GWLA
FREE

ILL Request:36453981

Borrower:RCE

SCIENCE
SHELVED BY TITLE V.1-2 1989-91 (LIMITED LOAN)
AVAILABLE

Ariel Address: 168.7.30.4

SCIENCE
SHELVED BY TITLE V.3 1992 (LIMITED LOAN)

CAN YOU SUPPLY? YES NO CONDITIONAL FUTURE DATE

Affiliation: TexShare, TEXPRESS (140/HOU), HARLiC, ARL, CARLA, BIG 12+

Status: PENDING 20071025 Request Date: 20071025 Need Before: 20071124

OCLC: 21018895 Source: ILLiad Due Date:

Lender: *ORU, COD, AZU, UBY, NTERequest

CALLNO: SCIENCE Format:unspecified

Title: International journal of neural systems.
Article: A Lansner and O Ekeberg: A one-layer feedback artificial neural network with a bayesian learning rule
Volume: 1
Date: 1989
Pages: 77-87
Imprint: Singapore ; Teaneck, N.J. : World Scientific, c1989-
ISSN: 0129-0657

Verified: <TN:237914><ODYSSEY:rice.illiad.oclc.org/ILL> OCLCMy Library's Holdings InformationLHR Summary:v.1-v.10(1989-2000)Lending Policies:Unknown / UnknownLocation:ORVN
Type: Copy
Patron: Cox, Steve
Bill To: same FEIN1-74-1109620-3; TEXSHARE;
Ship Via: TEXPRESS, ARIEL, FAX, UPS, FedEX, 1st class mailElectronic Delivery:Odyssey - rice.illiad.oclc.org/ILL
Maximum Cost: IFM - $50.00
Copyright Compliance: CCL
Fax: 713-348-4117;TEL#:713-348-2284
Email: ill@rice.edu ARIEL: 168.7.30.4
Borrowing Notes:
Lending Charges:
Shipped:
Lending Notes:
Lending Restrictions:
Return To:
Return Via:

140-HOU VIA TEXPRESS
ILL
Rice Univ. Library, MS-240
6100 Main St
Houston,TX 77005

International Journal of Neural Systems, Vol. 1, No. 1 (1989) 77–87
© World Scientific Publishing Company

A ONE-LAYER FEEDBACK ARTIFICIAL NEURAL NETWORK WITH A BAYESIAN LEARNING RULE

Anders Lansner and Örjan Ekeberg
Dept. of Numerical Analysis and Computing Science
Royal Institute of Technology, Stockholm, Sweden

Received 27 February 1989

A probabilistic artificial neural network is presented. It is of a one-layer, feedback-coupled type with graded units. The learning rule is derived from Bayes's rule. Learning is regarded as collecting statistics and recall as a statistical inference process. Units correspond to events and connections come out as compatibility coefficients in a logarithmic combination rule. The input to a unit via connections from other active units affects the *a posteriori* belief in the event in question.

The new model is compared to an earlier binary model with respect to storage capacity, noise tolerance, etc. in a content addressable memory (CAM) task. The new model is a real time network and some results on the reaction time for associative recall are given. The scaling of learning and relaxation operations is considered together with issues related to representation of information in one-layer artificial neural networks. An extension with complex units is discussed.

Introduction

During the last few years, research in the area of neural computation has brought forward a number of different models. Although based on the same general scheme, they differ considerably with respect to both performance, e.g. learning capability, computational demand, robustness, etc. and biological plausibility. It is common to distinguish between two classes of models, those of the one-layer, feedback-coupled (recurrent) type and the multi-layer feedforward ones. The former are essentially mathematical realizations of Hebb's theory of cell assemblies.[1] Examples are the linear[2,3,4,5] and binary[6,7] matrix models and the Hopfield model.[8] Models of this class have been used as auto- or heteroassociative CAMs, for solving constraint satisfaction problems, etc. Computer simulations primarily aimed at investigating the biological plausibility of Hebb's theory have also been undertaken.[9,10,11]

The second class of neural network models are the feedforward networks, e.g. multi-layer feedforward perceptrons. A well-known and commonly used learning rule for these systems is based on back-propagation of errors.[12] The multi-layer feedforward architectures are primarily used for hetero-association, i.e. for learning stimulus-response (S-R) mappings.

The binary, one-layer model

In an earlier investigation,[13] we studied the capabilities of a binary one-layer matrix model operating as a content addressable memory (CAM). A deterministic, asynchronous relaxation scheme was developed and we investigated the storage capacity and noise tolerance by means of computer simulation. Earlier theoretical results on the storage capacity[7] were verified. In a network with N units the number of random patterns possible to store increases proportional to $(N/\log N)^2$, provided that the number of active units in each pattern increases as $\log N$. As an example, a network with 3000 units could store close to 18 000 random patterns, each comprised of 16 active units. For each of these, when activating half of its units, the other half could be retrieved with 95% reliability. This is an efficient memory in the sense that the information that can be retrieved is proportional to the size of the physical memory (the connection matrix). The scaling properties and potential speed-up given a massively parallel machine were also considered.

A network with binary connections and binary output units is clearly a dramatic simplification with respect to biology and, furthermore, its functionality is limited. The aim of the work presented here was to modify the binary model to incorporate graded connections and continuous units. Graded connections would allow units to become more or less specific depending on their frequency of occurrence in the training set. The model also treats graded stimulation more properly since units can produce graded outputs in response to graded stimulation. Having units with continuous output makes a truly parallel, synchronous updating scheme feasible. Convergence to a stable state can be guaranteed provided the connection matrix is symmetric and the input-output relation of the units is monotonically increasing.[14] When searching for a suitable learning rule, unit input-output relation, and relaxation procedure for the new model, focus was on maintaining the "efficient storage" properties and noise tolerance in the CAM task related above.

A Bayesian Learning Rule

A model with graded connections but binary output units was recently presented.[15] It employs a Bayesian learning rule which gives it a slightly higher storage capacity and the same scaling behavior as the earlier binary model. In the following we will describe this learning rule in somewhat more detail and also give a formulation that allows for graded unit output. Thereafter, a couple of implementation issues will be discussed and some results from computer simulations of the improved model will be presented.

Derivation of the learning rule

A *probabilistic* perspective on learning and recall in neural networks often seems natural. Learning is then regarded as collecting statistics and recall as a statistical inference process based on the world model thus acquired, i.e. through experience. In the following, the training set consists of a number of activity patterns on a network with N units. Each unit may be seen as representing a feature or event in the observable world.

Assume that we have observed some events and want to determine the probability of some other events. Provided that the observed events are statistically independent, Bayes's rule can be used to calculate the *posterior* probability of the latter event in the following way.

According to Bayes's rule, if we know that event i has occurred and we want to know the *posterior* probability of event q, then

$$p(q \mid i) = p(q)\frac{p(i \mid q)}{p(i)} .$$

If we instead consider q conditioned by the composite event i & j,

$$p(q \mid i \,\&\, j) = p(q)\frac{p(i \,\&\, j \mid q)}{p(i \,\&\, j)} . \tag{1}$$

Assuming i and j are independent, both with and without q given, we have

$$p(i \,\&\, j) = p(i)p(j) \tag{2}$$

$$p(i \,\&\, j \mid q) = p(i \mid q)p(j \mid q) . \tag{3}$$

Inserting (2) and (3) into (1) gives:

$$p(q \mid i \,\&\, j) = p(q)\frac{p(i \mid q)}{p(i)}\frac{p(j \mid q)}{p(j)} . \tag{4}$$

The same line of reasoning holds even when there are several conditioning events i, j, k, etc.:

$$\begin{aligned} p(q \mid i \,\&\, j \,\&\, k \,\&\, \ldots) &= p(q)\frac{p(i \mid q)}{p(i)}\frac{p(j \mid q)}{p(j)}\frac{p(k \mid q)}{p(k)} \\ &= p(q)\prod_{h \in A}\frac{p(h \mid q)}{p(h)} . \end{aligned} \tag{5}$$

Here, A stands for the set of observed events. In the following, a one-to-one correspondence is assumed between events and units in a model neural network. An observed event is represented by activity in the corresponding unit. Inactive units represent a lack of knowledge rather than knowledge that the events in question have not occurred. Units, i.e. model neurons, are assumed to sum their inputs according to

$$s_q = \beta_q + \sum_h w_{hq}\pi_h . \tag{6}$$

Here s_q is the support of the receiving unit q, β_q is the bias of q, π_h is the output of the sending unit h and w_{hq} is the weight of the connection from h to q. If binary output units are assumed and A is the set of active units we have

$$s_q = \beta_q + \sum_{h \in A} w_{hq} . \tag{7}$$

Taking the logarithm of (5) we get

$$\log p(q \mid A) = \log p(q) + \sum_{h \in A}\log\frac{p(h \mid q)}{p(h)} . \tag{8}$$

By identifying terms in (7) and (8) we arrive at formulas for weights and bias values:

$$w_{iq} = \log \frac{p(i \mid q)}{p(i)} = \log \frac{p(q \mid i)}{p(q)}$$

$$\beta_i = \log p(i) \,. \tag{9}$$

Characteristics of the learning rule

Our learning rule is yet another instance of a Hebbian learning rule in the sense that persistent co-activation of two units leads to the strengthening of the positive (excitatory) connection between them. Moreover, it also provides lateral inhibition, since anti-correlated activation of two nodes brings about a negative weight between them. Statistically independent units will get zero connections between them. From (9) it also follows that the connection matrix **W** is always symmetric which, as already mentioned, is critical for having convergence to stable states.[14]

Implementation

Here, the purpose of learning is to collect statistics of unit activity and co-activity of pairs of units to be able to estimate the probabilities and conditional probabilities used to calculate **W** and β_q values. An input pattern consists of a stimulus strength in the range [0, 1] for each unit in the network. The logarithm of the stimulation replaces the bias value in (7). In our model, the network is entirely "stimulus driven" during learning. Otherwise the network would first interpret the input and thereafter learn its own interpretation which is not desirable. This also has favorable consequences for computing time during learning.

We have been working with two different implementations of the learning algorithm, one aimed at a stationary and the other at a non-stationary world. In the first case, which will be described here, we developed a model based simply on counting occurrences and co-occurrences. More recently we have developed an alternative incremental learning rule based on calculating running averages. Here weight decay ("forgetting") is also incorporated. By adjusting a time constant for connection modification the plasticity of the network can be changed. This implementation and its consequences will be described in a forthcoming paper.[16]

In the counter model, a matrix **C** is used to accumulate statistics. When values for weights are required these are calculated from **C**. For each unit i we want to determine its bias value and for each pair of units $\langle i, j\rangle$ the connection between them according to:

$$\beta_i = \log p(i) \qquad w_{ij} = \log \frac{p(j \mid i)}{p(j)} = \log \frac{p(i \,\&\, j)}{p(i)p(j)} \,.$$

Since our estimate of $p = c/Z$ we have

$$\beta_i = \log \frac{c_i}{Z} \qquad w_{ij} = \log \frac{c_{ij} Z}{c_i c_j}$$

where

$$Z = \sum_{\alpha} \kappa^{(\alpha)} \quad c_i = \sum_{\alpha} \kappa^{(\alpha)} \pi_i \quad c_{ij} = \sum_{\alpha} \kappa^{(\alpha)} \pi_i \pi_j \,. \tag{10}$$

Here, π_i is the output of unit i, α is an index over the patterns in the training set, and κ is the significance attributed to a certain learning event. The parameter κ actually serves an important functional role as a global, graded "print now" control providing a mechanism for over-representing subjectively important learning examples and ignoring unimportant ones. Some special care has to be taken when counters come out as zero. In case c_i or c_j is zero, w_{ij} is also set to zero. If c_i and c_j are both non-zero but c_{ij} is zero, w_{ij} is set to a large negative value, $\log(1/Z)$. This also happens for β_i when c_i is zero.

A Real Time Relaxation Procedure

During learning the connectivity of the network is formed. The actual processing of information takes place during relaxation of the network, during which the connection matrix **W** is assumed fixed. A stimulus pattern is fed to the network and the relaxation starts. Unit states are iteratively updated and support is distributed via connections. In the present model, conduction delays are assumed to be shorter than the time for one iteration. The relaxation process terminates when a stable state is reached. As already mentioned, since the connection matrix **W** is symmetric, convergence is guaranteed.

The input via connections to a unit accumulates in its support value. In our model, the output is not calculated directly from the support but from a *dynamic* support variable E. The differential equation for E is

$$\frac{dE_i}{dt} = \frac{s_i - E_i}{\tau_E} \,. \tag{11}$$

This makes a unit operate as a one-compartment "leaky integrator" model neuron which to some extent mimics the passive membrane dynamics of real neurons. The E value then corresponds to the overall transmembrane potential of the neuron. Such elements have been used for quite some time in neural modeling work[17] and were also later on introduced in more abstract models with non-spiking units.[14]

Input-output relation of units

The output of a unit is calculated from its E value. The input-output relation is naturally derived from the above learning rule. It seems reasonable to take the position that the output of a unit i should signal the *a posteriori* probability of observing the corresponding event. Ideally, according to (8), the support of unit i is the logarithm of this probability. Thus, the output of i should be calculated as the exponential of the support. Since in reality the independence assumptions made above are often violated, making support values become greater than zero, we must take special care to restrict the output to [0, 1]:

$$\pi_i = \begin{cases} 0 & E_i \leq \theta_i \\ e^{E_i} & \theta_i < E_i \leq 0 \\ 1 & 0 < E_i \end{cases} . \qquad (12)$$

In a resting system the support and E values are both equal to the logarithm of the *a priori* probability of the unit. Simulations have indicated that all units then have to be silent. This is obtained by letting each unit have a threshold equal to its bias value, i.e.

$$\theta_i = \beta_i . \qquad (13)$$

The input-output relation of a unit is shown graphically in Fig. 1.

Finally, the graded output values enter naturally into the relaxation scheme by generalizing the Bayesian learning rule given above to continuous units as suggested by (6):

$$\log p(q \mid A) = \log p(q) + \sum_h \left[\log \frac{p(q \mid h)}{p(q)} \right] \pi_h . \qquad (14)$$

Implementation

Simulation of the graded model could be quite time consuming since it comprises a set of coupled differential equations with discontinuities at unit thresholds. We will therefore, in the following, restrict ourselves to the study of a finite difference approximation of the true continuous system implemented as described below.

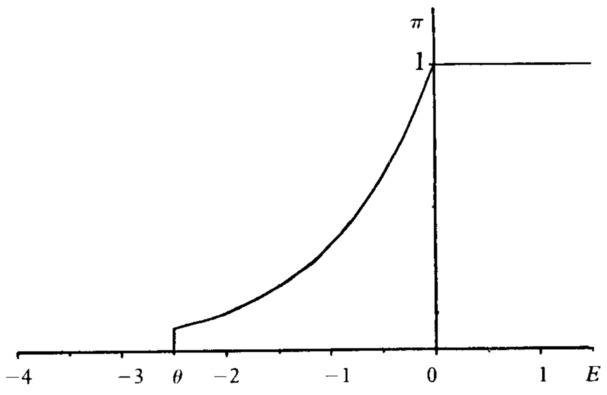

Fig. 1. Input-output relation for the units in the model. For E values between the threshold θ and 0.0 the output is the exponential of the E value. The threshold is individual for each unit according to formula (13).

In the relaxation procedure put forward here, unit outputs are restricted to make only small changes during each iteration. We regulate the time step such that the largest change is below a certain value, $\Delta\pi^{\max}$. This restriction may be violated when a unit passes its threshold. The change of its output is then equal to the *a priori* probability (frequency of occurrence) of the unit which may be larger than $\Delta\pi^{\max}$. A very small time step is made as soon as some unit passes its threshold during an iteration.

Typical values for τ_E and $\Delta\pi^{\max}$ in the simulations described below are 10 ms and 0.03, respectively. Both during learning and relaxation, the stimulus pattern is held constant for ten time constants (τ_E) during which time the network settles down. During this phase, internal connections are disabled. In the case of learning, modification of connections is then made whereas during relaxation, the internal connections are enabled and relaxation resumes. One shortcut possible in the implementation is to set E values directly from the support.

Another essential implementation feature is the *incremental* updating of the network. The connections of a sending unit are traversed only when the unit output value changes. Since the interval of E values where this happens is rather narrow, most units maintain their output constant from one iteration to another. This arrangement saves a lot of computing time when patterns have only a few active units as they usually do in our simulations (see below).

Some Simulation Results

We will here compare the old binary model and the new one with graded connections and units with respect to storage capacity, noise tolerance and computational complexity in the CAM task. We will also illustrate capabilities of the new model that go beyond those of the binary one. Furthermore, the real time character of the model also allows a study of reaction times in different situations. Results illustrating this point will also be presented.

CAM storage capacity

We have used a pure pattern completion version of a CAM task in which a number of binary random patterns ("prototypes") each comprised of m active units are first learned by the network. Thereafter, $m/2$ of the units in one stored pattern are stimulated and the ability of the system to recall the missing $m/2$ units and no others is checked. The error is calculated as the relative Euclidean distance between the prototype and the unit outputs in the recalled pattern. The number of active units in each pattern is a free parameter. In an earlier study $m = 6.25 \log_{10} N - 2.75$ was nearly optimal for the binary model.[13] We still use this result here (with a correction such that m is always even).

A number of simulations with networks of different sizes, N, were performed. For each value of N we determined how many random patterns could be stored without getting a mean error greater than 5% (averaged over twenty test patterns from the training set). The relation between the storage capacity determined in this manner and the size of the network is shown in Fig. 2.

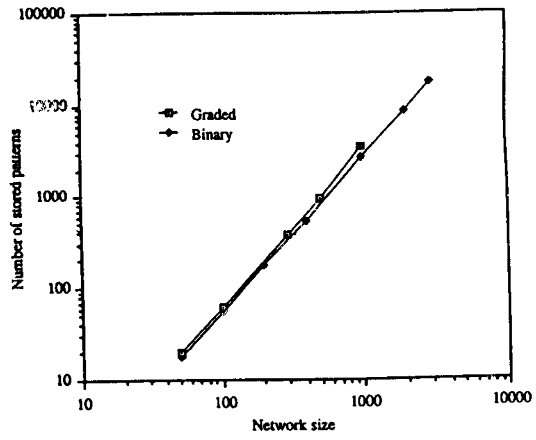

Fig. 2. The number of random patterns which networks of different sizes can store. Pattern completion from half the pattern was used as test. The lower curve shows results from an earlier binary model. The upper curve shows the capacity of the model presented here. The number of active units in each pattern for net sizes 50, 100, 300, 500 and 1000 was 8, 10, 12, 14, and 16, respectively.

It can be seen that the new learning rule actually gives a higher storage capacity than the old one. However, since the number of bits used is larger (we use 32 bits/connection but could probably do with much fewer) the ratio between physical memory and information retrieved is now lower. More importantly, these results show that we have succeeded in our aim expressed above, to design a model with graded connections and units which maintains the properties of an efficient associative memory. In fact, the learning rule put forward here gave superior performance in the CAM test problem compared to several alternative rules for setting up graded connections that we tried out in the course of this investigation.

Noise tolerance

In the earlier study of the binary model we also investigated the noise tolerance of the recall process. A distorted input pattern was presented to the network and its ability to recall the correct pattern among the ones learned was checked. This was quite straightforward with the binary model since we knew the correct answer for every input. As long as it was not saturated by storing too many patterns, this network operated as a Hamming net recalling the memory with the smallest Hamming distance to the input. In the case of graded connections we no longer have a simple Hamming matching network. The system now performs something much more like a maximum *a posteriori* (MAP) decision as to which of the learned patterns is the most likely one to have generated the input.[18] To illustrate the level of noise that can be tolerated in the recall process by the new model we give some examples.

Noise was introduced in two different ways. The first one gives binary distorted patterns. Starting with the active units in one of the patterns stored, a fraction p is randomly selected. These units are kept with the probability m/N and removed otherwise. A similar random selection is made among the inactive units. Here, units are activated with the same probability. In Fig. 3a we show the results for a network with one hundred units. The training set contained forty random patterns with ten units active in each plus the patterns "2" and "4" (resembling the corresponding digits). Figure 3a shows an example where "2" was distorted as above with p equal to 0.3. The reconstruction capability demonstrated is representative for the other stored patterns as well. The values displayed in this and the following similar figures are belief values, i.e. the exponential of the support value bounded upwards at 1.

In a second example, continuous valued noise with normal distribution was added to the input. The actual input to a unit was the confidence value for the hypothesis that, in the undistorted image, the input to the unit would have been one. Figure 3b shows the result when noise with $\sigma = 0.6$ was added to "2".

The results demonstrate that the recall process has a high tolerance towards both kinds of noise in the input. We have also studied learning in the presence of noise with interesting results. In this case, instead of showing the network the prototype patterns themselves, only a number of instances generated by adding noise to the prototypes were presented. It was shown that the prototypes could be extracted given a sufficient number of instances.

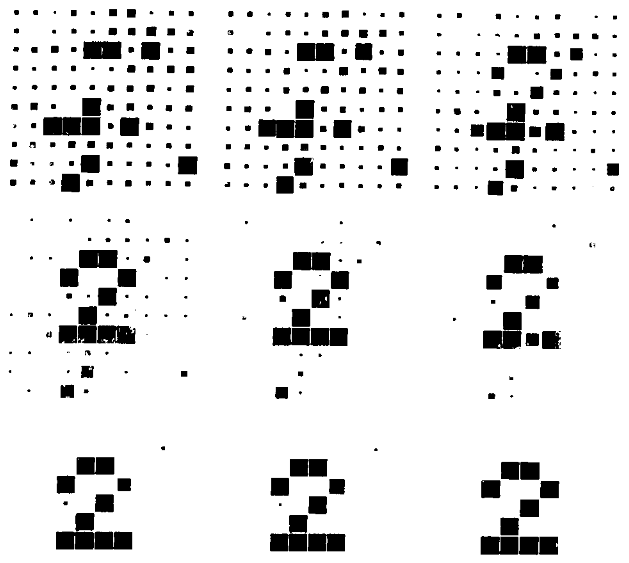
Fig. 3a. Relaxation with binary noise ($p = 0.3$) in the input. The input is shown in the upper left frame. Five iterations were performed between successive frames. The area of each square is proportional to the belief of a corresponding unit.

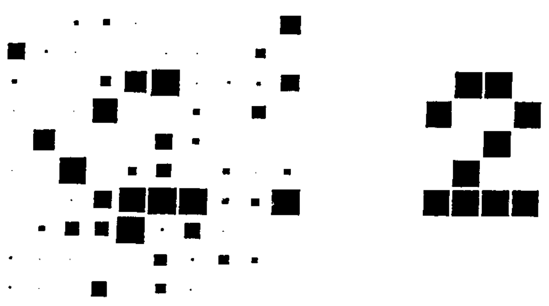
Fig. 3b. Relaxation with normal distribution noise ($\sigma = 0.6$) in the input. The actual input to the network was the confidence that the unit was one.

Competing interpretations

Another important issue concerns competition between interpretations when ambiguous input patterns are fed to the network. This demonstrates why graded weights and outputs are useful. In Fig. 4 the effect of giving a mixed input is illustrated. The "4" is the preferred interpretation since it is kept together by stronger connections, i.e. it represents an internally more consistent interpretion. This difference depends on the overlap with the other forty random patterns in this particular training set. This also shows up in the fact that the mean support of the activity pattern "4" was higher (6.4) than in "2" (5.5).

When the input to all units in "4" is gradually decreased, a sudden shift in the interpretation occurs (Fig. 4b). The stimulus of units common to "2" and "4" is the geometrical mean of the two stimulus strengths. In another experiment we wanted to show that a number of fully stimulated units can be overcome by more numerous but less potently stimulated ones. Thus, three units from "4" were fully stimulated. All the "2" units, including those common to both patterns, were given less stimulation. If the "2" units were stimulated with a strength below about 0.4 the "4" interpretation dominated (Fig. 4c), but above 0.4 the "2" pattern was recalled.

There is still another quite different way of shifting the balance between patterns. During learning the κ parameter can be used to weight a learning event. Thus we learn "four" with a lower κ which can signify, for instance, that we are uncertain during learning about the input pattern or that we want to discount it for some other reason. An example of this is when the "4" is given a weight during learning which is 0.5 compared to 1.0 for "2". Simultaneous stimulation of both patterns now yields "2" instead of "4".

Here we also want to illustrate another effect of graded connections, namely that units will take on different degrees of specificity depending on how many prototyes they participate in. Some features are shared among many prototypes and are thus less discriminative, i.e. more unspecific.

A data base containing a description of sixty different animal species was generated from a hierarchical description of the animal kingdom. There were some ten to fifteen characteristics for each species, 163 characteristics (including 60 species names) altogether. Each property (including the name of the species) was given a unique unit to represent it. The species descriptions was then learned by a network

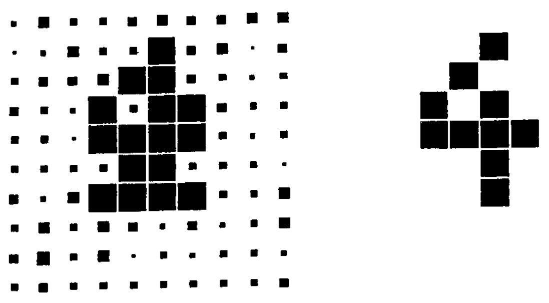
Fig. 4a. Relaxation with both "2" and "4" as input. "4" wins since it is the most tightly connected activity pattern, i.e. the most "internally consistent" interpretation.

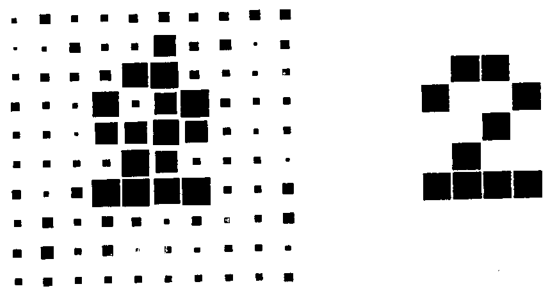
Fig. 4b. The input to "4" is lowered to 0.6 for each unit. Now the "2" interpretation is preferred.

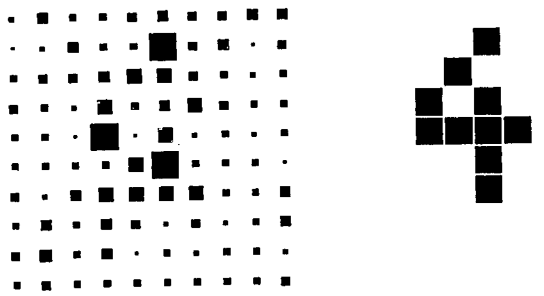
Fig. 4c. Effect of graded stimulation. Three units from "4" and all from "2" were stimulated with 1.0 and 0.3, respectively.

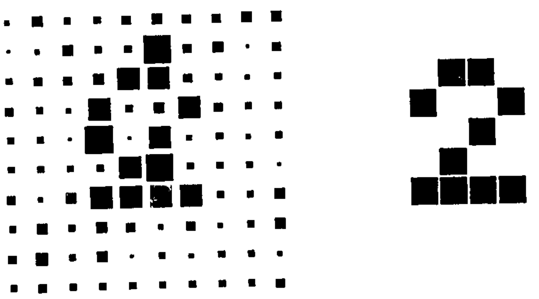
Fig. 4d. The same as 4c but the "2" units were stimulated with 0.6.

with 163 units. In the recall phase, a set of characteristics was given as input to the network. Relaxation was then performed in which, in the great majority of cases, one species was recalled. The following example illustrates the effect that some characteristics are specific, others more common.

In the data base, bear and lion share the characteristics moving, spine, fur, and eats-animals. The bear has several characteristics that the lion does not, for instance, brown and robust and the lion has yellow and tail which the bear does not. If we feed the input moving, spine, fur, eats-animals, tail, and robust (which fits no animal description in the data base) to the network the bear description is recalled. The property tail is at the same time deactivated. In our data base, the property robust occurs in only five species whereas nineteen species has tail. Consequently, robust is a more specific piece of information than tail and the bear interpretation is preferred. However, if we instead give moving, spine, fur, eats-animals, yellow, and brown the lion is recalled. In this case, yellow with four occurrences is more specific than brown with seventeen. By lowering the simulation on yellow we indicate an uncertainty in this piece of information. We found that for a stimulation of 0.7 or lower for yellow, the bear again became the preferred interpretation.

Reaction time

Reaction time experiments are common in psychology. Since the model presented here is a real time model it might be interesting to look into how reaction time is affected under various conditions. Reaction time is defined here as the time from the start of the relaxation until the output of units reach their final state. The actual relaxation time also includes the time for E values to settle down, which is longer. The questions addressed here were:

(i) Do we get a faster decision with more information given?

(ii) How do conflicting inputs affect reaction time?

(iii) What is the effect of relatively non-specific facilitation on reaction time?

Figure 5 shows the effect on reaction time when more and more of a pattern is provided as input. In Fig. 6 the effect of conflicting information and competing interpretations is illustrated. Here the input is produced by mixing two of the patterns learned. These particular patterns had no units in common. The input pattern always had ten active units. The abscissa gives the number of units originat-

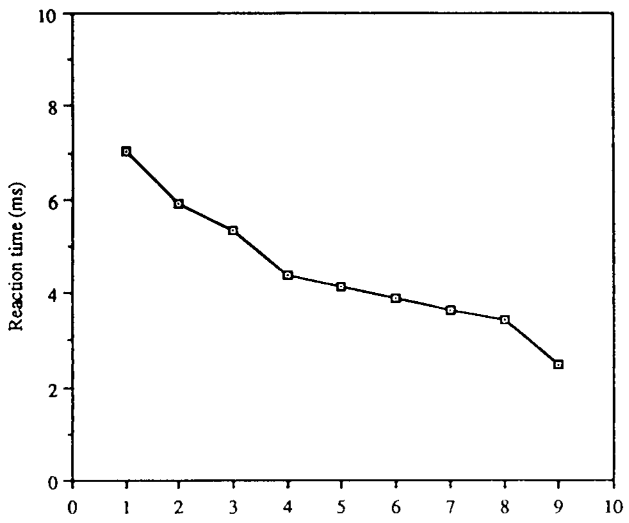

Fig. 5. Reaction time in milliseconds when more and more of an input pattern is given. The prototype was comprised of ten units. The abscissa gives the number of units from the prototype given as input.

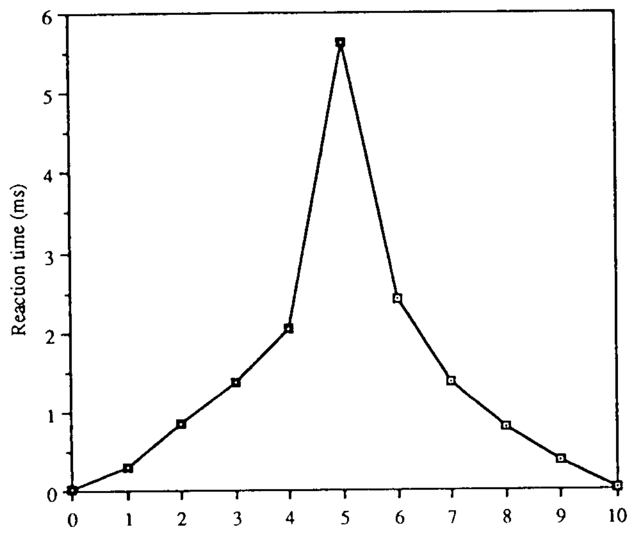

Fig. 6. Reaction time in milliseconds when a mixed input pattern is gradually moved from one prototype to another. Prototypes were comprised of ten units each. The abscissa gives the number of units from the second prototype mixed into the first prototype pattern. The prototype recalled is indicated by the markings (filled for the first, outline for the second).

ing from the first pattern and the rest were taken from the second one. Black and outline markings indicate when the first and second pattern, respectively, was retrieved.

Occasionally, when the balance between the two patterns was very even, the peak reaction time observed was two or three times as high as is shown in Fig. 6.

Our third question concerns the effect of non-specific facilitation on reaction time. Here we composed a pattern f that recalled "4" with a reaction time of 0.018 s. When we stimulated f together with weak stimulation on the entire "4", reaction time was reduced to a third of this. When we instead stimulated f and four other patterns in the training set, reaction time increased by a factor of more than three. In all cases the final result was the same, i.e. "4" was recalled. Stimulating only the four extraneous patterns recalled one of these. This can be interpreted such that supporting and conflicting information may affect reaction time by shortening and prolonging it respectively, without actually changing the outcome of the relaxation. These results indicate that the difficulty in making the decision as to which memory to recall shows up in the reaction time. The more difficult the decision the longer the reaction time.

Scaling of learning and relaxation

The scaling properties of the model, i.e. how requirements for computational capacity and memory increase with the size of the network, N, is an important question. Since real neural networks are extremely large this is important from the point of view of biological plausibility. Furthermore, the majority of serious technical applications of artificial neural networks are likely to involve the use of large systems with many thousands of units. In this case, too, scaling properties are critical. For obvious reasons, we also have to take the possibilities for parallel computation into account here.

If we first consider "one shot" learning of prototype patterns of size m, which is proportional to $\log N$, we know that the number of counters in the **C** matrix that must be updated for each prototype is $m^2/2$. Thus, the overall asymptotic complexity is $O[\log^2 N]$. The operations are entirely independent so given at least $O[\log^2 N]$ processors learning time is independent of N.

The computational demands of relaxation is somewhat harder to determine. We will here only give an estimate of the average case, partially based on simulation results. Each iteration involves updating E values for all units, calculating new output values and distributing support via connections. Updating E values and outputs are both $O[N]$ operations. Since iterations are incremental, support is distributed only via connections from units that really change their output. For these we have N connections to traverse.

The interval of E values, between the threshold and 0, in which units have $d\pi/dE \neq 0$ is fairly narrow. It is crucial to estimate how many units on the average are in this transient state. Empirically, it turns out that the units in a network are quite early split up into two groups. One is the bulk of units that ends up fully off, and the other is a small set of units belonging to the prototype recalled, that ends up fully on. During relaxation, the former group is quickly pushed towards negative E values by the latter one that is becoming more and more active in small steps.

A pessimistic estimate is that all the m, $O[\log N]$, units in the recalled prototype change their output in each iteration. Each unit has N connections whereby the computational load due to support distribution via connections becomes $O[N \log N]$. This overtakes E value and output updating and is, in fact, the overall complexity of the entire relaxation, since simulations indicate that the number of iterations is independent of N, being controlled only by the free parameter $\Delta\pi^{\max}$. In fact, operations in one iteration are independent and can thus, in principle, be carried out in parallel. This would make relaxation time independent of N. The limiting factor then becomes the dynamics of the units and conduction delays in the network.

Memory requirement of a fully connected network is obviously $O[N^2]$. However, large networks are likely to contain many uncorrelated pairs of units which will need no connection between them. This means that the **C** and **W** matrices could be sparse. The amount of memory required then depends on how many prototypes one likes to store. That information storage capacity per bit of physical memory is theoretically still maintained at a reasonable level also for sparse networks has been shown.[19]

Discussion

We have presented here a one-layer, feedback-coupled neural network model which is derived from an earlier model with binary connections and units. The new model employs graded connections and units, a Bayesian learning rule and a real-time, parallel relaxation procedure. It belongs to the same category of artificial neural networks as e.g. Anderson's "Brain-State-in-a-Box" model and the Hopfield model. In particular, it is similar to the continuous Hopfield model in that the units have dynamic properties reminiscent of the passive membrane properties of real neurons.

Our model is comparatively free from parameters and *ad hoc* assumptions, since, apart from providing updating rules for the connection matrix, the Bayesian learning rule also gives the bias values and the input-output relation of the units. The remaining parameters are the time constant τ_E and $\Delta\pi^{\max}$. The time constant can be chosen quite arbitrarily since it only determines the time course of the relaxation and not its trajectory through state space. The value of $\Delta\pi^{\max}$ is merely a tolerance parameter and it should, in principle, be made as small as possible. In practice, values below a certain limit give identical results.

The storage capacity of the model when it operates as an autoassociative, content addressable memory is comparable to the binary model. It is important here that, provided a suitable representation, the number of items that can be stored and retrieved by associative recall can be much greater than the number of units in the network. In the derivation of the learning rule some independence assumptions are made. In practice, violation of these assumptions are unavoidable. Yet, in the tasks we studied this caused no great problems. However, things might change when input patterns spread out less evenly than with the random patterns used here. We have also demonstrated that this graded model is qualitatively more capable than the earlier binary one in several respects. For instance, different units may take on different specificities depending on their frequency of occurrence in the training set and graded stimulation and uncertain information is treated appropriately.

With respect to noise tolerance, the model is capable of pattern completion as well as removing units from the input pattern to reconstruct the most similar of the memories stored. Instead of making a Hamming matching between input pattern and stored memories, something like a maximum *a posteriori* (MAP) decision is made regarding which object was the most likely one to have generated the input pattern. Interesting effects on reaction time for recall were noticed. For instance, in several examples reaction time seemed to be directly related to the difficulty in making the decision as to which memory to retrieve.

Complex units

When given an appropriate representation, e.g. via an adequate set of feature detectors, a one-layer, feedback model performs quite well. However, if the representation is inappropriate, function degrades due to cross-talk between the items to be stored. The need for an internal mechanism to automatically improve, i.e. orthogonalize, representations becomes acute. In an earlier model we tried to deal with this problem by providing a mechanism that generated complex units by replacing the original "feature detectors" one by one until a test criterion was met.[15,20] Augmented in this way, this model was capable of solving the 4-bit ADDER problem. The number of learning epochs (passes through the entire training set) required to solve the 2-bit ADDER problem was around ten which is considerably less than the more than 3000 epochs required by a multi-layer feedforward network using back-propagation.[12] Our model is in many respects quite different from back-propagation, e.g., in that it is a recurrent network and utilizes only global error information in learning. At present we are modifying the scheme for generating complex units to suit our new graded model. Alternative solutions to this problem are also considered. If a satisfactory solution can be found, the functional capabilities of one-layer feedback and multi-layer feedforward models will be combined in the same artificial neural network (ANN).

Biological plausibility

When discussing the biological relevance of present neural network models one must bear in mind that they should not be expected to reach anywhere near higher nervous systems in terms of level and multiplicity of function. To do this we at least have to design artificial neural systems (ANS) comprised of many interacting ANNs. A complete ANS should have sensory, motor and integrative sub-systems as does the brain. We certainly have a long way to go before reasonably competent systems of this sort could be made reality. For the moment, our primary objective is to design ANNs that are sufficiently capable to serve as a module in such a construct.

Of course, all computational models today have to go much beyond what can be verified by neurobiology and can at best be considered as first approximations. A reasonable minimal requirement is that the models may not violate what is plausible from a neurobiological point of view. The biological plausibility of neural network models, in particular feedforward, multilayer models, applying back-propagation has recently been discussed by Crick.[21]

The model proposed here is essentially one out of may possible mathematical formulations of Hebb's theory of cell assemblies. One component in the judgment of the biological plausibility of a model is the input-output relation of its units. In fact, real neurons show quite a bit of variability here. Often one observes an initial period of transient high frequency firing that levels off to a lower sustained firing rate. The input-output relation of our model is likely to serve the purpose of first approximation as well as any sigmoid or linear function. The incorporation of membrane dynamics brings the model fairly close to those used in more realistic modeling work.[11,17]

In at least one respect we note an obvious deviation from biology. This concerns the fact that the connections in our model change readily between being excitatory and inhibitory. However, this is a contradiction only if we interpret a unit strictly as modeling one neuron. If we modify our model slightly by splitting up each unit into two as suggested in Fig. 7 the problem can be circumvented. Here, inhibition is passed on via the inhibitory unit which inverts the signal. This unit has a zero threshold and a linear input-output relation. The excitatory connections making contacts on this unit are modifiable according to the Bayesian learning rule although with inverted sign. Connections are assumed to disappear when they are driven to negative values. Moreover, each unit in Fig. 7 might in reality correspond to several neurons.

In other respects, the learning rule suggested here is quite plausible from a physiological point of view. Rules for synaptic modification are often discussed in terms of correlations and conditional probabilities with respect to pre- and postsynaptic activity.[22]

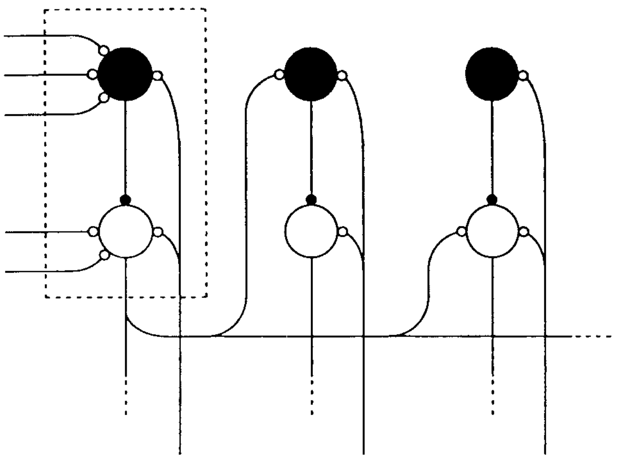
Fig. 7. The model set up with artificial neurons to avoid connections that change freely between being excitatory and inhibitory. Inhibitory neurons have a threshold of zero and act merely by inverting the sign of the input. One model unit consisting of two artificial neurons is surrounded by a dotted rectangle. External stimulation to the unit driving both types of neurons enters from below.

Finally, the assumption of full connectivity is not very biological. It is also technically unfeasible when large systems are considered. The learning rule proposed here results in zero connections between units having statistically independent activities. This gives the possibility to remove such connections without any degradation in performance, either because we know beforehand of such independencies or because they are discovered during learning. We are currently investigating the performance of our model when connectivity is limited. Other essential extensions of the functional capabilities relating, e.g., to temporal association and short-term memory are also considered.

Acknowledgment

Support for this work from The National Swedish Board for Technical Development (STU, grant no: 87-01224, and STUF, grant no: 87-0321) is gratefully acknowledged.

References

1. D. O. Hebb, *The Organization of Behavior*, John Wiley, New York, 1949.
2. T. Kohonen, "Correlation matrix memories", *IEEE Trans. Comput.* **C-21** (1972) 353–359.
3. T. Kohonen, *Self-Organization and Associative memory* (2nd Ed.), Springer-Verlag, 1988.
4. J. A. Anderson, "A simple neural network generating an interactive memory", *Math Biosci* **14** (1972) 197–220.
5. J. A. Anderson, J. W. Silverstein, S. A. Ritz and R. S. Jones, "Distinctive features, categorical perception, and probability learning: some applications of a neural model", *Psychol Rev.* **84** (1977) 413–451.
6. D. J. Willshaw, O. P. Buneman and H. C. Longuet-Higgins, "Non-holographic associative memory". *Nature* **222** (1969) 960.
7. G. Palm, "On associative memory", *Biol. Cybernetics* **39** (1980) 19–31.
8. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. USA*, Vol 79, (1982) 2254–2558.
9. N. Rochester, J. H. Holland, L. H. Haibt and W. L. Duda, "Tests on a cell assembly theory of the action of the brain using a large digital computer". *IRE Trans. Information Theory, IT-2*, 1956.
10. R. J. MacGregor and T. McMullen, "Computer simulation of diffusely connected neuronal populations", *Biol. Cybernetics* **28** (1978) 121–127.
11. A. Lansner, "Information processing in a network of model neurons", 1982, Tech. Rep. TRITA-NA-8211, Dept. of Numerical Analysis and Computing Science, The Royal Institute of Technology, Stockholm, Sweden.
12. D. E. Rumelhart, G. Hinton and R. J. Williams, "Learning internal representations by error propagation" in D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, Vol. 1, 318–362, MIT Press, Cambridge, 1986.
13. A. Lansner and Ö. Ekeberg, "Reliability and speed of recall in an associative network", *IEEE Trans. Pattern Analysis and Machine Intelligence* 7(4) (1985) 490–498.
14. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Natl. Acad. Sci. USA*. **81** (1984), 3088–3902.
15. A. Lansner and Ö. Ekeberg, "An associative network solving the "4-bit ADDER problem", *Proc. IEEE First Ann. Intl. Conf. Neural Networks*, pp. II-549, San Diego, USA, June 21–24, 1987.
16. Ö. Ekeberg, "An incremental learning rule for an artificial neural network", 1989 (in preparation).
17. R. J. MacGregor and R. M. Olivier, "A model for repetitive firing in neurons", *Kybernetik* **16** (1974) 53–64.
18. R. Golden, "A unified framework for connectionist models", *Biol. Cybernetics* **59** (1988) 109–120.
19. G. Palm, "On the storage capacity of an associative memory with randomly distributed storage elements", *Biol. Cybernetics* **39** (1981) 125–127.
20. Ö. Ekeberg and A. Lansner, "Automatic generation of internal representation in a probabilistic artificial neural network", in *Neural Networks from Models to Applications*, eds. L. Personnaz and G. Dreyfus, (178–186, I.D.S.E.T., 1989, Paris).
21. F. Crick, "The recent excitement about neural networks", *Nature* **337**(12) (1989) 129–132.
22. W. B. Levy and N. L. Desmond, "The rules of elemental synaptic plasticity", in *Synaptic Modification, Neuron Selectivity, and Nervous System Organization*, eds. W. B. Levy, J. A. Anderson and S. Lehmkuhle, (Lawrence Erlbaum, Hillsdale, New Jersey).