

Memory Model with Unsupervised Sequential Learning: The Effect of Threshold Self-Adjustment

Anatoli Gorchetchnikov

Department of Computer Science

Middle Tennessee State University, Murfreesboro, TN 37132

ag2a@frank.mtsu.edu

Abstract. The goal of this research is the creation of a neural net-based autonomous AI system. Such a system, to be successful, should satisfy the following requirement: the learning must be unsupervised, consecutive and depend only on locally available knowledge. Adaptive Bidirectional Associative Memories (ABAM) provide unsupervised learning and flexibility of architecture, but they suffer from catastrophic interference when attempting to add some new knowledge after the completion of the initial training. The solution to this problem for back-propagation (BP) and Hopfield type nets is a pseudorehearsal of the base knowledge (Robins, 1998). This research applies this method to ABAM with attempt to pseudorehearse not the whole base population, but only the neighborhood of the new incoming knowledge. This allows the creation of pseudopatterns within the system without intervention of the experimenter, but requires the extension of conventional neural nets, which are currently viewed as sets of uniform simple units. The basis for this extension is provided by using the Cohen-Grossberg activation dynamics (Cohen & Grossberg, 1983) for the units and a SIMD computer that allows the user to change the parameters in this formula for a specific group of units in the net. Custom software has been written to achieve the required flexibility. The conceptual description of this software and parameters of the model are presented in this paper. Each of those parameters requires individual study; this paper focuses on threshold self-adjustment, which makes each unit more responsive to the change of input and provides additional sensitivity to the net.

Introduction

An autonomous AI agent requires the memory system that allows it to act and learn independently. If we choose neural nets as an implementation of an agent's memory, then from all known types of unsupervised nets, multilayer ABAM provides the most flexible and problem-adjustable architecture. Before we start modeling, we should consider the following problem: the ideal memory system should be stable enough to preserve the important information over time on one side, but sensitive enough to incorporate a new knowledge on the other. ABAM, as well as other neural networks, usually are sensitive to new knowledge, but is not stable in the sense that this new information in some cases severely disrupts the previous knowledge. This problem (known as catastrophic forgetting, catastrophic interference or the sequential learning problem) led to the limitation of existing neural network based AI systems to concurrent learning, when all information has to be presented at once before the system starts its functioning. Industry would, however, prefer an AI system that can learn while acting without complete retraining each time new information becomes available. Catastrophic forgetting also attracted a lot of criticism to connectionist cognitive modeling since such behavior of a neural net appeared to be not human-like.

A number of studies recently attempted to solve the problem of consecutive learning, and most of them used multi-layer perceptrons with back-propagation (Robins, 1998). The first solution, suggested by French (1991), involved reducing the overlap between the distributed representations in the hidden layer. French used "activation sharpening" to reduce this overlap: activations of the most active units were artificially increased by a small value, while activations of the less active units were

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

'99 ACM Southeast Regional Conference

©1999 ACM 1-58113-128-3/99/0004 5.00

decreased by a small value on every iteration of learning. French (1994) improved this method adding the orthogonalization of hidden representations and called it "context-biasing". Context biasing reduced the catastrophic forgetting, but did not solve the problem completely.

The general idea for the solution was proposed by McClelland, McNaughton & O'Reilly (1995). They pointed out that in the human brain the interaction of the two complementary memory systems provides interleaved learning without catastrophic forgetting. The key point is that the hippocampal area is involved in reinstating new memories while neocortex is reinstating old ones, thereby providing gradual consolidation of the new information into an existing knowledge system.

Following these guidelines several models have been built based on the interaction between two neural networks. Gluck and Myers (1997) used two back propagation (BP) networks in their cortico-hippocampal model, with the transmission of the hidden representation of new knowledge from the hippocampal network to the neocortex. Another such model was the interactive tandem network by French (1995). In his model, the hidden representation of the short-term network served as a target output for the long-term network. The actual output on its turn influenced the hidden representation back in the short-term network. Both those models are not biologically realistic since there is no mechanism for such transfers of activation between units.

Two more realistic ways to implement this kind of learning involving reinstatement of memories without activation transfer are rehearsal and pseudorehearsal were discussed in Robins (1988). Rehearsal methods are based on the retraining of the old knowledge items as the new items are learned. Robins (1995) argued that only 15% of old patterns have to be reinstated to provide the acceptable results. In his "sweep" method those old patterns were selected randomly for each epoch of training. But even for random selection, all old items should be "available on demand" (Robins, 1998). To avoid this restriction he introduced pseudorehearsal (Robins, 1995). In spite of actual old patterns, a set of pseudoitems is created for each new item to be learned. The mechanism is simple for BP

nets: a set of random inputs is presented to the net and each of those inputs is associated with generated outputs. For the Hopfield net, Robins & McCallum (1998) suggested the use of stable state (SS) pseudoitems. They allowed the random input first to settle down in a stable state (attractor), and then used these attractors as pseudoitems. In both cases the image of the general knowledge of the net was created for rehearsal in spite of reinstating the particular knowledge patterns.

The example of the BP pseudorehearsal model is a pseudo-recurrent network (French, 1997). Learning in this model consists of two phases: mixing of the new pattern with the set of pseudopatterns created from random input that occurs in the "early-processing" part of the net, and learning the resulting set by the "final-storage" area. This model (unlike others e.g. Robins (1995)) creates pseudopatterns within itself, but still it is not suitable for an autonomous AI system, since BP is used as a learning algorithm. In this sense models based on the associative memories and hebbian learning algorithm are favorable. The only application of pseudorehearsal to these nets was discussed in Robins & McCallum (1998). They had a base population of 40 items, and in order to learn one more pattern, the set of 256 pseudoitems was created. This number of pseudopatterns appears too large for the size of the problem, especially considering that my pilot study showed only the selective disturbance of the previous knowledge based on the level of contradiction between the existing knowledge and the new pattern.

Those contradictions disturb mostly the knowledge that is in some sense related to the new pattern.¹ Therefore, the natural mechanism for creating pseudoitems evolves (see Figure 1). The net consists of three subnets: input area, short-term area (STM) and long-term area (LTM), containing the previously learned knowledge. Each of those subnets is a multilayer associative memory. When the input

¹However, in personal communication Robins pointed out that he conducted the similar study and did not find selective disturbance. This question requires more research, and the suggested model will be one of the subjects of this study.

pattern is presented to the net, it takes some time for the net to settle down and produce the output association. This delay is critical for the functioning of the model. Suppose that a new pattern arrives in the input area, and after some delay is propagated to both STM and LTM. In STM, it is learned quickly because of the high learning rate, and then this subnet switches to recall mode. The mechanism of this switch is discussed by Hasselmo & Cekic (1996). They showed that for effective performance an associative net has to suppress feedback connections during the learning mode, and enable them during the recall mode.

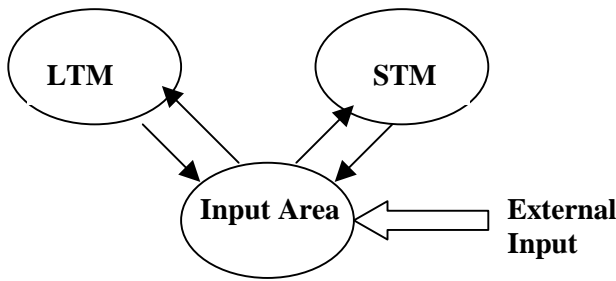


Figure 1: Proposed Architecture of the Net.

On the next step this newly-learned pattern is sent back to input area and starts to circulate between input area and STM providing its own reinstatement. Meanwhile, in LTM the pattern (with some random distortions to explore the wider neighborhood) raises the set of associations that are related to it, and those associations start their circulation between the input area and LTM. This provides the mechanism for reinstating the old memories during training. A small learning rate in LTM increases the effective sample size by forcing a network to run through a larger number of associations (McClelland, McNaughton & O'Reilly 1995). To allow the proper circulation of knowledge, this net requires precise timing and synchronization. Moreover, this timing should be different for STM and LTM areas. In the brain this difference exists due to specific electrical parameters of cells in various areas (see McKenna et al, 1992). The number of those parameters in biological neurons is huge, and the purpose of this study is to figure out which of them are relevant to the functionality of a “complementary memory system” (McClelland, McNaughton and O'Reilly, 1995).

Model Description

Hardware

Following the brain structure, on one side our units should be similar to one another in functionality as well as in structure. On the other side, the parameters of those units should be subnet-specific. This requires the extension of the conventional basic neural net approach; i.e., the net can not be a set of simple uniform units any more. A natural choice for the implementation of a neural net with distinct features of specific units is the MasPar MP1 parallel SIMD computer. It provides the possibility of running a single algorithm on up to 2048 processors in synchronous parallel mode. It also allows the parameters of the operations to be processor-specific and to vary them for different functionality of different subnets. Since all computation occurs in parallel, the model can also capture the specific timing features.

Activation function

The analysis of the mathematical foundations for conventional neural networks was presented by Kosko (1992). Neural network is a dynamic system; time is an important factor for the analysis of its behavior. There are two levels of dynamics: fast activation changes and slow synaptic changes, which differ in the levels of magnitude in their speed. The general formula for the activation function in associative nets is the Cohen-Grossberg activation dynamics (Cohen & Grossberg; 1983):

$$\dot{x} = -a(x) \left(b(x) - \sum_j S(y_j) m_j \right), \quad (1)$$

- where m_j represents the weight in j -th connection;
- $S(y_j)$ is the signal through this connection;
- nonnegative $a(x)$ and arbitrary $b(x)$ are functions of current activation.

Depending on the nature of $a(x)$ and $b(x)$, different nets can be created. Grossberg (1988) classified all neural networks into two major groups: one with additive and another with multiplicative (shunting) dynamics. If $a(x)$ is constant and $b(x)$ is linear, then the net has additive dynamics. An example of this case is the Hopfield net, where $a(x) = 1/C$ and $b(x) =$

Ax - I. Here C serves as a membrane capacitance; A as membrane permeability; and I as external input.

For simplicity, the current implementation of the model uses these functions. If the functionality requires more complex activation, the structure of the model allows these functions to change or even to have different functions in subnets. Even in the current simple model, the generality of the Cohen-Grossberg formula provides enough degrees of freedom to vary any parameters of the specific groups of units without changing the major algorithm. Several ideas of possible parameter relationships for different cells were taken from McKenna et al (1992). Since this book contains a set of articles by different researchers, whose data relates to different areas of research, the following assumptions are by no means infallible or complete, but reliable enough to serve as a starting point for the analysis of their relevance.

Sizes of Cells

For this study the model required only two different types of cells:

1. Small interneurons (Stellate cells) do not have apical (distal) dendrites and provide communication within layer. They usually serve as inhibitors and can receive input to soma and basal dendritic tree.
2. General neurons (Pyramidal cells) with apical tree provide connections between layers. They receive input from other layers to apical dendrites and from other cells in the same layer to soma and basal dendrites.

To capture the signal travelling time, for each of those input levels – somatic, basal, and apical – a separate FIFO queue was maintained in each unit. Unit grabs a summary input for the level and puts it in a respective queue. The size of the queue is one for somatic, two for basal, and three for apical input. To simulate the voltage gradient in a dendrite the value of input decays by 50% in each step, when it is in the queue.

Permeability and Capacitance

The permeability of a soma membrane is significantly higher than that of dendrite. The time constant for distal input appears to be longer than for basal input. These assumptions led to the cell parameters in the model listed in Table 1. Since the somatic input is usually inhibitory and comes from the fast-spiking

cells, parameters for this input reflect not only the features of soma membrane, but also the shaping of those fast spikes (fast return to resting potential). That allows keeping uniform outputs for all types of units.

Type and parameter	Somatic input	Basal input	Apical input
Capacitance C (μF)	0.5	0.8	0.86
Permeability A ($1/\text{k}\Omega\text{sm}^2$)	0.8	0.375	0.285

Table 1: Permeability and Capacitance for Different Input Levels.

Threshold values

Each type of cell had a different initial threshold value. These initial values were the results of a single cell simulation on an AMD 5x86 160 MHz under Linux Red Hat. The value was set to provide the 50% probability of firing with random input. When the model operates, the threshold of each unit undergoes constant changes. If the output of the unit is higher than some level (for example 0.95), the threshold increases slightly (1%). Biologically, it can be explained with the cell running out of essential chemicals to produce neurotransmitters. On the other hand if the output of the unit is lower than some level (for example 0.05; henceforth saturation level), the threshold decreases. Since this self-adjustment of threshold was the most ambiguous part of the model, it became the subject of the following set of experiments.

Method

Network

The first step in creating the model shown in Figure 1 is to create and test each of the subnets separately. The simple net for current simulation (see Figure 2) contained two layers: layer 1 functioned as input/output; layer 2 stored the hidden representation of the pattern. Six units in the input layer and four units in the hidden layer were general pyramidal neurons. Each of them received the apical input from all units in the other layer. This major architecture replicates the net used by Hasselmo & Cekic (1996) in their study of feedback suppression. Their results serve as a control in the analysis of net learning. In

addition, in the hidden layer each of the pyramidal units served as input for one small stellate inhibitor, which receive this input in the basal level and provide the somatic inhibitory input to all other pyramidal units in the layer. This inhibition (again, following Hasselmo & Cekic, 1995) provides the “winner takes all” strategy for the hidden unit’s activation. The initial weights for all connections were randomly set in the interval $[0, 12]$. Output of all neurons on every step was recorded to a text file, and further analyzed using Microsoft Excel[®] spreadsheet.

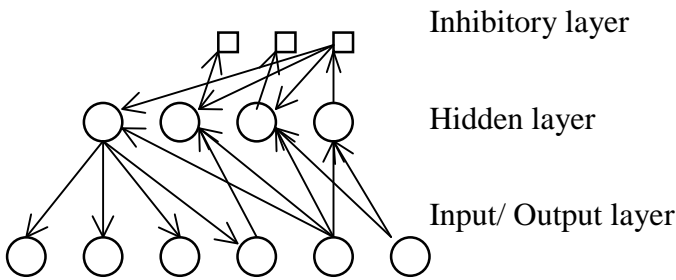


Figure 2: Architecture of the Net (most of the connections and some inhibitory units are not shown).

Procedure

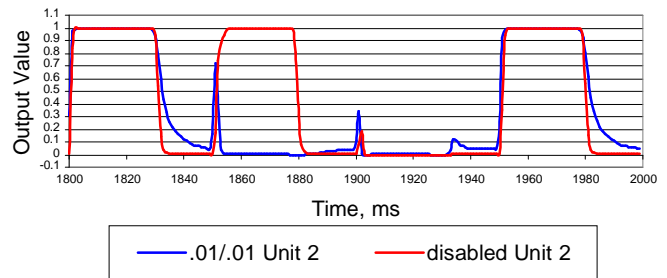
Within each trial the same set of random initial weights was used for all running conditions to compare the effects of different threshold adjustment on a specific set of weights. Each trial consisted of ten consecutive epochs with the presentation of the same set of four patterns to ensure the consistency of the result. Three major comparisons were studied. In the first part of experiment, the behavior of the net with and without threshold self-adjustment was compared. Saturation level was set to 0.01, rate of adjustment to 1% per turn. Then the study of behavior was made for different rates of adjustment: 2%, 1% and 0.5% per turn. Finally, different saturation levels (0.1, 0.05 and 0.01) were compared.

Results

Typical results for the first part of the experiment are presented in Figure 3. They show that the introduction of the threshold self-adjustment changes the behavior of the net significantly. In this example, self-adjusted threshold forced unit 4 to overtake unit 2 and fire when the second pattern was presented to the net. Usually the first epoch shows different activation

comparing to all consecutive epochs, when units settle down to some specific activation pattern and follow those guidelines without any differentiation from epoch to epoch.

Activation of Hidden Unit 2



Activation of Hidden Unit 4

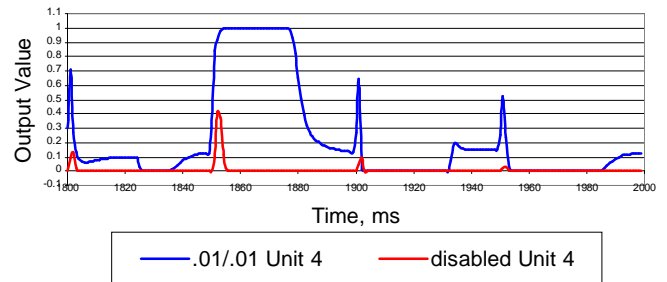


Figure 3: Activations of Hidden Units 2 and 4 with Threshold Self-Adjustment Enabled vs. Disabled (The presented graph shows the behavior during the last epoch, previous epochs are similar after system settles down during the first epoch. Numbers in the legend represent saturation level/rate of adjustment).

	Unit 1	Unit 2	Unit 3	Unit 4
Mean square difference between activations	2×10^{-4}	8×10^{-4}	8×10^{-4}	12×10^{-4}
Same as % of maximum activation	0.02%	0.08%	0.06%	0.12%

Table 2: Comparison of Activations for Saturation Level of 0.01 and Speeds of Adjustment 0.5% and 1% of Current Threshold Level.

In the second part of the experiment three different rates of adjustment – 1%, 2%, and 0.5% – were compared for the value of saturation 0.01. The

resulting activation over ten epochs for each rate was compared for each output unit. Typical mean square difference between activations for all four hidden units is presented in Table 2.

In the third part of experiment the different behavior was found for different saturation levels. Those results are summarized in Table 3.

Saturation Level	0.1	0.05	0.01
Resting level (hidden units)	0–0.02	0.05–0.12	0.05–0.18
Resting level (inhibitors)	0.1	0.05	0.04
Sensitivity (hidden units)	High	Normal	Normal
Sensitivity (inhibitors)	High	High	Normal
Threshold adjustment (hidden units)	constant	When pattern presented	When pattern presented
Threshold adjustment (inhibitors)	constant	constant	When pattern presented
Net Behavior	stable	unstable	stable

Table 3: Threshold Self-adjustment and Net Behavior as Results of Relation Between Saturation Level and Resting Output Level.

Discussion

Introduction of the threshold self-adjustment changed the behavior of the net significantly. The reason for this change is increased sensitivity of the units in the net to the change of input value. Threshold self-adjustment increases the chances for the unit inhibited for a long time to fire as well as for long-active unit to stop firing. When the new pattern arrives, those units that did not respond to previous patterns have the advantage over those that just were active, since former thresholds were decreased during waiting in inhibited state. In Figure 3 Unit 4 got better chance to respond on the second pattern, since Unit 2 has been active for a long time responding to the first pattern. The negative effects of threshold self-adjustment were slowing down the decrease of output signal after the end of pattern presentation and related non-zero activation levels in-between the patterns. Those

effects were caused by the constant input to hidden units from input/output layer. This input was absent without the self-adjustment, since the input/output units' activations decayed to zero. With threshold self-adjustment, those activations decayed only to the saturation level and the continuous activation function produces some level of output in hidden units.

The rate of threshold adjustment did not show any significant effect at least in the explored range. However, forthcoming study of the influence of threshold self-adjustment on learning can provide different results. The other significant difference is expected in modeling of hippocampal cells, since they have multi-threshold activation and switch between several thresholds that are far apart from each other (Tam, 1992).

The saturation level appeared to be important for the studied effect. Levels of 0.1 and 0.01 showed similar activation patterns but differed in the values of output in-between patterns. Saturation level 0.1 provided high level of inhibitory activation for the whole time of simulation. This inhibition forced the output values of hidden units to fall down almost to zero, and even adjustment of threshold was not able to increase these values. Moreover, inhibitors oscillated around the saturation level and also got very sensitive due to threshold decrease. In such situation, hidden units became extremely sensitive, even small increase of input to one of those units makes the instant jump of activation and following shut down of all competitors by respective inhibitor. Saturation of 0.01 led to equilibrium between hidden units and interneurons with both activations above 0.01. In this situation the adjustment of thresholds took place only when the patterns were presented, and one of the hidden units fired causing the inhibition of others. This appeared to be the best choice, since no adjustment takes place in-between the pattern presentation. Saturation level of 0.05 gave the least stable configuration. The typical example of this instability was switching between two hidden units within the presentation of the same pattern. This switch can be explained by high sensitivity of the saturated inhibitors interfering with lesser than in 0.1 case sensitivity of hidden units. Hidden units start their competition gradually increasing their output

values. Inhibitors respond to this delicate change by the energetic jump of their activation level followed by the drop when the real winner takes the lead.

Overall, threshold self-adjustment showed itself as a useful addition to network functionality, especially in the case when the appropriate saturation level provides the adjustment only when the pattern is presented to the net. The choice of this level is not trivial, since it depends on the connection weights between inhibitors and hidden units, which provide the activation equilibrium. Forthcoming research will investigate this dependency to provide the stable solution not only for possible random initial weight matrix, but also for the matrices that can emerge during learning. This, in turn, will provide the creation of stable and sensitive components for the general memory model presented in Figure 1.

Acknowledgements

Author appreciates the patience of his major professors Dr. C. Pettey and Dr. A. Cripps, who are always ready to provide competent help and/or advice. Special thanks to Dr. W. Langston, who introduced author to the problem and cognitive modeling as a field of study.

References

1. Cohen, M., Grossberg, S. (1983). Absolute stability of global formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13, pp. 815-826.
2. French, R. (1991). Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. *Proceedings of the 13th Annual Cognitive Science Society Conference*, NJ: LEA, 173-178.
3. French, R. (1994). Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference. *Proceedings of the 16th Annual Cognitive Science Society Conference*, NJ: LEA, 335-340.
4. French, R. (1995). Interactive tandem networks and the sequential learning problem. *Proceedings of the 17th Annual Cognitive Science Society Conference*, NJ: LEA, 222-227.
5. French, R. (1997). Pseudo-recurrent connectionist networks: an approach to "sensitivity-stability" dilemma. *Connection Science*, 9 (4), 353-379.
6. Gluck, M., Myers, C. (1997). Psychobiological models of hippocampal function in learning and memory. *Annual review of Psychology*, 48, 481-514.
7. Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1), 17-61.
8. Hasselmo, M., Cekic, M. (1996). Suppression of synaptic transmission may allow combination of associative feedback and self-organizing feedforward connections in the neocortex. *Behavioral Brain Research*, 79, 153-161.
9. Kosko, B. (1992). *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Prentice-Hall, Englewood Cliffs, NJ.
10. McClelland, J., McNaughton, B., O'Reilly, R. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102, 419-457.
11. McKenna, T., Davis, J., Zornetzer, S. (Eds.). (1992). *Single Neuron Computation*, Academic Press, San Diego, CA.
12. Robins, A. (1995). Catastrophic forgetting, rehearsal, and pseudorehearsal. *Connection Science*, 7, 123-146.
13. Robins, A. (1998). Solutions to the catastrophic forgetting problem. *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, NJ: LEA, 899-904.
14. Robins, A., McCallum, S. (1998). Catastrophic forgetting and pseudorehearsal solution in Hopfield type networks. *Connection Science*, in press.
15. Tam, D. (1992). Signal processing in multi-threshold neurons. In McKenna, T., Davis, J., Zornetzer, S. (Eds.), *Single Neuron Computation*, (pp. 481-502). Academic Press, San Diego, CA.