisition intensity is tuned to the present level of crosstalk “noise” from other patterns. This makes the most recently learned pattern the most stable; new patterns are stored on top of older ones, which are gradually overwritten and become inaccessible, a so-called “palimpsest memory”. This system retains the capacity to learn at the price of forgetfulness.

Another smoothly forgetting learning scheme is “learning within bounds” (originally suggested by Hopfield [16]), where the synaptic weights w_{ij} are bounded $-A \leq w_{ij} \leq A$. The learning rule for training patterns ξ^n is

$$w_{ij}(n+1) = c\left(w_{ij}(n) + \frac{\xi_i^n \xi_j^n}{\sqrt{N}}\right)$$

where c is a clipping function

$$c(x) = \begin{cases} -A & x < -A \\ x & |x| < A \\ A & A < x \end{cases}$$

and the optimal capacity $0.05N$ is reached for $A \approx 0.4$ [23, 3]. For high values of A catastrophic forgetting occurs, for low values the network remembers only the last pattern. This implies a decrease in storage capacity from $0.137N$ of the standard Hopfield Hebbian rule; total capacity has been sacrificed for long term stability.

A learning rule for attractor networks derived from Bayes' rule [2] has previously been developed [17, 18, 19, 14]. It is a Hebbian rule that reinforces connections between simultaneously active units and weakens or makes connections inhibitory between anti-correlated units. This version of the learning rule is based on a probabilistic view of learning and retrieval, with input and output unit activities representing confidence of feature detection and posterior probabilities of outcomes, respectively. The synaptic strengths are based on the probabilities of the units firing together, estimated by counting occurrences in the training data. This learning rule gives a symmetric weight matrix, allowing for fixed point attractor dynamics (c.f. equation 3 and [16]). It also generates a proper balance between excitation and inhibition, avoiding the need for external means of threshold regulation. The update of the weights in the network resembles what has been proposed as rules for biological synaptic plasticity [20].

In this paper we demonstrate that by estimating the probabilities underlying synaptic modification by running averages instead of pre-calculated counters it is possible to derive a continuous, real-time Bayesian learning rule with palimpsest memory properties. The forgetfulness can conveniently be regulated by the time constant of the running averages. We evaluate this learning rule in the context of long-term and short-term memory of an attractor network. The consequences of a time dependent energy landscape in terms of convergence times are also investigated.

2. Bayesian Learning

Bayesian Confidence Propagation Neural Networks (BCPNN) [17, 18, 19, 14] are based on Hebbian learning derived from Bayes' rule:

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

where $A = \{i\}$ and $\{i \cup j\}$ are sets of independent and conditionally independent events or features that has occurred 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:

$$\begin{aligned}\log(P(j|A)) &= \log(P(j)) + \sum_{i\in A} \log\left(\frac{P(i,j)}{P(i)P(j)}\right) \\ &= \log(P(j)) + \sum_{i} \log\left(\frac{P(i,j)}{P(i)P(j)}\right) o_i \end{aligned} \tag{1}$$

where $o_i = 1$ if $i \in A$ and zero otherwise. The last equation is similar to the common form of neural network action ($x_j \leftarrow \beta_j + \sum w_{ij} f(x_i)$), and o_i can be identified as the output of unit i, the fact that event i has occurred or an inference that it has occured. Since inferences are uncertain, it is reasonable to allow values between zero and one, corresponding to different levels of confidence in i. $f(x)$, the transfer function, is a clipped exponential (see below). The term $\log(P(j))$ can be identified as a bias term, and the fractional expressions within the sum as weights:

$$\beta_j = \log(P(j)) \tag{2}$$

$$w_{ij} = \begin{cases} \log\left(\frac{P(i,j)}{P(i)P(j)}\right) & i \neq j \\ 0 & i = j \end{cases} \tag{3}$$

The network is used with an encoding mode where the weights are set and a retrieval mode where inferences are made. Input to the network can be introduced by setting the activations of the relevant units (representing known events or features). As the network is updated the activation spreads, creating a posteriori inferences of the likelihood of other features. These inferences can be used as input for further (informal) inferences, allowing the net to relax towards a stable and consistent state.

One way of measuring the confidence in the presence of event j is $P(j|A)$, which can be retrieved from unit activation by the help of an exponential transfer function (the base for the logarithms and transfer function is irrelevant; for performance reasons the natural logarithm is used here)

$$\Theta(x) = \begin{cases} \exp(x) & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{4}$$

We introduce the update rule

$$o_j \leftarrow \Theta(\beta_j + \sum_i w_{ij} o_i) \tag{5}$$

where updates can be synchronous, asynchronous or neurodynamic (as in equation 12).

Since the weight matrix is symmetric an energy function will be defined, and convergence to a fixed point is assured [13]. When initialized with a pattern of confidence and allowed to relax, the network state tends towards a consistent "low energy" state.

Although the learning rule is based on the assumption that the events in A are independent, in practice the network works well even when there are some correlations between them [14]. Extended variants of the network have been developed that can deal with more strongly correlated events [14].

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 as C observed pattern vectors ξ^k with component events ξ_i^k the estimates can be easily calculated by counting the number of occurrences of events i, j and ij in the training data.

$$\begin{aligned} c_i &= \sum_{k=1}^{C} \xi_j^k \\ c_{ij} &= \sum_{k=1}^{C} \xi_i^k \xi_j^k \end{aligned}$$

giving probability estimates $\hat{p}_i = c_i/C$ and $\hat{p}_{ij} = c_{ij}/C$. Since the logarithm of these values will be used, special care has to be taken with counters that are zero. In practice the logarithm of zero is replaced with a number that is more negative than all the other values of biases or weights in the network [15].

The weights are set to

$$w_{ij} = \begin{cases} 0 & c_i = 0 \text{ or } c_j = 0 \text{ or } i = j \\ \log(1/C) & c_{ij} = 0 \\ \log\left(\frac{c_{ij}C}{c_i c_j}\right) & \text{otherwise} \end{cases} \tag{6}$$

and the biases to

$$\beta_i = \begin{cases} \log(1/C^2) & c_i = 0 \\ \log(c_i/C) & \text{otherwise} \end{cases} \tag{7}$$

On the other hand, if the data is arriving over time, the probability estimates and weights instead have to be estimated from the sequentially available data on-line. This may be handled by an incremental Bayesian learning.

2.1. Incremental Bayesian Learning

The original formulation of the BCPNN is based on estimating probabilities of events from a given training set, with no time dependence in the estimate: all data is weighted equally. A continuously operating network will need to learn during operation. For example it cannot rely on fixed probability estimates in a non-stationary environment. Instead it will need to continuously derive estimates of the rates of events, updating the weights correspondingly.

The input $X(t)$ to the network can be viewed as a stochastic process $X(t, \cdot)$ in discrete or continuous time ($t \in \mathbb{N}$ or $t \in \mathbb{R}$) with an underlying probability space (Ω, A, P) (the environment) to $\mathbb{B}^d$ (the corners of the d-dimensional unit cube $\{0,1\}^d$, representing which of the d features are present).

Let $X_i(t)$ be component i of $X(t)$, the observed input. Then we can define $P_i(t) = P[X_i(t) = 1]$ and $P_{ij}(t) = P[X_i(t) = 1, X_j(t) = 1]$. Equation 1 becomes

$$\log(P_{j|A}(t)) = \log(P_j(t)) + \sum_i \log\left(\frac{P_{ij}(t)}{P_i(t)P_j(t)}\right) o_i$$

In order to use this equation for a BCPNN $P_i(t)$ and $P_{ij}(t)$ need to be estimated given the information $\{X(t'), t' < t\}$. What we want is an estimate with the following properties: i) it will converge towards $P_i(t)$ and $P_{ij}(t)$ in a stationary environment, ii) it gives more weight to recent than remote information and iii) it smooths or filters out noise and adapts to longer trends in a non-stationary environment.

The incremental Bayesian learning rule proposed here estimates the rates (and hence the weights) using exponential smoothing of unit activities. The units are assumed to be clamped by the input as the learning takes place. The rate λ_i of unit i can be estimated from the current unit activity $o_i(n)$ at time n with the following estimator

$$\frac{d\Lambda_i(t)}{dt} = \frac{o_i(t) - \Lambda_i(t)}{\tau} \tag{8}$$

where τ is a suitable time constant.

The rate Λ_{ij} of coincident firing can be similarly estimated

$$\frac{d\Lambda_{ij}(t)}{dt} = \frac{o_i(t)o_j(t) - \Lambda_{ij}(t)}{\tau'} \tag{9}$$

where τ' is a time constant, possibly but not necessarily equal to τ.

These estimates can be combined into a connection weight which is updated over time:

$$w_{ij}(t) = \log\left(\frac{\Lambda_{ij}(t)}{\Lambda_i(t)\Lambda_j(t)}\right) \tag{10}$$

$$\beta_i(t) = \log(\Lambda_i(t)) \tag{11}$$

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 could exhibit palimpsest properties if τ has a proper value.

The probability estimates converge towards the correct values given stationary inputs for sufficiently large time constants. In practice it turns out that $\tau' = \tau/2$ minimizes transients in the weight values. Since Λ_{ij} has to balance the product of two factors, each with a time constant τ, it needs to have a half as long time constant. Since $\Lambda_i(t)$ and $\Lambda_{ij}(t)$ behave as exponential functions in the absence of input

$$w_{ij}(t) \approx \log\left(\frac{\Lambda_{ij}(0)e^{-t/\tau'}}{\Lambda_i(0)e^{-t/\tau}\Lambda_j(0)e^{-t/\tau}}\right) = \log\left(\frac{\Lambda_{ij}(0)}{\Lambda_i(0)\Lambda_j(0)}\right) + (2/\tau - 1/\tau')t$$

The second term represents a bias due to the different rates of decay in the estimates; in order to keep $w_{ij}(t)$ constant in the absence of input τ' has to equal $\tau/2$, which makes the term vanish.

In the absence of any information, there is risk for underflow in the calculations. The counting approach (equation 6 and 7) solves this by setting the values that would otherwise have been infinitely negative to a negative number of large magnitude. Another interpretation of this becomes possible in this model: a basic low rate $\lambda_0 \ll 1$ of events will be assumed to occur, a kind of background activity (noise) that is present regardless of outside signals. Events will occur at rates between λ_0 and 1. In the absence of signals $\Lambda_i(t)$ and $\Lambda_j(t)$ now converges towards λ_0 and $\Lambda_{ij}(t)$ towards λ_0^2, producing $w_{ij}(t) = 0$ for large t. The smallest possible weight value if the state variables are initialized to λ_0 and λ_0^2 respectively is $\log(4\lambda_0^2)$, and the smallest possible bias $\log(\lambda_0)$. The upper bound on the weights becomes $\log(1/\lambda_0)$. This learning rule is hence a form of learning within bounds, although in practice the magnitude of the weights rarely come close to the bounds.

3. A Bayesian Attractor Network with Incremental Learning

The learning rule (equations 8–11) of the preceding section can be used in an attractor network similar to the Hopfield model by combining them with an update rule similar to equation 5. 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 from other units) or by random or synchronous updating similar to an ordinary attractor neural network, moving it towards a more consistent state. This latter approach is used here.

The continuous time version of the update and learning rule takes the following form (the discrete version is just a discretization of the continuous version):

$$\begin{aligned}
\tau_o \frac{do_i}{dt} &= \Theta\left(\beta_i(t) + \sum_{j=1}^{N} w_{ij}(t) o_j(t)\right) - o_i(t) \\
\tau \frac{d\Lambda_i}{dt} &= [(1-\lambda_0) o_i(t) + \lambda_0] - \Lambda_i(t) \\
2\tau \frac{d\Lambda_{ij}}{dt} &= [(1-\lambda_0^2) o_i(t) o_j(t) + \lambda_0^2] - \Lambda_{ij}(t) \\
\beta_i(t) &= \log(\Lambda_i(t)) \\
w_{ij}(t) &= \log\left(\frac{\Lambda_{ij}(t)}{\Lambda_i(t)\Lambda_j(t)}\right)
\end{aligned} \tag{12}$$

where τ_o is the time constant of change in unit state and $\Theta(x)$ the transfer function from equation 4. As is shown in Appendix A it is possible to derive expressions in closed form of the Λ and derived variables.

$\alpha = 1/\tau$ is the inverse of the learning time constant; it is a more convenient parameter than τ. By setting α temporarily to zero the network activity can change with no corresponding weight changes, for example during retrieval mode.

In the rest of this paper λ_0 will be set to 0.001.

3.1. Single Synapse Behavior

We first study how the weight w_{ij} between units i and j changes with correlation of unit activities (the activities are simply set to zero or one, i.e the activity update rule is not used here).

The somewhat counter-intuitive behavior of w_{ij} can be seen in figure 1 for two correlated and two anti-correlated units. When both units are activated together for a period of time the connection is strengthened significantly but quickly decays to a long-lasting steady state. At the end of the steady state period the weight dips to the negative side and goes to zero; this occurs when the more slowly changing Λ_i and Λ_j factors catch up with the faster changing Λ_{ij} factor. If $\lambda_0 = 0$, the time after the end of stimulation to this event is

$$t = \frac{2\log\left(1-(1-\alpha)^V\right) - \log\left(1-(1-2\alpha)^V\right)}{\log(1-2\alpha) - 2\log(1-\alpha)}$$

where V is the time the two units were activated together (this can be derived by explicitly solving the recurrence formula for Λ_i and Λ_{ij} in the discrete time case). t scales roughly as $1/\alpha^2$.

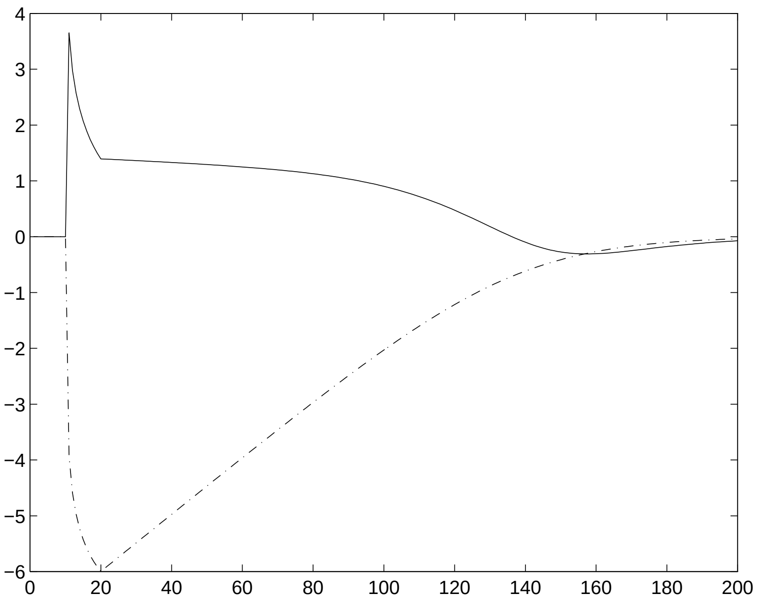


Figure 1. Weights between two units that are active together at $10 \leq t \leq 20$ (solid line) and when only one unit is active (dot-dash line) and $\alpha = 0.05$.

During training weights may change sign depending on whether the units they connect belong to the active units of different patterns (negative weight) or the same pattern (positive weight), as in figure 2.

3.1.1. Non-stationary Activities The above situation of two units firing together against a background of no activity is somewhat extreme given the assumptions of the model. A more canonical example of the changes in weights due to randomly firing units as well as the response to non-stationary activities can be seen in figure 3 where the two units are correlated, uncorrelated or anti-correlated with each other at different times. As can be seen, after a brief transient the weight moves towards a correct steady state value of log(2) as the units remain correlated. As the correlation vanishes the weight begins to move erratically due to spurious (anti) correlations, with a mean close to zero. When the units become anti-correlated the weight decreases practically linearly towards the baseline negative value of $\log(4\lambda_0^2) \approx -12.43$. Finally, when the correlations reappear, the weight quickly increases to the steady state value.

3.2. Network Behavior

In the following experiments the network was first trained by the presentation of the training patterns (the input patterns were each shown to the network for 10 units of time followed by 10 units of time of no input), followed by a testing period where no learning took place (α was set to zero). The update rule was not used during training, i.e the network state was clamped by the input. A discretized version of the update rule (12) was used.

Figure 4 shows the weight matrix after training with orthogonal patterns. The

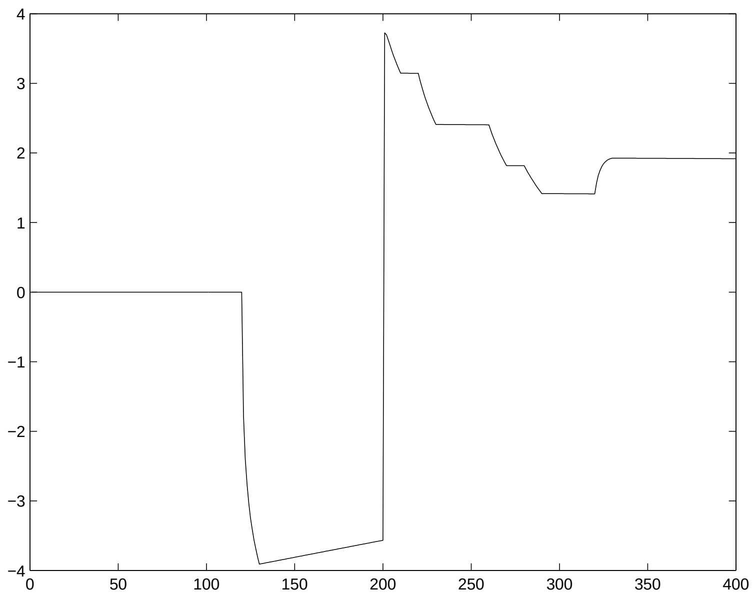


Figure 2. The weight between two units during the training of the network with sparse random patterns (10% activity, 10 time steps of presentation of each pattern followed by 10 units of decay with no activity) with $\alpha = 0.05$. At time 120 one of the units is recruited into a pattern while the other remains silent, causing a strong negative weight to develop. At time 200 both units become recruited by the same pattern, making the weight positive. It then remains positive for the rest of the run, despite occasional activations and coactivations.

ability to retrieve stored patterns depends on the time constant and the number of patterns. During fast learning (large α) only the latest patterns will be remembered even if the sequence of patterns is repeated several times.

The performance for a given pattern ξ_i was measured as the percentage of perturbed patterns which were correctly recalled after relaxation to a tolerance of 0.85 overlap (the overlap was defined to be $\xi \cdot o/||\xi||||o||$, the cosine of the angle between the pattern vector and the retrieved vector). In figure 5 the perturbed pattern is $\xi_i + \delta$ where δ is a normally distributed vector with variance 0.3, while in figure 6 and 7 two active units had been randomly moved. The different forms of perturbations do not show any qualitative change in network behavior.

Figure 5 shows a comparison between a traditional sparse Hopfield network and a network using our real-time Bayesian learning rule. As can be seen the learning rule avoids catastrophic forgetting by forgetting the oldest patterns while the recent patterns remain accessible. Figure 6 shows this in more detail. The forgetting does not occur immediately: for longer learning time constants the pattern is stored well until a certain number of interfering patterns have been stored, when it starts to gradually fade.

Figure 7 shows the number of retrievable patterns to the total set of training patterns as a function of α. In this experiment exposure to the training set of patterns

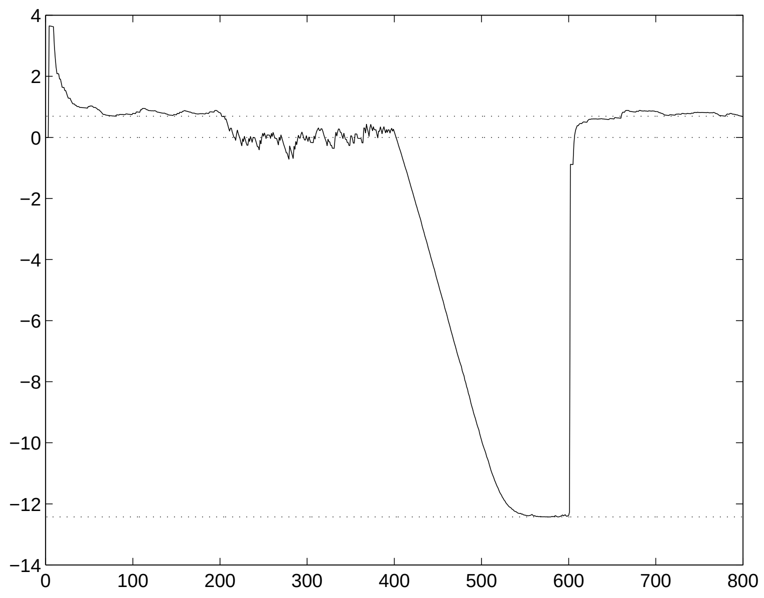


Figure 3. Weights between two units that are active 50% of the time, completely correlated for $0 \leq t \leq 200$, uncorrelated for $200 \leq t \leq 400$, completely anti-correlated for $400 \leq t < 600$ and finally correlated again. The dotted lines correspond to the predicted values $\log(2)$, 0 and $\log(4\lambda_0^2)$. In this run $\alpha = 0.05$.

was repeated a number of times until the weights were stationary. The number of retrieved patterns scales roughly as $O(\alpha^{-1})$, i.e directly proportional to the learning time constant, down to $\alpha \approx 10^{-3}$ for 50-pattern training sets and $\alpha \approx 10^{-4}$ for 500-pattern training sets in this run.

The capacity becomes maximal when the time constant equals the time to run through the entire training set ($\alpha_{\text{opt}} = 1/Vm$, where V is the time each pattern is shown and m is the size of the training set). If the time constant is larger than the repetition time the patterns will be mixed by the averaging process and few individual patterns can be retrieved well, unless the size of the set of training patterns is below the capacity of the traditional BCPNN (roughly $O((N/\log(N))^2)$ for sparse activation [18]). With a training set smaller than the capacity, the network would retrieve close to all patterns for time constants beyond this maximum.

If a stored memory cannot be retrieved, the activity in general tends towards a state corresponding to the a priori support $\exp(-\beta_i)$, the "a priori attractor"; this makes it possible to detect a retrieval failure by measuring the total activity level. We have observed few spurious states in these experiments. This is not surprising, considering that we are in the low load regime due to the short learning time constant. It has been shown that in this regime of Hopfield networks spurious states are relatively rare [13].

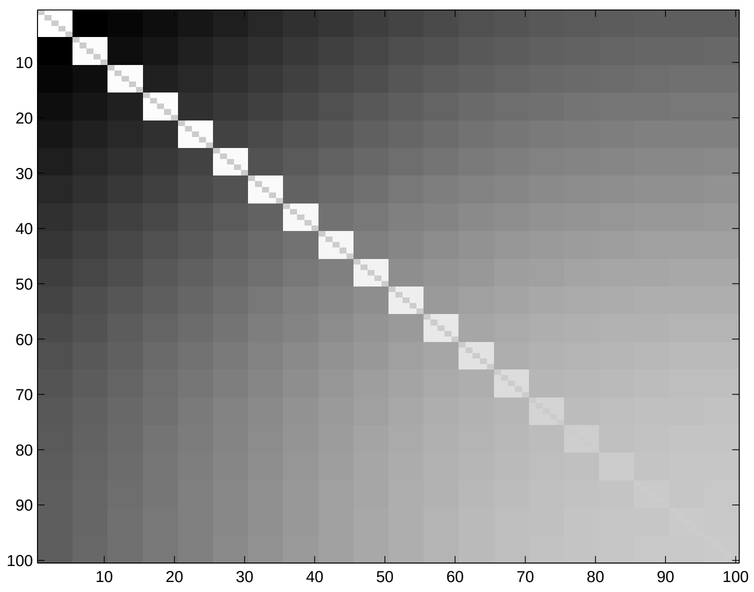


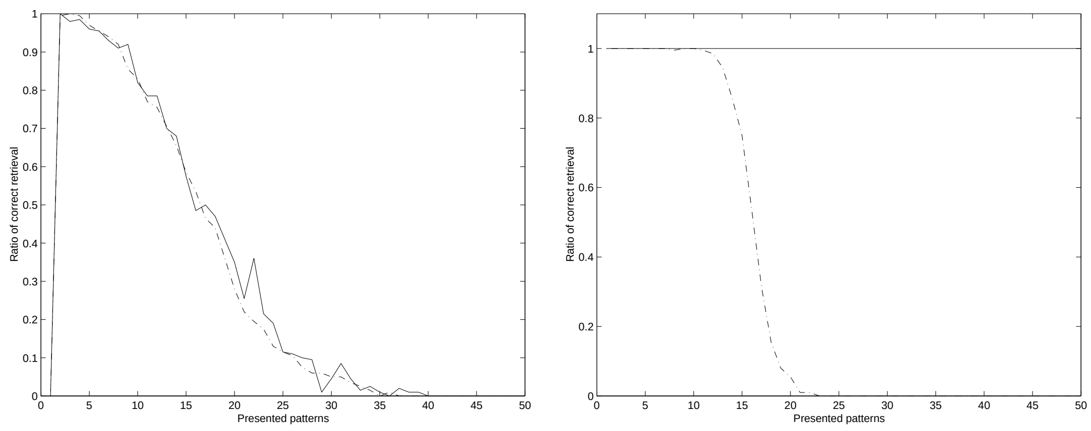
Figure 4. Weight matrix after sequential training with 20 orthogonal patterns consisting of five active units and 95 inactive. Dark = inhibitory connection, grey = weak connection, white = excitatory connection. The diagonal blocks consist of the excitatory connections between the units stimulated by the orthogonal patterns, which gradually fade as they age. The dark region consists of the inhibitory connections between mutually exclusive patterns. The strongest excitatory weight is 2.88, the strongest inhibitory -9.63. $\alpha = 0.2$.

3.3. Convergence Speed

Figure 8 shows convergence times for the network with an energy stopping criterion. The energy criterion stops the convergence when the energy $E = -(1/2)\sum_{ij} w_{ij} o_i o_j + \sum_i \beta_i o_i$ reaches below an arbitrary pre-determined level (in this case -100). The continuous update rule from equation 12 was used (the model was solved using Euler's method with step length $h = 0.1$, $\tau_o = 1$. Trials where convergence did not occur within 40 time steps were not counted. $N = 150$, $\alpha = 0.067$).

The energy criterion is interesting because it does not need knowledge of the target vector; it is in some sense an intrinsic stopping criterion rather than an extrinsic criterion. When retrieval occurs in a learning system only partial knowledge of the target is available in the form of cues, and for recognition tasks (where full retrieval is not necessary) it is more useful to get a response quickly than to wait for full convergence before reacting. Similar results can be obtained with a criterion that stops when the Euclidean norm of the change of state becomes smaller than a threshold.

Using this learning rule, as more patterns are learned, the basins of attraction of old patterns become more shallow, eventually disappearing altogether. This results in a change in convergence speed if an energy criterion is used. There is both a difference between patterns, the latest patterns are completed faster than older patterns, and a linear increase in the convergence time between networks where few patterns have



(a) Summing (b) Incremental

Figure 5. Comparison of the summing BCPNN learning rule and the incremental learning rule for sparse random patterns ($\rho = 0.1$). Solid line: latest learned pattern, dash-dotted line: first learned pattern. $\alpha = 0.01$ in the incremental case. Network size 100 units. The ratio of correctly retrieved patterns is shown for the first learned pattern ξ_1 and the latest ξ_n.

been learned and networks where the capacity has been reached (figure 8 and 9).

Convergence is slowed as the initial activation pattern starts farther and farther from an attractor. Figure 10 shows convergence times when the network is started with a mixture between two patterns plus noise. The noise is necessary in order to break the symmetry; in the absence of noise the network converges to a mixed equilibrium when given a perfectly 50% mixed input. For the two most recently learned patterns the convergence time is maximal at a 50% mixture (this is practically identical to the result in [18] for the non-incremental Bayesian learning rule). When interpolated between a recent and an old memory the maximum is moved away from the recent memory, a sign that the old memory has a smaller and weaker attractor than the newer memory. It can, however, still be retrieved given a similar enough cue.

When stimulated with noise (randomly activated units with the same density as the patterns) the network either converges to one of the learned patterns or to the a priori attractor state. The probability of convergence towards a certain memory decreases with its age (figure 11). This is in accordance with the results on learning within bounds due to Geszti and Pázmándi [8]. They found that as more patterns were stored thebasins of attraction of old patterns were more and more flattened energy-wise, and relaxation tended to move the system to a deeper attractor (i.e. a recent pattern).

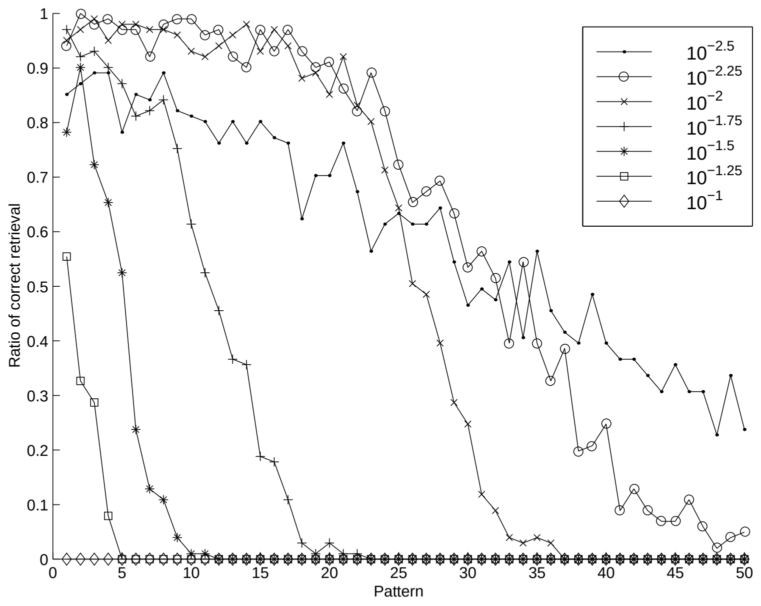


Figure 6. Forgetting curve during continuous learning. Pattern recall as a function of list position for different values of α (see legend) in a network with 100 units. Recall estimated as the frequency of retrieval with overlap greater than 0.85 150 iterations after a presentation of a pattern where two active units had been randomly moved. Random patterns, 10% activation, 20 timesteps presentation followed by 20 timesteps of no input.

4. Discussion

We have proposed and characterized a continuous, real-time extension of a previous Bayesian learning rule [18]. The new rule allows for continuous learning from a stream of examples without leading to catastrophic forgetting. Instead, old information is gradually forgotten and only the most recent examples are retained, as in a palimpsest memory. The memory capacity increases linearly with the learning time constant up to a limit where it becomes equal to the standard summating BCPNN. This means that by setting the size of the network and the learning time constant the memory capacity can be regulated. The time of convergence to a memory state depends both on the age of the memory and the load on the network.

The learning rule proposed is in some sense similar to marginalist learning [22], where the new patterns are exponentially amplified. In our case the past instead decays exponentially which is more biologically plausible. Moreover, one could readily conceive of ways in which the moving averages employed could be realized by biological synapses. The learning rule is Hebbian, and can exhibit a graded behavior with multiple synapse activations as well as a more step-like behavior for single synapse activation similar to experimental observations in LTP [24].

The original BCPNN learning rule is interesting in that it is based on probabilities and statistics rather than being a standard Hebbian outer product rule. Furthermore, palimpsest memories like learning within bounds [16, 23] ignore new information

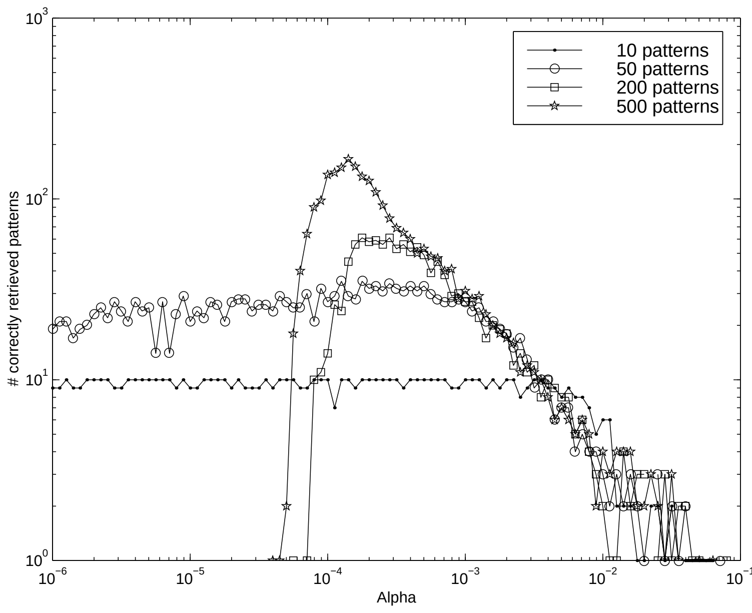


Figure 7. Number of correctly retrieved patterns as a function of α for different sizes of the training set. 100 units, ten randomly active units in each pattern, 10, 50, 200 and 500 patterns in the training set. Successful retrieval is defined as in figure 6.

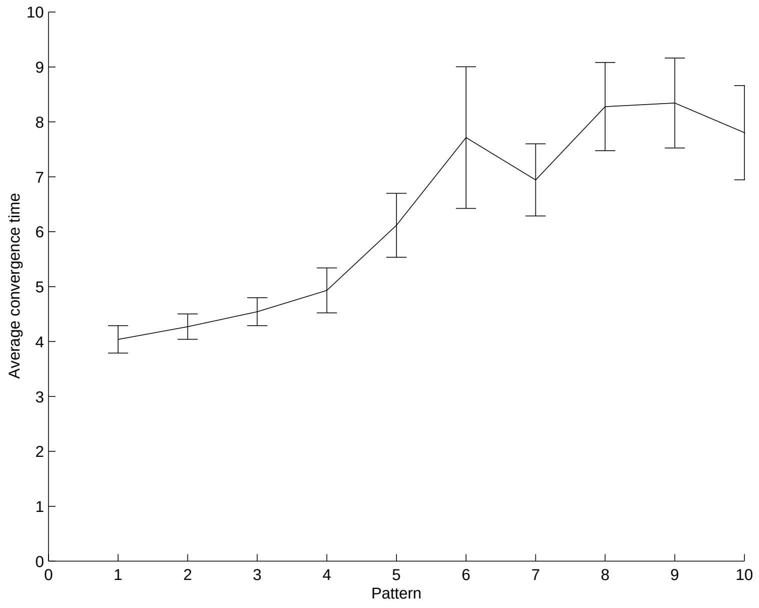


Figure 8. Convergence time for retrieval of stored patterns at full memory load. Pattern 1 is the most recently learned pattern. The line represent the mean of the convergence times. Ten patterns, $\alpha = 0.067$, $N = 100$.

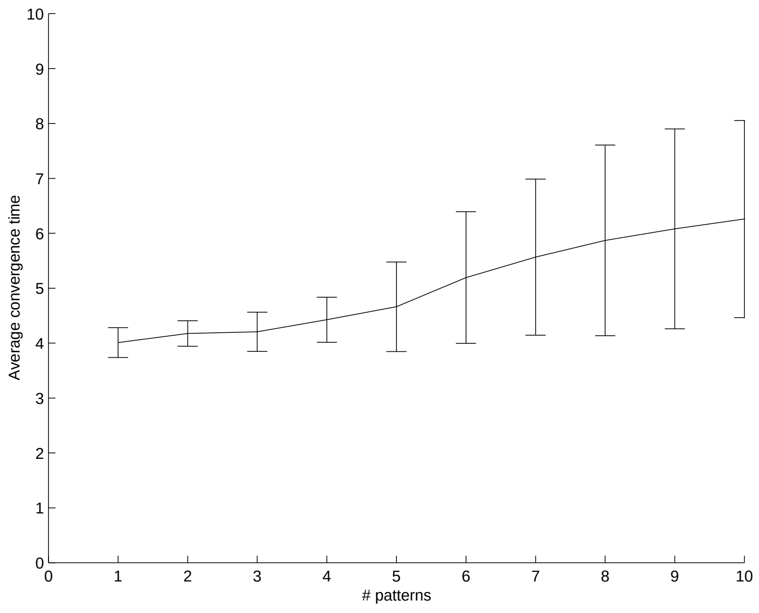


Figure 9. Convergence time for increasing memory load. The network is trained with 10 patterns, but has capacity for only around 7. Parameters as in figure 8.

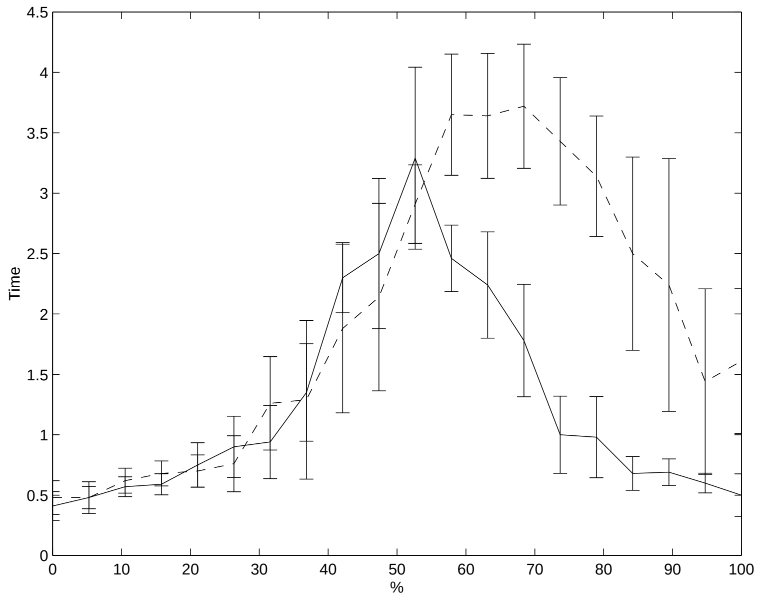


Figure 10. Convergence time for mixed starting activations plus noise. The input to the network consists of a linear interpolation between two trained patterns plus 16% noise. The solid curve represents interpolation between the latest and second latest pattern, the dashed curve interpolation between the latest and an old pattern (#6). Parameters as in figure 8.

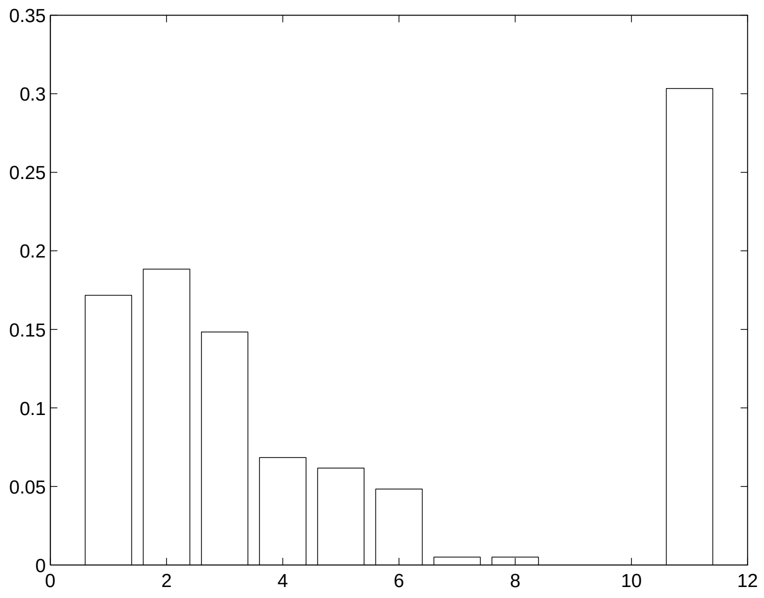


Figure 11. Probability of ending up in pattern 1–10 or the a priori attractor (11) when activating the network with a random pattern (10% activity, 600 trials, parameters as in figure 8).

supporting an already saturated synapse, while non-supporting information will affect it; throwing out old knowledge is favored regardless of how much evidence has accumulated for them. The learning rule proposed here does not suffer from such a drastic cut-off non-linearity.

Tanaka and Yamada [28] showed that in a Hopfield-type autoassociative network the convergence time is on the order of $\log(N)$ for successful recall and much longer for unsuccessful recall. The differences in convergence time between recent and remote memories seems to depend on the flattening of the basins of attraction for the memories as new information is overlaid old. This fits well with our observation of the emergence of a tail in the convergence time distribution as the oldest patterns are forgotten; the tail corresponds to slow convergence to the a priori attractor. It might be interesting to explore the effect of sparseness and network size on convergence time. Since the weights between units are independent of network size, the average magnitude of the support and hence the convergence speed will increase for larger networks, since each unit will be connected to more units. If the mean activity grows as slow as $\log(N)$ [18] the memory capacity might be optimal but the support will also only grow logarithmically.

In our simulations with a fast learning and forgetting "working memory" we have found that the average convergence time increases significantly with memory load. This relates to the classical finding by Sternberg of a reaction time dependence on list length [27] which has been used to support hypotheses of scanning processes underlying working memory [21]. Our results suggest that a similar effect can be achieved in an attractor network with fast learning dynamics due to attractors of different strength and a convergence time dependent on the attractor depth.

There remain several interesting further extensions to the real-time Bayesian learning rule proposed here. We are currently working on a soft simultaneity measure and machinery to cope with delayed reinforcement signals. If the inputs $o_j(t)$ to the learning rule are filtered using IIR filters $o'_i(t) = \gamma o_j(t) + (l - \gamma)o'_j(t - l)$ it becomes possible to correlate stimuli with slightly different timing and to do simple temporal association. Such filtering could be related to the calcium concentration dynamics inside neurons, which is increased e.g. by NMDA receptor activation and presumably involved in learning processes. By giving the pre- and postsynaptic inputs different time constants γ_j and γ_j the weight symmetry can be broken, which opens the possibility for sequence learning and the creation of more complex attractors than point attractors.

In addition, in a real environment changes are likely to occur at multiple time-scales. By selecting one single learning time constant, a scale of temporal detail is selected. The learning system will average out the events that occur on faster time-scales than the learning time constant and adapt to slower changes. It is interesting to consider the possibility of having a memory system comprised of multiple networks with different learning dynamics and degrees of plasticity. A quickly adapting network would learn and remember presented objects in working memory, while a more slowly forgetting network might learn from single presentations (as in episodic memory), and even slower learning and forgetting networks would average individual presentation events into a "prototypic" semantic memory (cf. [4] for one implementation). A similar kind of structure is thought to exist in human memory systems [26].

A further important aspect of memory is that of relevance information and print-now mechanisms, external signals regulating the strength of memory encoding in a behaviorally useful way. The learning rule proposed here will result in memory traces that are volatile in the absence of input since the weights are continuously changing to obey the current rate estimates. This leads to a gradual decay of memory over time even when little new information arrives. An alternative possibility is to control learning rate by some kind of relevance or "print-now" signal. In this case, only simultaneous pre- and postsynaptic activation is not enough to result in weight changes. It is only when plasticity is enabled by the print-now signal that changes occur, equally for imprinting and decaying. With regard to our learning rule, we have found that this can easily be implemented as a change in the learning time constant (α is increased with the relevance of the situation). This improves retention since the rate estimates do not decay in the absence of new relevant data. Such a mechanism relates closely to neuromodulation in the brain, e.g. the effect of dopamine [29] and acetylcholine in synaptic plasticity [31, 11].

Acknowledgments

Thanks to Mikael Djurfeldt for much helpful criticism. This work was supported by TFR grant No. 97274 and MFR grant No. 12716.

References

[1] Amit, D. J. *Modeling Brain Function: The World of Attractor Neural Networks.* Cambridge University Press, 1989.

[2] Bayes, T. An essay towards solving a problem in the doctrine of chances. *Biometrica (reprint of orig art in Philos. Trans. R. Soc. London 53, pp. 370–418, 1763) 45* (1958), 296–315.

[3] Bonnaz, D. Storage capacity of generalized palimpsests. *J. Phys. I France 7* (1997), 1709–1721. December.

[4] Brunel, N., Carusi, F., and Fusi, S. Slow stochastic Hebbian learning of classes of stimuli in a recurrent neural network. *Network 9* (1998), 123–152.

[5] Fransén, E., and Lansner, A. Low spiking rates in a population of mutually exciting pyramidal cells. *Network 6*, 2 (1995), 271–288.

[6] Fransén, E., and Lansner, A. A model of cortical associative memory based on a horizontal network of conneted columns. *Network 18* (1998), 115–124.

[7] Freeman, W. J. The physiology of perception. *Scientific American 264* (1991), 78–85.

[8] Geszti, T., and Pázmándi. Learning within bounds and dream sleep. *J Phys. A: Math Gen 20* (1987), L1299–L1303.

[9] Haberly, L. B., and Bower, J. M. Olfactory cortex: model circuit for study of associative memory? *Trends Neurosci 12*, 7 (July 1989), 258–64.

[10] Hasselmo., M. E., Anderson, B. P., and Bower, J. M. Cholinergic modulation of cortical associative memory function. *J. Neurophysiol. 67* (1992), 1230–1246.

[11] Hasselmo, M. E., Wyble, B. P., and Wallenstein, G. V. Encoding and retrieval of episodic memories: role of cholinergic and gabaergic modulation in the hippocampus. *Hippocampus 6*, 6 (1996), 693–708.

[12] Hebb, D. O. *The Organization of Behavior.* John Wiley Inc., New York, 1949.

[13] Hertz, J., Krogh, A., and Palmer, R. G. *Introduction to the Theory of Neural Computation*, vol. 1 of *Lecture Notes.* Addison-Wesley, Santa Fe Institute for studies in the sciences of complexity, 1991.

[14] Holst, A. *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, Sept. 1997. TRITA-NA-P9708.

[15] Holst, A., and Lansner, A. A Bayesian neural network with extensions. Tech. Rep. TRITA-NA-P9325, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, 1993.

[16] Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A 79*, 8 (Apr. 1982), 2554–8.

[17] Lansner, A., and Ekeberg, Ö. An associative network solving the "4-Bit ADDER problem". In *IEEE First International Conference on Neural Networks* (San Diego, CA, June 21–24 1987), M. Caudill and C. Butler, Eds., pp. II–549.

[18] Lansner, A., and Ekeberg, Ö. A one-layer feedback artificial neural network with a bayesian learning rule. *Int. J. Neural Systems 1*, 1 (1989), 77–87.

[19] Lansner, A., and Holst, A. A higher order Bayesian neural network with spiking units. *Int. J. Neural Systems 7*, 2 (May 1996), 115–128.

[20] Levy, W. B., and Desmond, N. L. *Synaptic Modification, Neuron Selectivity, and Nervous System Organization.* Lawrence Erlbaum Associates, Hillsdale, 1985, ch. The rules of elemental synaptic plasticity.

[21] Lisman, J. E., and Idiart, M. A. Storage of 7 $\pm$ 2 short-term memories in oscillatory subcycles. *Science 267*, 5203 (Mar 10 1995), 1512–5.

[22] Nadal, J., Toulouse, G., Changeux, J., and Dehaene, S. Networks of formal neurons and memory palimpsests. *Europhysics Letters 1*, 10 (1986), 535–542.

[23] Parisi, G. A memory which forgets. *J. Phys. A: Math. Gen 19* (1986), L617–620.

[24] Petersen, C. C., Malenka, R. C., Nicoll, R. A., and Hopfield, J. J. All-or-none potentiation at CA3-CA1 synapses. *Proc Natl Acad Sci U S A 95*, 8 (Apr 14 1998), 4732–7.

[25] Quinlan, P. *Connectionism and Psychology. A Psychological Perspective on Connectionist Research.* Harvester, Wheatsheaf, New York, 1991.

[26] Squire, L. R. Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review 99* (1992), 195–231.

[27] Sternberg, S. High-speed scanning in human memory. *Science 153*, 736 (Aug 5 1966), 652–4.

[28] Tanaka, T., and Yamada, M. The characteristics of the convergence time of associative neural networks. *Neural Computation 5*, 3 (1993), 463–472.

[29] Wickens, J., and Kötter, R. Cellular models of reinforcement. In *Models of Information Processing in the Basal Ganglia*, J. C. Houk, J. L. Davis, and D. G. Beiser, Eds. MIT Press, 1995, pp. 187–214.

[30] Willshaw, D., Buneman, O., and Longuet-Higgins, H. Non-holographic associative memory. *Nature 222* (1969), 960.

[31] Woolf, N. J. The critical role of cholinergic basal forebrain neurons in morphological change and memory encoding: a hypothesis. *Neurobiol Learn Mem 66*, 3 (Nov. 1996), 258–66.

Appendix A. Explicit derivation of $\Lambda(t)$

$\Lambda_i(t)$ and $\Lambda_{ij}(t)$ can be calculated explictly in the continous case given a stimulus sequence $\phi(t)$ and the time variation of the time constant $\alpha = \alpha(t)$. The derivation is the same for $\Lambda_i(t)$ and $\Lambda_{ij}(t)$, so the subscript will be dropped and α and $\phi(i)$ will denote the respective input.

Note that equation 12 may be reduced ti the form

$$\frac{d\Lambda}{dt} = \alpha\phi - \alpha\Lambda$$

$$\frac{d\Lambda}{dt} + \alpha\Lambda = \alpha\phi$$

Define $g(t)$:

$$g(t) = \int_0^t \alpha(s)ds \Rightarrow \frac{dg}{dt} = \alpha(t)$$

Introducing the intergrating factor e^g, we get

$$\frac{d}{dt}[e^g\Lambda] = e^g\frac{d\Lambda}{dt} + \Lambda e^g\frac{dg}{dt} = e^g[\frac{d\Lambda}{dt} + \alpha\Lambda] = \alpha\phi e^g$$

$$\Rightarrow e^g\Lambda = \int_0^t \alpha(s)\phi(s)e^{g(s)}ds + C$$

$$\Lambda(t) = Ce^{-g(t)} + e^{-g(t)}\int_0^t \alpha\phi e^g ds = \Lambda(0)e^{-g(t)} + \int_0^t \alpha(s)\phi(s)e^{-(g(t)-g(s))}ds \quad \text{(A.1)}$$

Define $\chi(t,s)$ as

$$\chi(t,s) = \begin{cases} 1 & 0 \le s \le t \\ 0 & s > t \end{cases}$$

Then equation A.1 becomes

$$\Lambda(t) = \Lambda(0)e^{-g(t)} + \int_0^\infty \overbrace{\chi(t,s)\alpha(s)e^{-(g(t)-g(s))}}^{\psi(s,t)}\phi(s)ds$$

$$= \Lambda(0)e^{-g(t)} + \int_0^\infty \psi(t,s)\phi(s)ds$$

From this an explicit formula for the weights and biases can be calculated.

For example, let

$$\phi(t) = \begin{cases} 1 & t_n \le t \le t_n + \delta_n \\ 0 & \text{otherwise} \end{cases}$$

where t_n and $\delta_n \in \mathbb{R}$. Let $\alpha(t) = \alpha$. For periods when $\phi(t) = 0$ we get

$$\Lambda(t) = \Lambda(0)e^{-\alpha t} + \int_0^t \alpha e^{-\alpha(t-s)}\phi(s)ds = \Lambda(0)e^{-\alpha t} + \alpha \sum_{t_n+\delta_n<t} \int_{t_n}^{t_n+\delta_n} e^{-\alpha(t-s)}ds$$

$$= \Lambda(0)e^{-\alpha t} + \sum_{t_n+\delta_n<t} e^{-\alpha(t-t_n-\delta_n)} - e^{-\alpha(t-t_n)} = \Lambda(0)e^{-\alpha t} + \sum_{t_n+\delta_n<t} e^{-\alpha(t-t_n)}(e^{\alpha\delta_n} - 1)$$

and if $\phi(t) = 1$ $(t_{n+1} < t < t_{n+1} + \delta_{n+1})$

$$\Lambda(t) = \Lambda(0)e^{-\alpha t} + \sum_{t_n+\delta_n<t} e^{-\alpha(t-t_n)}(e^{\alpha\delta_n} - 1) + 1 - e^{-\alpha(t-t_{n+1})}$$