Abstract

In order to avoid catastrophic forgetting memory systems with limited capacity need to forget old or irrelevant information. 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 neural network with hypercolumns that has palimpsest properties when employed in an attractor neural network. The network does not exhibit catastrophic forgetting, has an effective capacity controlled by on the learning time constant and exhibits decreasing convergence speed for older patterns.

1 Background

- Learning systems with finite capacity have a problem when dealing with the real world: the world refuses to give a limited number of training examples, and their capacity will be overloaded. If they are to act in a complex environment that can change, they also have to continue to learn indefinitely.

- The way out of the dilemma is to have some way of forgetting information to make place for new.

- A popular class of learning systems is autoassociative neural networks, which often have been used as models for associative memory. However, the standard correlation-based learning rules used suffer from catastrophic forgetting: all memories are lost as the system gets overloaded. This runs contrary to what is believed to happen in working and intermediate memory such as the hippocampus, where new information needs to be continuously learned, stored and retrieved but not necessarily remembered beyond a certain time [1]

- One solution is palimpsest memories [2]: new patterns are stored on top of older ones, which are gradually overwritten and become inaccessible. Various palimpsest modifications of the Hopfield learning rule exist, such as learning within bounds [3, 4, 5]

- One form of autoassociative ANN is the Bayesian Confidence Propagation Neural Network (BCPNN) [6, 7, 8, 9]. It is based on a probabilistic view of of learning and retrieval, with input and output unit activities representing confidence of feature detection and posterior probabilities of outcomes, respectively. The learning rule is a Hebbian rule based on Bayes' rule, setting weights depending on the probability of units being co-active and biases depending on their a priori likelihood.

- The traditional BCPNN can as we will show be modified to perform incremental learning, enabling it to act as a palimpsest memory and hence avoid catastrophic forgetting.

2 Bayesian Confidence Propagation Neural Networks (BCPNN)

- The BCPNN (Bayesian Confidence Propagation Neural Network) is based on a Hebbian learning rule derived from Bayes rule [6, 7].

- Unit activity represents the estimated probability of an event or feature.

- If $P(i)$ is the probability that unit i is activated by one of the training patterns and $P(i,j)$ the probability that both units are co-activated we get (where A denotes the set of units that have been activated)

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

  If we take the logarithm of this formula we get:

$$\log(P(j|A)) = \log(P(j)) + \sum_i \log\left(\frac{P(i,j)}{P(i)P(j)}\right) o_i$$

  where o_i is 1 if $i \in A$. This can be turned into the activation of a neural network by identifying $o_i = P(i|A)$ as unit outputs, $\log(P(j))$ as biases, $\log(P(i,j)/P(i)P(j))$ as weights and a clipped exponential activation function.

- The network is trained by giving a series of example patterns, from which estimates of probabilities are derived; these are used to set the weights. When in retrieval mode, units corresponding to known features are activated and other units change their state to a posteriori estimates.

- The BCPNN can both be used as feedforward network, calculating posterior probabilities of events, and as an autoassociative network. In the latter case the unit outputs will not be strict probability estimates, but converge towards a consistent state representing a decision made on the input.

3 Incremental Probability Estimates

- To derive the connection weights estimates of the probabilities $P(i)$, $P(j)$, $P(i,j)$ have to be made.
- The traditional BCPNN learning rule does this by counting the number of occurrences of events i, j and ij in the training data, dividing this with the total number of training examples.
- Here we replace the fixed probability estimates with time varying estimates of rates of events that are updated as new information arrives. We want an estimate that
  - Will converge towards $P_i(t)$ and $P_{ij}(t)$ in a stationary environment
  - Gives more weight to recent than remote information
  - Has a time scale that leads to smoothing and adaptation to trends in a non-stationary environment.

  The simplest such estimate is exponential smoothing of unit activities:

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

  where τ is a suitable time constant. The rate Λ_{ij} of coincident firing can be similarly estimated

$$\frac{d\Lambda_{ij}(t)}{dt} = \frac{o_i(t)o_j(t) - \Lambda_{ij}(t)}{\tau}$$

- In order to prevent underflow in the probability estimates a low background or noise activity λ_0 is assumed.
- While τ has a straightforward interpretation as the learning time constant, the inverse $\alpha = 1/\tau$ is more practical in the following formulas.

  The inverse time constant α determines the baseline speed of how quickly the network will adapt to new information. Low α correspond to slow learning and forgetting, high α fast learning and forgetting. In other words, α defines the relevant time-scale for the learning dynamics.

4 Hypercolumns

- The BCPNN derivation assumes independence and conditional independence of events, which is a strong assumption. This can be avoided if units corresponding to correlated events are gathered into mutually independent groups, and the update rule modified slightly. This approach also allows the network to handle input data that can take on a number of values (instead of being a binary signal) [8, 9].

- Such a group of units will be called a hypercolumn, in analogy with the cortical orientation hypercolumns assumed to exist in primary visual cortex [10].

- Instead of having one unit i for every attribute we use a hypercolumn of events ii' $(1 \leq i' \leq M_i)$ representing the probability of observing value i' of attribute i. The total probability within a hypercolumn is normalized to 1.

- The update rule becomes

$$
\begin{aligned}
\eta \frac{dh_{ii'}(t)}{dt} &= \beta_{ii'}(t) + \sum_{j}^{N} \log\left(\sum_{j'}^{M_i} w_{ii'jj'}(t) o_{jj'}(t)\right) - h_{ii'}(t) \\
o_{ii'}(t) &= \frac{e^{h_{ii'}}}{\sum_j e^{h_{ij}}} \\
\beta_{ii'} &= \log(\Lambda_{ii'}) \\
w_{ii'jj'}(t) &= \frac{\Lambda_{ii'jj'}(t)}{\Lambda_{ii'}(t)\Lambda_{jj'}(t)}
\end{aligned}
$$

- Experiments with BCPNN using hypercolumns and incremental learning has shown qualitatively the same behavior as the non-hypercolumn case, but due to the inherent error-correction and management of correlations in the input data performance is significantly increased.

5 Network Equations

Using the previous equations, the network learning rule and update equations become:

$$
\begin{aligned}
\frac{dh_{ii'}(t)}{dt} &= \beta_{ii'}(t) + \sum_{j}^{N} \log\left(\sum_{j'}^{M_i} w_{ii'jj'}(t) o_{jj'}(t)\right) - h_{ii'}(t) \\
o_{ii'}(t) &= \frac{e^{h_{ii'}}}{\sum_j e^{h_{ij}}} \\
\frac{d\Lambda_{ii'}(t)}{dt} &= \alpha([(1-\lambda_0)o_{ii'}(t) + \lambda_0] - \Lambda_{ii'}(t)) \\
\frac{d\Lambda_{ii'jj'}(t)}{dt} &= \alpha([(1-\lambda_0^2)o_{ii'}(t)o_{jj'}(t) + \lambda_0^2] - \Lambda_{ii'jj'}(t)) \\
\beta_{ii'}(t) &= \log(\Lambda_{ii'}(t)) \\
w_{ii'jj'}(t) &= \frac{\Lambda_{ii'jj'}(t)}{\Lambda_{ii'}(t)\Lambda_{jj'}(t)}
\end{aligned}
$$

$h_{ii'}$ the potential of the i'th unit in hypercolumn i

N number of hypercolumns

M_i number of neurons in hypercolumn i

$o_{ii'}$ the output of unit ii'

$\beta_{ii'}$ the bias of unit ii'

$w_{ii'jj'}$ the weight between unit ii' and unit jj'

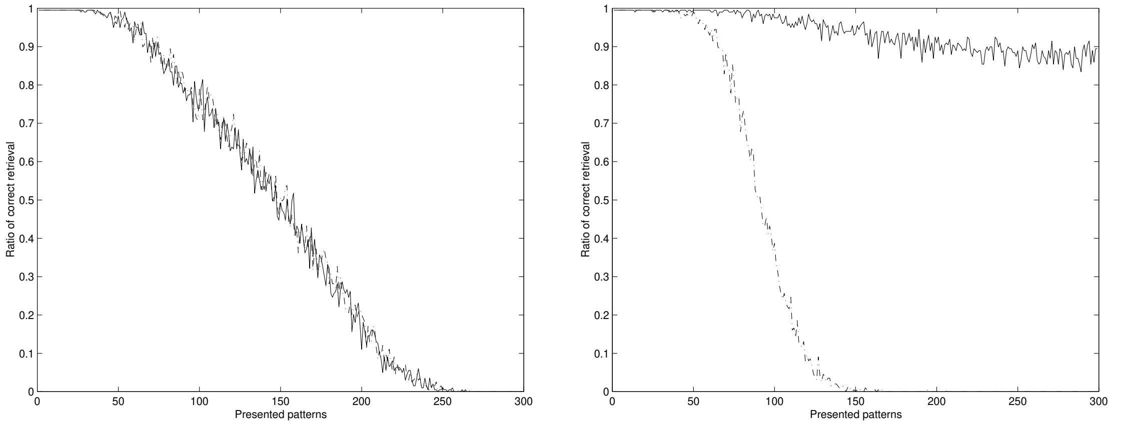
$\Lambda_{ii'}$, $\Lambda_{ij'jj'}$ rate estimates

λ_0 a low background activity (set to 10^{-4} throughout this paper)

α inverse of the learning time constant.

6 Forgetting

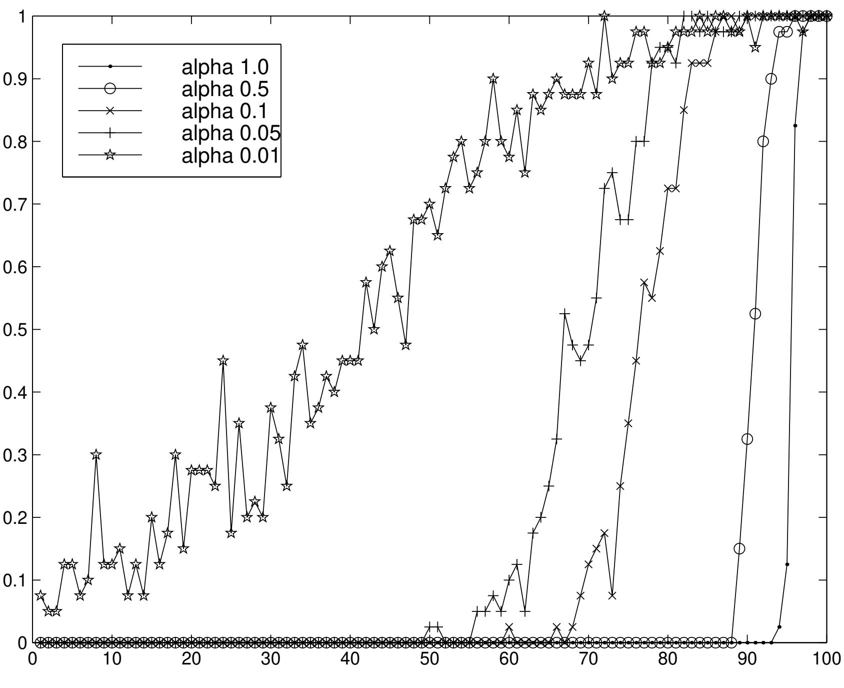
Summing vs. Incremental Learning



(a) Summing (b) Incremental

Comparison of the summing BCPNN learning rule and the incremental learning rule for sparse random patterns. Solid line: latest learned pattern, dash-dotted line: first learned pattern.

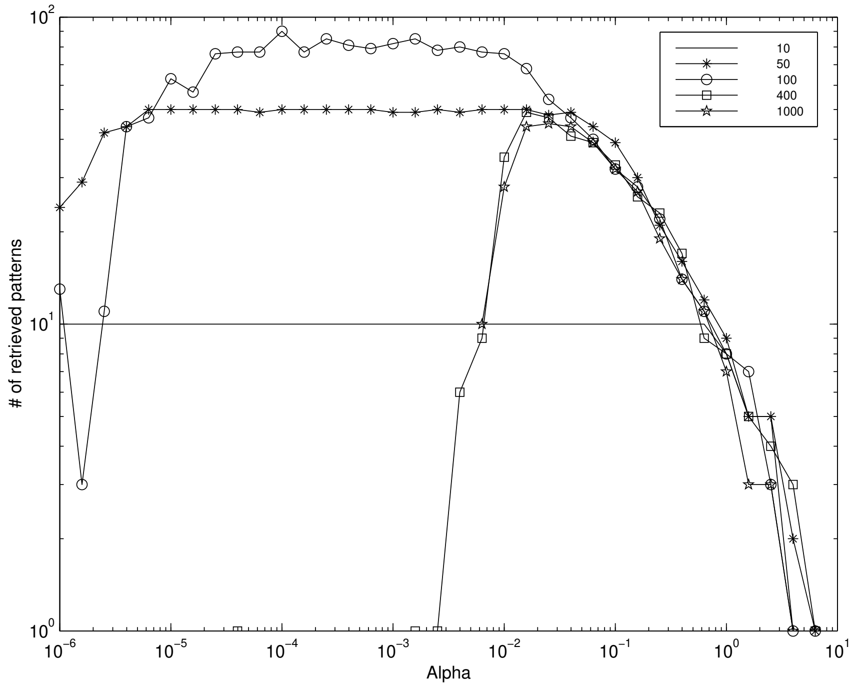
Forgetting curves



Forgetting curve during continuous learning. Pattern recall as a function of list position for different values of α (see legend).

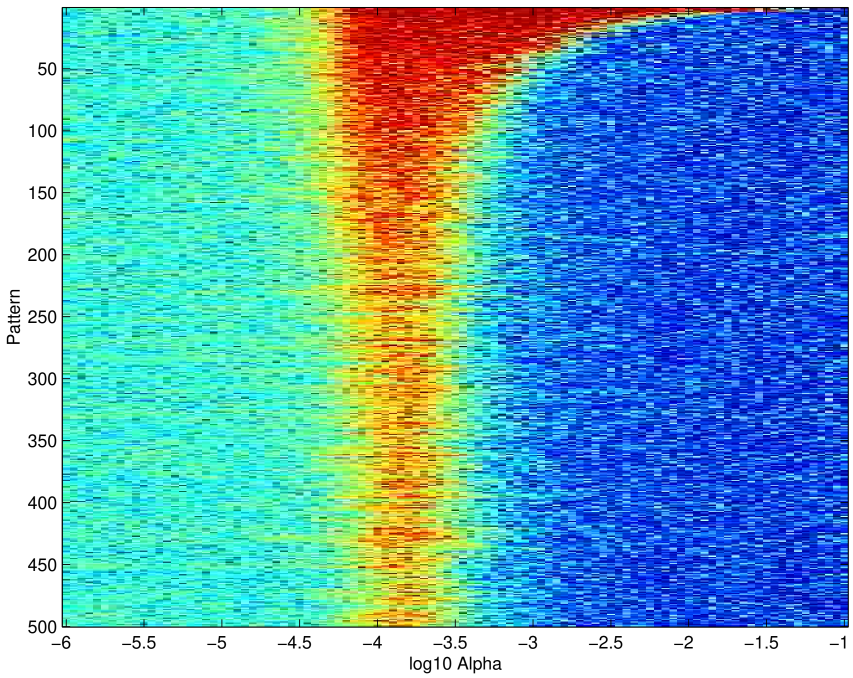
7 Capacity

Memory Capacity



Number of correctly retrieved patterns as a function of α for different sizes of the training set.

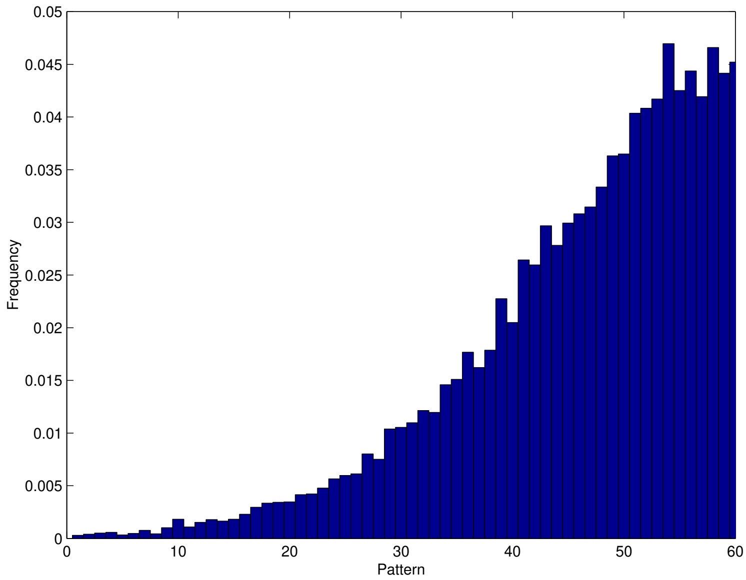
Retrieval



Retrieval of 500 training patterns for different values of α. Red = high overlap between target and result, blue = little overlap.

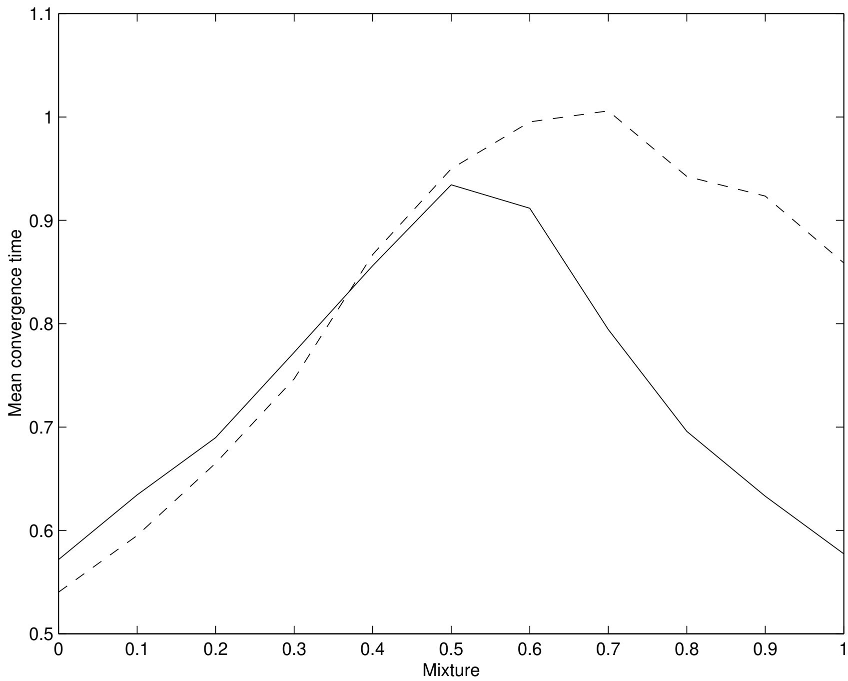
8

Random convergence



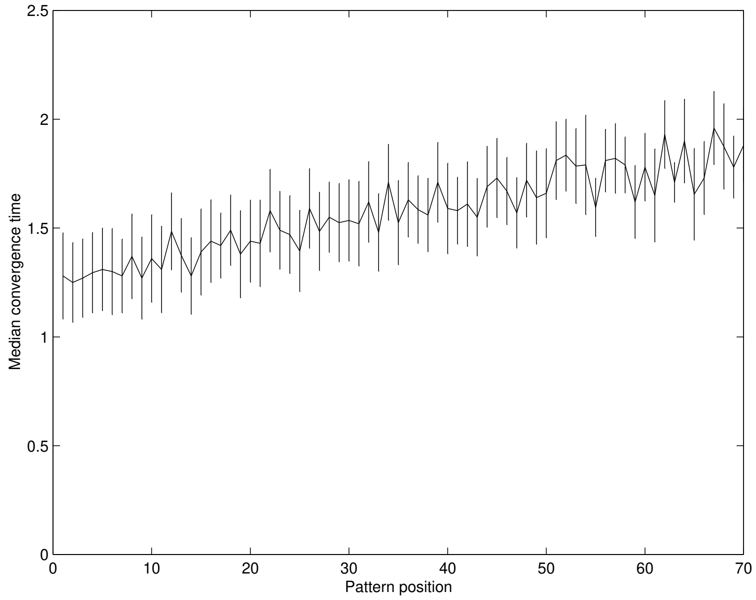
Frequency of ending up in the different patterns when activating the network with a random pattern (one randomly selected unit active in each hypercolumn) and allowing it to converge. Each bar corresponds to one pattern. The closest pattern to the resulting end state was used for the diagram.

Decision making

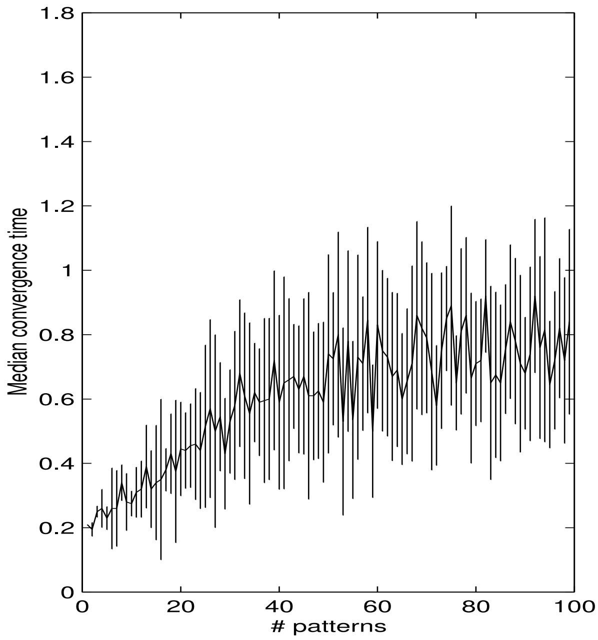


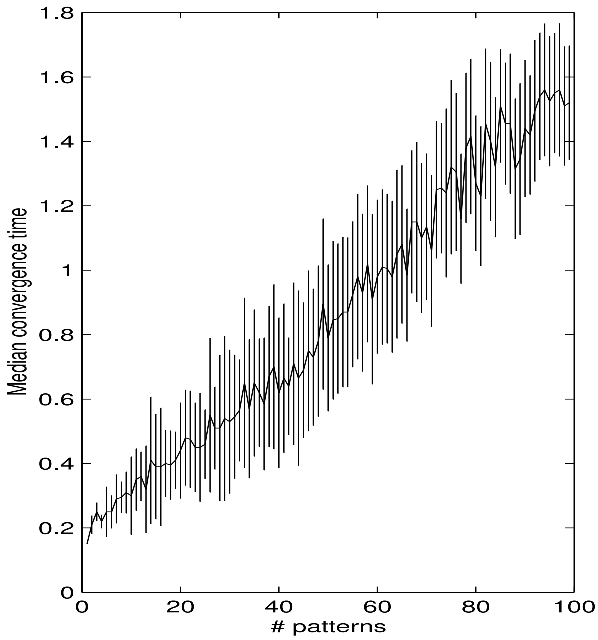
Convergence time for mixed input patterns. The input to the network consists of 0–10 hypercolumns of one pattern and the rest from the other pattern. The solid curve represents interpolation between the latest and second latest pattern, the dashed curve interpolation between the latest and an old pattern (#10).

9 Convergence Speed



Median convergence time and standard deviation for retrieval of stored patterns from a disturbed input, as a function of position in the training set. Pattern 1 is the most recently learned pattern. Only retrieval of correct patterns was counted, and since this was sparser beyond pattern 70 only the first 70 patterns are shown. $\alpha = 0.01$, the network was trained with 100 patterns.

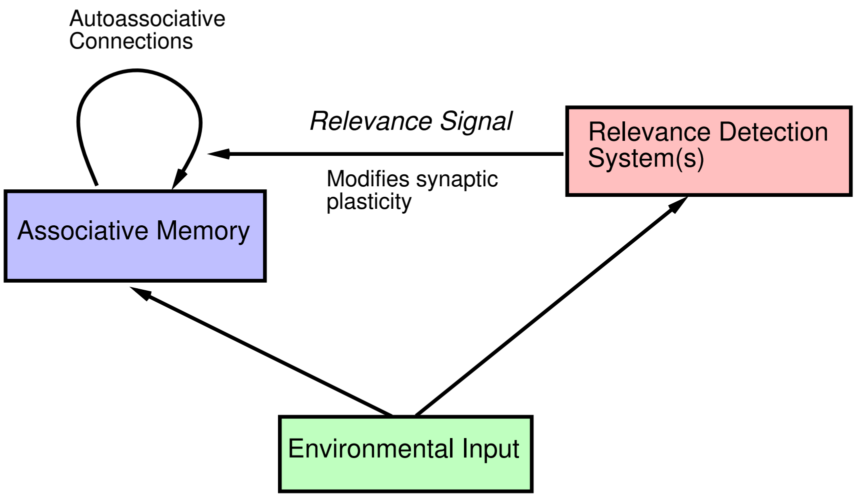




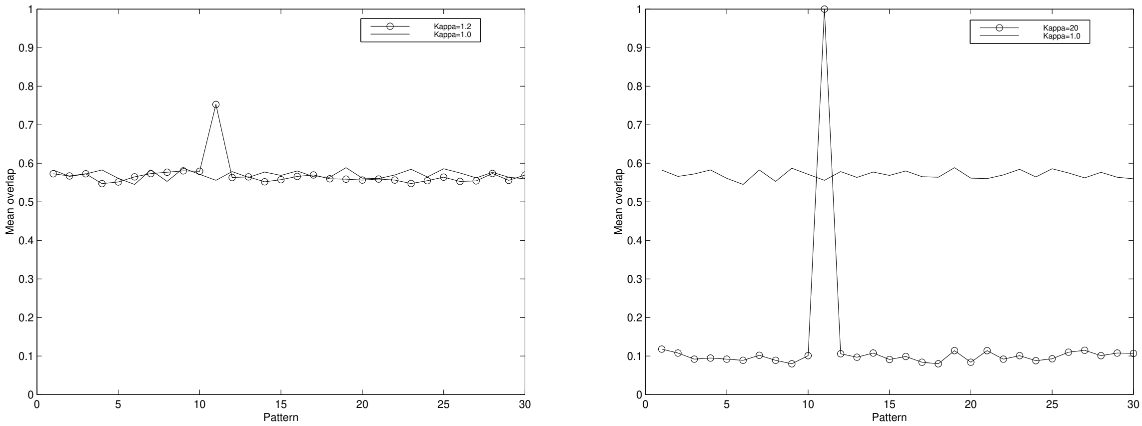
Median convergence time and standard deviation for increasing memory load for $\alpha = 0.1$ (left) and $\alpha = 0.01$ (right). The network is trained with up to 100 patterns, but for $\alpha = 0.1$ has capacity for only around 20–30.

10 Memory Modulation and the von Restorff Effect

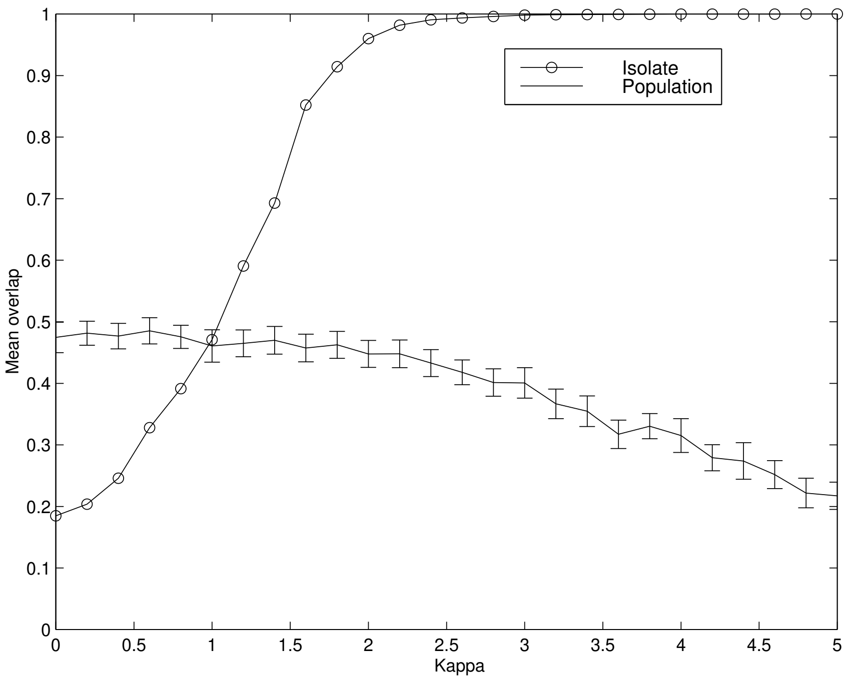
- The strength of the memory trace in biological memory systems appears to depend on among other things meaning, context, relational processing as well as factors like emotional significance and attentional allocation.
- Endogenous processes activated by experience can modulate memory strength in terms of recall probability. For example, emotionally arousing or humorous experiences are generally better remembered than less affective experiences. Hormones, neuromodulators and substances affecting them can affect how strongly experiences are encoded, i.e. the information is better retained.
- The novelty of a stimulus can modulate memory. The *von Restorff effect* consists of improved recall or recognition of a distinct or different item (the *isolate*) in a collection, while the other items are less well recalled (retroactive and proactive inhibition).
- Some of these factors can be interpreted in the framework of memory consolidation as a relevance modulation of the "print-now" signal by regulating memory encoding and synaptic plasticity. During ordinary events the plasticity is lower than when salient, arousing or motivational stimuli increase the learning rate, for example through a modulating mechanism. Such a mechanism may be related or serve as a model for neuromodulatory mechanisms in the brain, e.g. the effect of dopamine and acetylcholine on synaptic plasticity.
- This can be modeled in the incremental BCPNN by multiplying α by a stimulus-dependent number κ_i.



11 Recall quality depending on strength of the relevance signal



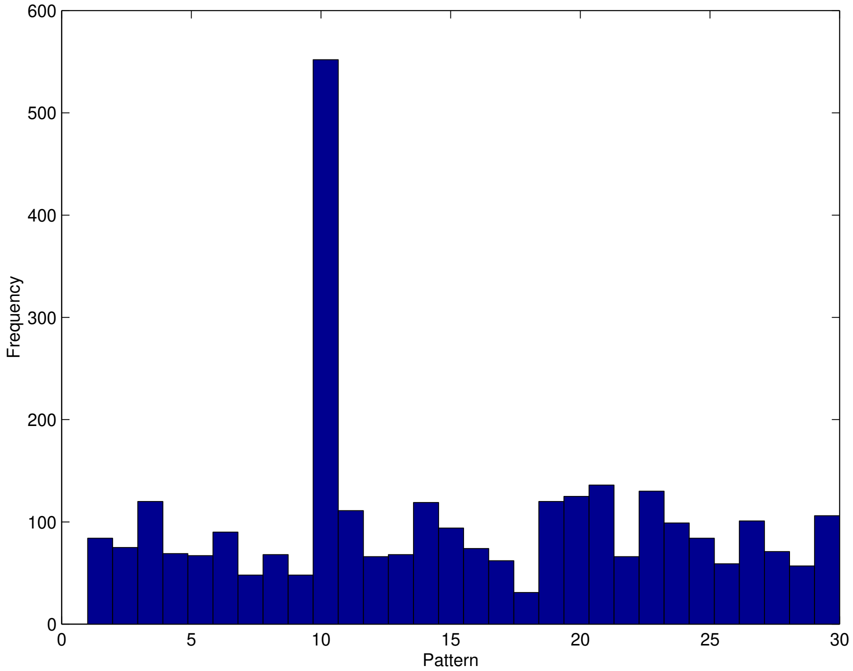
Mean overlap with target of the different patterns after convergence from noisy cue. Solid line: no isolate pattern, marked line: one isolate pattern. In the dotted run, κ_i was set to 1.2 (left) or 20 (right) for pattern 11. $\alpha = 10^{-8}$ in both plots.



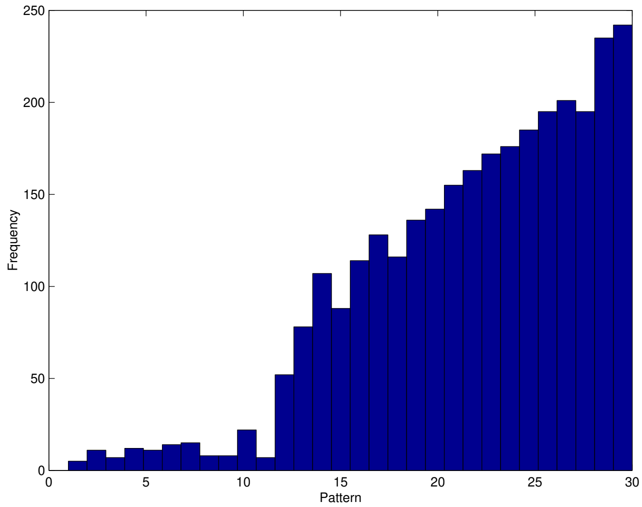
Mean overlap with target of the isolate pattern and the other patterns as a function of the increase in plasticity κ_i. For high values of κ_i other patterns are inhibited, but for $1 < \kappa_i < 2$ retrieval of the isolate is enhanced without significant inhibition of other learned patterns.

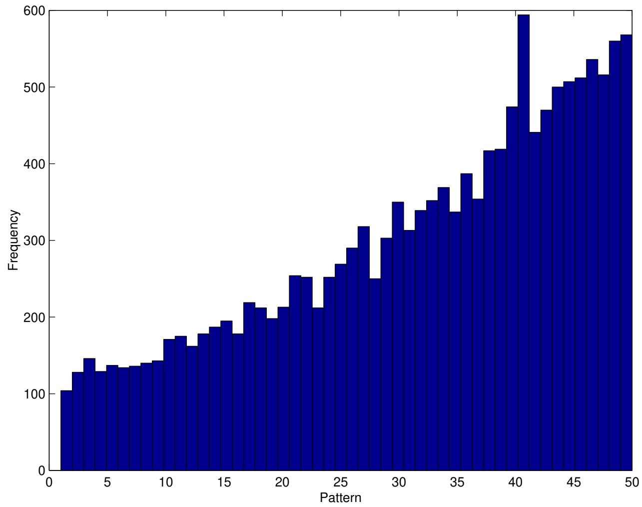
12 Free Recall

Frequency of ending up in the attractors corresponding to the learned patterns when activating the network with a random pattern (one randomly activated unit per hypercolumn) for low and high levels of plasticity and modulation.



Enhanced probability of recall of isolate compared to other patterns. $\kappa_i = 1.5$, $\alpha = 10^{-8}$, pattern 10 isolate.





Interaction with forgetting curve. To the left retrograde inhibition due to an isolate ($\kappa_i = 20$, $\alpha = 10^{-1.5}$). Right: enhancement without inhibition ($\kappa_i = 2$, $\alpha = 10^{-2}$. Pattern 41 is the isolate).

13 Summary

The incremental BCPNN with hypercolumns has many attractive features:

- Palimpsest memory properties: old memories are forgotten to give room for new, preventing catastrophic forgetting.
- A capacity that can be regulated by the network size and α. The maximal capacity equals the capacity of the non-incremental network and is higher than other palimpsest memories.
- Convergence time depends both on the age and number of stored patterns, showing an increase near decision boundaries. This can be used to demonstrate how a non-serial system can produce the linear reaction time increase in the Sternberg Effect [11].
- By modulating the plasticity level the network can be used to model the von Restorff effect [12].
- While the learning and update rules have been derived from abstract statistical considerations, it has many properties similar to biologically realistic neural networks. The non-incremental form of the BCPNN learning rule has been used for modeling cortical cell assemblies [13]. A modified form of the learning rule has been used as a quantitative fit to biological LTD/LTP-data [14].

Contact Information

Anders Sandberg	asa@nada.kth.se
Anders Lansner	ala@nada.kth.se

SANS, Department of Numerical Analysis and Computing Science, Royal Institute of Technology S-100 44 Stockholm, Sweden. Tel: +46 (8) 790 77 84 Fax: +46 (8) 790 09 30

Karl Magnus Petersson	karlmp@neuro.ks.se

PET/Cognitive Neurophysiology Research Group, Karolinska Hospital, Stockholm, Sweden

14 References

[1] J.L. McClelland and N.H. Goddard. Considerations arising from a complementary learning systems perspective on hippocampus and neocortex. *Hippocampus*, 6:654–665, 1997.

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

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

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

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

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

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

[9] A. 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.

[10] D.H. Hubel and T.N. Wiesel. Uniformity of monkey striate cortex: A parallel relationship between field size, scatter and magnification factor. *J. Comp. Neurol.*, 158:295–306, 1974.

[11] A. Sandberg, A. Lansner, and K.M. Petersson. A bayesian connectionist model for memory scanning. Poster, 2000. Presented at XXVII International Congress of Psychology, Stockholm, poster 11609.06.

[12] A. Sandberg, A. Lansner, and K.M. Petersson. Selective enhancement of recall through plasticity modulation in an autoassociative memory. *Neurocomputing*, 2001. Accepted for publication.

[13] E. Fransén and A. Lansner. A model of cortical associative memory based on a horizontal network of conneted columns. *Network*, 9:235–264, 1998.

[14] N. Wahlgren and A. Lansner. Biological evaluation of a hebbian-bayesian learning rule. *Neurocomputing*, 2001. Accepted for publication.