Apoptosis, Neurogenesis, and Information Content in Hebbian Networks

Christopher Crick and Willard Miranker
Department of Computer Science
Yale University
May 13, 2005
TR1322

Apoptosis, Neurogenesis, and Information Content in Hebbian Networks

Christopher Crick and Willard Miranker
Department of Computer Science
Yale University
christopher.crick@yale.edu, willard.miranker@yale.edu

May 13, 2005

Abstract

The functional significance of alternate forms of plasticity in the brain (such as apoptosis and neurogenesis) is not easily observable with biological methods. Employing Hebbian dynamics for synaptic weight development, a three-layer neural network model of the hippocampus is used to simulate non-supervised (autonomous) learning in the context of apoptosis and neurogenesis. This learning is applied to the characters of a pair of related alphabets, first the Roman and then the Greek, resulting in a set of encodings endogenously developed by the network. The learning performance takes the form of a U-shaped curve, showing that apoptosis and neurogenesis favorably inform memory development. We also discover that networks that converge very quickly on the Roman alphabet take much longer to handle the Greek, while networks which converge over an extended timeframe can then adapt very quickly to the new language. We find that the effect becomes increasingly pronounced as the number of neurons in the dentate gyrus layer decreases, and identify a strong correlation between cases where the Roman alphabet is quickly learned and cases where a few neurons saturate many of their weights almost immediately, minimizing participation of other neurons. Cases where learning the Roman alphabet requires more time lead to larger numbers of neurons participating with a larger diversity in synaptic weights. We present an information-theoretic argument about why this implies a better, more flexible learning system and why it leads to faster subsequent correlated Greek alphabet learning, and propose that the reason that apoptosis and neurogenesis work is that they promote this effect.

1 Introduction

The hippocampus, a brain structure located in the medial temporal lobe, appears to be responsible for establishing novel associations during the learning process. As the brain forms new associative memories, hippocampal neurons change their stimulus-selective response patterns [9]. This change in response patterns suggests similarities to the learning processes

of artificial neural networks. Since the detailed internal behavior of neurons *in vivo* is difficult to investigate, we seek to gain insight by studying the behavior of their simulated parallels.

Adult neurogenesis occurs in man as well as other mammalian species [5] [8]. This phenomenon appears most robustly in certain brain regions, particularly the dentate gyrus (DG) of the hippocampus and in the olfactory system [3] [7]. The functional significance of neuronal plasticity (such as apoptosis and neurogenesis in the brain) is not easily observable with biological methods [1]. Neural network simulations, therefore, can provide a salient procedure for direct analysis of learning and memory properties of plasticity in neural systems. For a review of earlier hippocampal network modeling and simulation, see [4].

In [2], computer simulations were used to study the possibility that replacing neurons could favorably impact cognition as well as a variety of other brain functions informed by hippocampal activity (e.g. short and long term memory formation, adaptations to sex and stress hormones, and various forms of mental illness). Those simulations modeled learning tasks employing a three-layer neural network model of the hippocampus. These layers modeled, in turn, the entorhinal cortex, the dentate gyrus, and CA3. The network was made to learn a representation of the Roman alphabet, the first task. Upon completion of this task, the network was made to learn a representation of the similar but not identical Greek alphabet, the second task. The postulate that neurogenesis favorably influences the second task was demonstrated.

We take information in the brain (i.e. memory traces) to be recorded in a distributed manner in the synapses of the relevant neuronal assemblies. The recording mechanism takes the form of adjustments to the strength of these many synaptic connections. This synaptic adjustment proceeds by means of a dynamic learning process that must be simulated as part of the model. In the previous study ([2]), the model used the back-propagation algorithm, a method of learning (of the so-called supervised type) commonly employed in neural net simulations [6]. Here we shall invoke Hebbian learning, an unsupervised form of neural net learning dynamics that more realistically models potential brain circuitry. We shall show that neurogenesis favorably informs learning (i.e. memory trace formation) in the more realistic modeling context of Hebbian learning. We still find that the rate of learning of the Greek alphabet vis-a-vis the rate of apoptosis and neurogenesis, on average, forms a U-shaped curve. A moderate amount of churn in the lifecycles of individual neurons enhances the learning ability of the network, while too much reduces the ability of a network to retain and exploit information.

The use of an unsupervised form of learning requires development of an intrinsic representation (an endogenous encoding) of the memory traces. That is, in place of guiding the model's output to take on an extrinsically (and arbitrarily) specified encoding of the information to be recorded (as with the supervised learning protocol of back-propagation), we take as a crtitical aspect of the learning ability, the capacity of the model (i.e. of the hippocampus) to implement an intrinsic and autonomous method for encoding of the information being presented. That a neural system has the functionality to do this is a key and novel feature of the present approach to the study of neurogenesis, and one that we believe accurately models memory establishment in the brain. We are aware that our observations may inform issues of natural selection.

Using an autonomous learning model in this fashion, we can begin to approach questions of why this cycle of cell death and rebirth increases learning plasticity. Our simulation

allows us to characterize and investigate the conditions under which a task such as memory formation succeeds and fails. We uncovered surprising behavior with regards to the ease with which a network learned the two alphabets. Briefly put, when we present a randomly-weighted network with the Roman alphabet, sometimes convergence occurs very swiftly, other times more slowly, and yet others not at all. We discovered that quick-converging networks have a much more difficult time when presented subsequently with the Greek alphabet than do those that were required to invest more time in learning the Roman.

A computational model such as ours allows us to investigate why this might be so. We demonstrate patterns based on the neuron participation rate and the level of saturated neurons, and suggest an information-theoretic explanation for how apoptosis and neurogenesis might help the hippocampus to escape situations where its plasticity and information capacity are compromised. In other words, cytotoxicity prevents situtations where a neural network settles into a configuration where all decisions are made by relatively few neurons, to the detriment of its overall capacity, flexibility and power.

This paper is organized as follows: Section 2 describes the architecture of our hippocampus model, including a discussion of the Hebbian dynamics used in the learning process. Section 3 explains the specific process and parameters used during the alphabet learning tasks, and presents our results. Section 4 discusses the experimental results and what they demonstrate about the efficacy of the apoptosis and neurogenesis mechanisms. In addition, we demonstrate how our results support an information-theoretic argument as to why these two processes should be so important to the flexibility of the learning process. Finally, we sum up our results and contributions in Section 5.

2 Experimental Setup

2.1 The Neural Net Architecture, I/O Dynamics, Learning Dynamics

2.1.1 The neural circuit

The hippocampus is comprised of three layers. The first layer, the entorhinal cortex, is connected by the perforant path to the second layer, the dentate gyrus. The latter in turn is connected by the mossy fibers to CA3, the third layer. In the model, we shall refer to these as layer k ($k = 1, 2, 3$ respectively). The kth layer shall have a number, N_k of neurons. The perforant path and the mossy fibers are simulated by forward excitatory (synaptic) connections between the model's layers. The model also includes lateral inhibitory connections within layers 2 and 3. (See Figure 1). Each synaptic connection is characterized by an associated weight w_{ij}^{kl}, a real valued parameter, where the indices denote a connection from neuron j in layer l to neuron i in layer k. Of course only the superscript pairs 12 and 23 (for forward connections) and 22 and 33 (for lateral connections) are relevant, and a synaptic weight corresponding to any other indices that may appear is taken to be zero. We suppose that when a neuron is connected to another neuron that it is connected to all of the neurons in the latter's layer. We shall describe such an arrangement as being a fully connected one.

Setting an associated synaptic weight to zero accomodates missing connections and al-

lowed us to experiment with different levels of connectivity. In the simplest arrangement, each node was completely connected – each neuron's output connected to every neuron in its own layer and every neuron in the succeeding layer. Connections could disappear if the learning process drove their associated weights to zero, but every synapse began the process with a nonzero value. In other tests, each weight had a certain chance of being set to zero during the network's initialization, representing a nonexistent connection, and we furthermore prevented these weights from changing during the learning process. We used values of 25%, 50% and 75% respectively for the proportion of these missing connections. Motivated by the process of neurogenesis, we are modeling the case where neurons don't necessarily spring into being fully connected with all of their counterparts.

2.1.2 Input/output dynamics

Layer 1 is the input layer to the neural net, while the output of layer 3 is taken to represent the output of the network as a whole. The neuronal input/output dynamics are defined as follows. Let y_j^l denote the output of neuron $j (j = 1, \ldots, N_l)$ in layer $l (l = 1, 2, 3)$. Then v_i^k, the total weighted input to neuron i in layer $k (k = 2, 3)$ is specified as follows.

$$v_i^k = \sum_{j=1}^{N_l} w_{ij}^{k,k-1} y_j^{k-1} + \sum_{j=1}^{N_k} w_{ij}^{kk} y_j^k \tag{1}$$

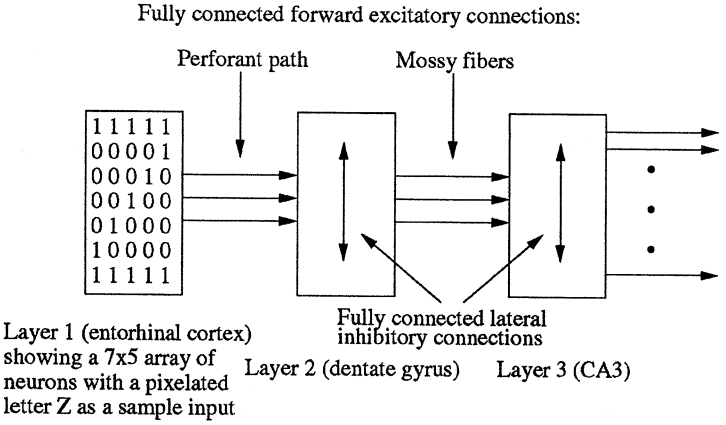

Figure 1: Schematic of the neural network. Arrows indicate a fully connected set of synapses, inter- or intra-layer, as the case may be.

The neurons are taken to be McCulloch-Pitts neurons.[1] This implies that for $k = 2, 3$, the output y_i^k is

[1]McCulloch-Pitts neurons are among the most basic model neurons. Since we are able to demonstrate the relevant apoptosis/neurogenesis effects of interest with these neurons, the use of a more complex model neuron is not called for.

$$y_i^k = \begin{cases} 1, & v_i^k \geq \Theta^k \\ 0, & v_i^k < \Theta^k \end{cases} \tag{2}$$

where Θ^k is the neuronal firing threshold.

2.1.3 Input/output sequencing

Each exogenous input (an alphabetic character) is presented at layer 1. (See Figure 1) The outputs of all neurons in layers 2 and 3 are specified using (1) and (2). These neurons are taken to fire in the following order.

$$y_1^2, y_2^2, \cdots, y_{N_2}^2, y_1^3, y_2^3, \cdots, y_{N_3}^3 \tag{3}$$

So layer 1 fires first and then layer 2 fires, and as indicated in (3), the neurons fire in sequence within each layer as well. The numbering of the neurons within a layer is chosen arbitrarily.[2]

2.1.4 Learning

Synaptic weights are initialized randomly, and they change according to *Hebb's law*, that is, according to correlation between input at a synapse and subsequent firing/not-firing of that synapse's neuron. This law is implemented as follows.

$$\Delta w_{ij}^{kl} = \tau(a_0^{kl} y_i^k y_j^l + a_l^{kl} y_i^k + a_2^{kl} y_j^l) \tag{4}$$

where τ is the learning rate.[3] The coefficients a_j^{kl} appearing in (4) are chosen so that the computed weight change is consistent with the correlations (i.e., with Hebb's law) that arise during the learning stage. A neuron's weights are updated immediately upon that neuron's firing.

2.1.5 Clock cycle

The learning algorithm proceeds with a clock timing indexed with n, say. To indicate that a variable changes with clock cycle, this time index n will be appended to that variable accordingly. A tick (an advance of the clock) is specified in the next section.

2.2 The Learning Tasks

2.2.1 The alphabets and their representation

The neural net is to learn the upper case characters of two different alphabets, the Roman and the Greek. These are represented in pixelated form, an example of which (the letter "Z") can be seen in Figure 1. Illustrations of the complete set of inputs can be found in [2]. Note that these two alphabets have 14 character symbols in common.

[2]The averaging of many randomly initiated runs, characteristic of our approach to the simulation, accomodates our fixing of a single ordering choice for all of the neurons withn a layer; a choice made for reasons of simplicity, causing no loss of generality.

[3]Use of this basic choice of learning rule is validated for reasons analogous to those cited in footnote 1.

2.2.2 The internal coding

As a first task, the neural net (neural circuit) is to learn the characters of the Roman alphabet, and to accomplish this task, the net is repeatedly presented with representations of those characters lexicographically. Referring to Figure 1, we see that layer 1 is modeled as a pixelated retina. The characters are presented in pixelated form on that retina, and so this representation defines the successive inputs $y^1(n), n = 1, 2, \cdots$ as binary vectors. (Note that this assignment of the time index implies that the clock ticks once after the last neuron in layer 3 fires and the latter's weights are updated.) The net creates an *evolving* binary encoding of each character as the latter is inputted. This encoding is defined as the corresponding vector of network outputs $y^3(n), n = 1, 2, \cdots$, and so, a character's encoding is a vertex of the unit binary 2^{N_3}-cube. The alphabet itself is encoded by a collection of such vertices. We take the net to have learned the alphabet (ceasing thereby the learning presentations) if the following two conditions are met.

1. The output encoding the alphabet is a collection of M (where M is the alphabet size) vertices, that is, each character corresponds to a unique vertex.

2. This encoding is repeated exactly without exception during presentation of the entire alphabet an agreed-upon number (say $R > 0$) of times.

2.2.3 The learning task changes

After the net has learned the Roman alphabet, the task is switched to learning the Greek alphabet. The learning of the Greek alphabet proceeds as in the earlier manner for the Roman. Finally, for some of our experiments, the network attempts a third task – that of relearning the Roman alphabet.

2.3 Modeling Apoptosis and Neurogenesis

Finally, we modeled cytotoxicity and neurogenesis by assuming that neurons with highly-saturated weights were more likely to perish from overuse. Whenever we conducted a round of apoptosis, we assigned to each node a probability of death. Each node added the absolute values of all of its weights together, and the probability of cell death increased linearly from zero, according to the amount by which the weight sum exceeded a threshold.[4]

To implement this, let $w_i^{12}, i = 1, \cdots, N_2$ denote the vector of synaptic weights corresponding to forward connections into neuron i in layer 2, and let w_i^{22} be the analogous vector corresponding to lateral connections within layer 2 into that neuron. Then we compute

$$T_i = ||w_i^{12}|| + ||w_i^{22}|| \tag{5}$$

Here $||z||$ denotes the Euclidean norm of a vector z. T_i is a measure of the cyto-toxicity of the corresponding neuron. In addition to the probabilistic cell death mentioned above, we also performed experiments where T_i was treated as a simple threshold – all cells with cytotoxicity above a certain threshold perished and were replaced by new, randomly initialized neurons.

[4]Those neurons with the largest weights will tend to work hardest and age the fastest, and so, would seem to have a biological need to be replaced.

3 The Simulation

3.1 Simulation Protocols and Parameter Values

3.1.1 Layer sizes

We take $N_1 = 35$, corresponding to a 7x5 input retina. N_2 and N_3 are taken to vary over collections of different values. In particular, $N_2 \in \{16, 20, 24, 28, 32\}$ and $N_3 \in \{11, 12, 13, 14\}$.

3.1.2 Synaptic weights, initial values, floor and ceiling

The magnitudes of the initial values of synaptic weights are chosen randomly from a specified interval I, with the forward excitatory weights positive and the lateral inhibitory weights negative. The weights develop according to (4), the learning formula, but they are not allowed to change sign nor are their magnitudes permitted to exceed a ceiling C. Specifically, we take $I = [0, 0.1]$ and $C = 0.125$. If a computed weight change would cause the value of the weight changed to exit the interval $[0, C]$ on the left or right, its actual value is truncated and taken to be the floor or ceiling value 0 or C, as the case may be.

3.1.3 Learning rate adjustment

The displacements calculated by iterative systems, such as the dynamical systems ((1), (2), and (4)) in our simulation, change as the learning progresses, typically trending smaller as convergence is approached. It is useful to vary the learning rate τ, decreasing and increasing it to stimulate the weight displacements to trend smaller or larger more responsively.[5] Among the many ways to install such a feature, the following autonomous choice was taken. Specifically the learning rate τ is varied according to the following rule.

$$\tau_{n+1} = \tau_n \frac{||y^3(n) - y^3(n-1)||_H + 1}{||y^3(n-1) - y^3(n-2)||_H + 1} \tag{6}$$

Here $||y^3||_H$ denotes the Hamming norm of $y^3 = (y_1^3, \cdots, y_{N_3}^3)$, the binary valued vector of layer 3 outputs (that is, the net's output vector).

3.1.4 Hebb rule parameters

For the forward connections from layer l to layer k (i.e., for $(k, l) = (2, 1)$ and $(3, 2)$), we take $a_0^{kl} = 1.5$, $a_1^{kl} = -0.5$, and $a_2^{kl} = -0.5$. For the lateral connections (i.e., for $(k, l) = (2, 2)$ and $(3, 3)$), we take $a_0^{kl} = -1.5$, $a_1^{kl} = 0.5$, and $a_2^{kl} = 0.5$. These choices are seen to accomodate the correlation requirements of Hebb's rule.

[5] Global convergence of iterative dynamical systems is accelerated when the displacements they compute, usually trending larger-to-smaller, are exogenously exaggerated through correlated variations in the learning rate. This commonly used algorithmic feature is sometimes referred to as over/under relaxation.

3.1.5 Learning epoch, repeat parameter, and remission factor

The process of displaying the characters of an entire alphabet in lexicographical order on the input retina (each character display followed by the specified neural firings and weight updates associated with that display) is called a learning epoch.[6] During the set of experiments described in Section 3.2.2, the number of such epochs allowed in a training run was limited arbitrarily to 400. For the tasks described in Section 3.2.3, we removed from consideration all trials that failed to converge within 200 epochs.

The value of the repeat number (for encodings) is set arbitrarily to $R = 3$. We don't expect learning always to be perfect (complete) in a reasonable number of epochs, and so we also introduce a learning remission factor denoted by f. That is, we specify a fraction f of the alphabet that, if learned, is considered adequate. In one experiment, values for f are chosen from $\{0.8, 0.85, 0.9\}$, while in the others, only $f = 0.9$ is used. Allowing the network to misclassify more letters improved the number and speed of convergences, of course, but not enough to be worth sacrificing accuracy. Besides, when investigating differences in convergence timing, reducing the classification threshold would reduce the disparities between high- and low-performing networks, blunting our perception of the difference.

3.1.6 Threshold parameters

The specific choices of thresholds are $\Theta^2 = \Theta^3 = 0.1$.

3.2 Simulation Results

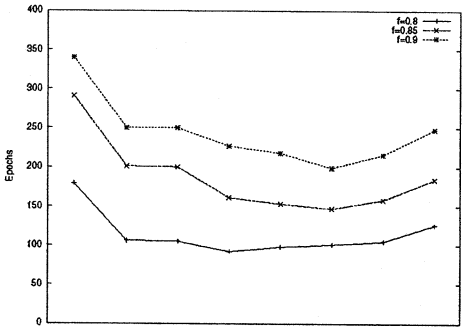

Figure 2: Number of epochs required to reach convergence

3.2.1 Partial connectivity

An attempt to model partially-connected networks failed to produce any useful results. With only 25% connectivity, we never managed to produce a network that converged within the

[6]Since there are $M = 24/26$ characters in the Greek/Roman alphabets, a learning epoch takes 24/26 clock ticks.

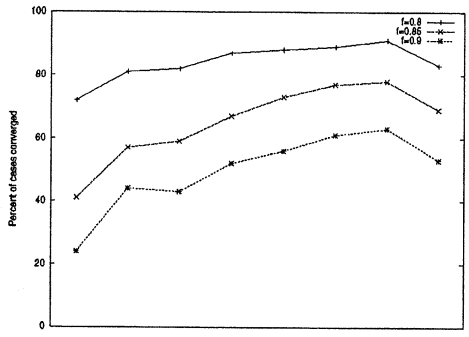

Figure 3: Fraction of runs completing learning

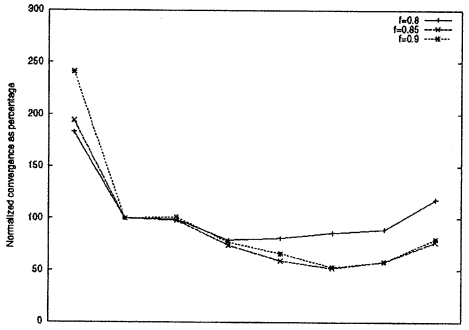

Figure 4: Normalized convergence fraction

time limit. Performance improved at the 50% and 75% connectivity levels, but the only noticeable difference between the results from a fully connected network and those from the partially connected ones was that the latter required more time to reach the same results.

3.2.2 Learning Greek with cytotoxicity

In this experiment, a neural network is exposed to the Roman alphabet until it converges on a unique output encoding for each letter. From this configuration, the network then embarks on the task of learning the Greek alphabet, with varying levels of apoptosis and neurogenesis occurring.

A particular simulation case corresponds to a triplet (f, N_2, N_3). ($60 = 3\text{x}5\text{x}4$ cases in all.) Each case was run 20 times with newly chosen random starting weights each time, 1200 runs in all. Any particular one of these runs is indexed by the symbol[7] $(f, N_2, N_3)(r), r = 1, \ldots, 20$. The results are presented in 4 plots. These are the 3 plots corresponding to $(f, \bar{N}_2, \bar{N}_3)(\bar{r})$, where the upper bars represent averaging over all values of the barred symbol, and one plot

[7]Other parameters (such as a and Θ) associated with a run are not displayed as an index, since those parameters are fixed in value.

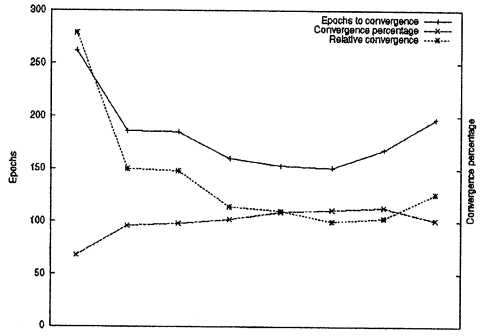

Figure 5: Average over all values of f

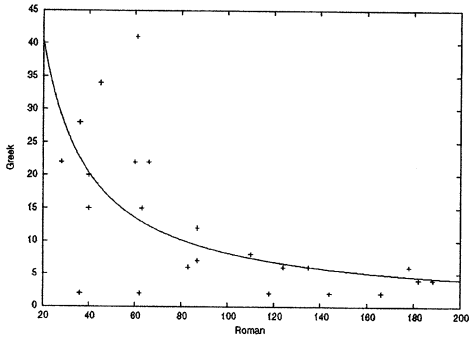

Figure 6: Convergence performance with 16 DG nodes

corresponding to $(\bar{f}, \bar{N}_2, \bar{N}_3)(\bar{r})$. The first three plots, therefore, represent an average over 400 runs each, while the last represents an average over 1200 runs. The plots, each containing three curves, are given in Figures 2 - 5. For the first three plots, the three curves correspond respectively to the three choices of f, namely $\{0.8, 0.85, 0.9\}$.

The abscissas of the plots represent the different learning tasks. The first data point on each curve represents the initial task of learning the Roman alphabet. Each succeeding point denotes the task of learning the Greek alphabet with g neurons replaced ($g = \{0, 1, 2, 4, 5, 12, 16\}$)[8] in layer 2.

Figure 2 is a plot of the convergence time in epochs needed for the learning of an alphabet to occur. In this experiment, if learning in a run does not occur before the limit of 400 epochs is reached, then 400 is taken as a default value of the number of learning epochs required for that run.

Figure 3 is a plot of the fraction of runs (out of 20) that complete the learning before reaching the 400-epoch limit. (Recall that completion of learning means the invariance of

[8]Replaced, meaning as customary, the apoptosis of g neurons followed by their replacement through neurogenesis, moreover with randomly-selected synaptic weights.

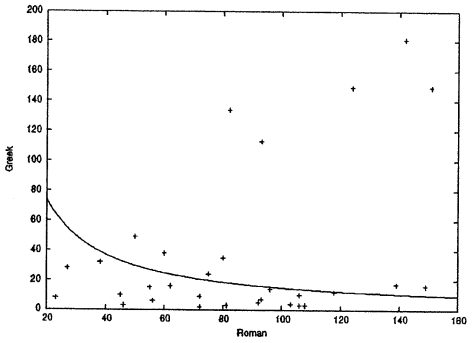

Figure 7: Convergence performance with 24 DG nodes

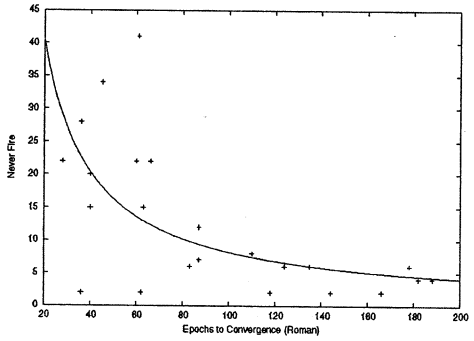

Figure 8: Percentage of DG neuron nonparticipation with 16 DG nodes

the intrinsic encoding achieves the repetition requirement (R=3) subject to the remission factor f.

Figure 4 is a plot of the normalized fraction of cases that converged. In particular, we first compute the ratio of the corresponding curves in Figure 2 by those in Figure 3. This curve is in turn normalized by its own value at the second point (where Greek is learned without neurogenesis).

Figure 5 is a plot corresponding to $(\bar{f}, \bar{N}_2, \bar{N}_3)(\bar{r})$. Namely it is a plot of the average of the three curves in each of Figures 2 to 4. These averages are therfore over 1200 runs.

3.2.3 Patterns of learning and network activity

Having investigated the alphabet-learning behavior over a wide variety of network configurations and initial parameters, we turned to investigate the learning process and parameters in greater detail. We sought patterns between the rates of learning of the two alphabets, the activity level of the various neurons involved, and the distribution of neuronal weights within individual network neurons. In this experiment, we used only two network configurations, because we were interested in clarifying the relationship between the process of learning the

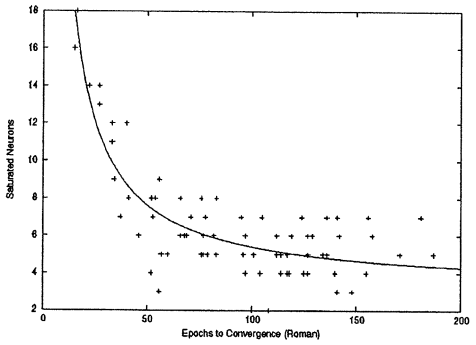

Figure 9: Number of saturated DG neurons after Roman alphabet learning (out of 24 total DG nodes)

two alphabets. Thus each simulation again corresponds to a triplet (f, N_2, N_3), but these take on only the values $(0.9, 16, 13)$ and $(0.9, 24, 13)$. Also, we conduct each trial with only 200, as opposed to 400, epochs.

Figure 6 plots the number of epochs it took for a network to learn the Greek alphabet against the number of iterations it previously spent learning the Roman alphabet. The continuous plot (in this and all subsequent figures) is a least-squares best-fit of the function $f(x) = ax^b + c$. In nearly every case, the Greek was easier for a network to learn after having been exposed to the Roman. However, the key result is that a network that happened to converge very quickly with the Roman alphabet almost always took a comparatively long time to converge on the Greek, while networks that had to work for a long time before finally learning Roman letters took to Greek very quickly.

Figure 7 shows the same kind of data, but for a network with a larger dentate gyrus (24 as opposed to 16). Here, the time trade-off is much less pronounced, and several outliers disturb the picture. Disregarding those, the trend still exists.

Figure 8 shows the fraction of neurons within the dentate gyrus which *never* fire after converging on the Roman alphabet. In other words, these neurons fail to participate in letter identification in any way. Not one of the 26 input patterns elicits any activity from them – they may as well not be there, as far as the letter-recognition task is concerned. Networks that converge quickly have a far greater incidence of these non-participatory neurons.

Figure 9 shows the number of neurons that are saturated after converging on the Roman alphabet. A saturated neuron is one whose weights exceed the apoptosis threshold, and are thus candidates for potential death. Once again, the same curve type emerges. Quick convergence leads to a large number of stressed, saturated neurons.

3.2.4 Relearning and persistence of memory

The final set of experiments involved allowing the networks to revisit the Roman alphabet after they had converged on encodings for the Greek letters. We were interested in the persistence of memory, and whether apoptosis would adversely affect recall of old memories, just as it enhances learning new data. This does not seem to be the case. There were

no statistically significant differences in the relearning rate, whether apoptosis occurred or didn't, and whether Roman or Greek learning went quickly or slowly.

4 Analysis

The ability to learn the Greek alphabet after the Roman has been learned is favorably informed by apoptosis and neurogenesis. After having learned the Roman alphabet, learning the Greek alphabet usually takes 30-40% fewer learning epochs. The Roman learning places the network in a favorable posture for the subsequent learning of the Greek. By favorable posture, we mean that the synaptic weights developed by the learning of the Roman characters already reflect information about the 14 upper-case characters that the alphabets have in common.

Neurogenesis increases the efficaciousness of the Greek learning. The curves in Figure 2 descend with each increase of the number g of new neurons made available by the neurogenesis to the Greek learning, up to a point. When too many neurons are replaced the improvement levels off and then begins to reverse. We interpret this by noting that while the new neurons are aiding the learning of the Greek by replacing older neurons which have become saturated,[9] the apoptosis of the old neurons trained on the Roman alphabet causes a progressive loss of that Roman alphabet information (the commonality of the alphabets) that was responsible for the initial learning-performance gain. The U-shape of the learning curve comes from a tradeoff between a learning and a forgetting effect.

The experiments described in section 3.2.3 throw more light on this tradeoff, and exactly what is happening within the network structure. Rapidly converging networks start out with a few neurons that, by chance, have weights that help a great deal to differentiate among letters of the Roman alphabet. These neurons tend to dominate the network, quickly suppressing competing neurons to the point of quiescence, as shown in Figure 8.

Furthermore, the weights of the neurons that do participate become saturated (Figure 9). Thus, when the alphabet is learned quickly, the network consists of many neurons that are effectively ignored and others which fire often and indiscriminately.

From an information-theoretic perspective, the poor capacity for learning additional information exhibited by these inflexible, saturated, dominating neurons comes as no real surprise. A network in which all neurons participate, and where each synapse is weighted differently, can convey a large amount of information. A neuron with a well-distributed set of weights has a huge number of possible internal states, and therefore potentially a large amount of entropy (in the information-theoretic sense). The better-distributed the weights are, the closer the entropy will come to the theoretical maximum $\log(2K+1)$, where K is the number of internal states available to the neuron.

Entropy, of course, translates directly to information content. In the case of the networks that resist learning the Greek alphabet after settling on the Roman, this content is quite low. Many neurons don't participate at all, so their potential to encode information is completely ignored. Furthermore, the neurons that do participate have a very limited internal state

[9] As previously mentioned, a neuron is taken to be saturated when the Euclidean norm of the vector of input weights reaches a certain threshold. This happens when the weights have been driven by the dynamics to be near the ceiling C.

space – a huge fraction of all possible input encodings lead to exactly the same output, since many of their weights are saturated at the maximum. This being the case, the fact that they fire comes as no surprise. Low surprise means little information.

Thus networks that slowly converge on the Roman alphabet carefully refine their weights, preserving a wide diversity of possible firing patterns and exploiting the capacity of many more neurons than in the case of rapidly-converging networks. Such networks are simply more effective – they are able to encode and transmit more information than their fast-learning counterparts that overwork some of their constituent neurons and underemploy others. Our evidence suggests that this is why they can learn new information much more quickly.

By preferentially replacing ill-behaved neurons that simultaneously suppress participation by others and carry little information themselves, apoptosis and neurogenesis help to maintain a network information capacity that is closer to ideal. This informs the puzzling result that apoptosis does not have a negative impact on relearning old information. The loss of these dominating but information-poor neurons is more than made up for by the increased responsiveness and activity of the other neurons in the network.

5 Contributions

We employed an autonomous (i.e., unsupervised) form of learning to model and simulate the recording of information in the hippocampus. This required that the model be capable of developing memory traces that are intrinsic representations (endogenous encodings) of the information to be learned. Demonstrating that a neural system has the functionality to do this is a key and novel feature of the present approach.

We have uncovered striking patterns in the learning behavior of unsupervised neural networks, suggesting a relationship between the time and effort it takes to learn something and the flexibility and adaptibility of that knowledge once learned. To express it in human terms, it is as if a person who develops an unthinking "knack" for a particular process often has a harder time adapting and applying the process to new circumstances than someone who acquired the skill through dint of careful study. Whether human experience bears this out is debatable.

These findings provide more support and justification for the idea that apoptosis and neurogenesis in the hippocampus promote increased and sustained learning ability. We have accounted for differences in learning ability by demonstrating the deleterious effects of both saturated and silent neurons, effects that can be mitigated through the mechanism of cell death and replacement.

Finally, we have introduced some ideas about the theoretical information capacity of neural networks, as illustrated by the alphabet learning experiments. We expect these findings to inform the development of procedures and rules of thumb for extracting good performance out of neural network frameworks in the future.

References

[1] Aakerlund, L., Hemmingsen, R., (1998). Neural networks as models of psychopathology. *Biol Psychiatry*, vol. 43, pp. 471-482.

[2] Chambers, R. A., Potenz, M. N., Hoffman, R. E., Miranker, W. I., (2004). Simulated apoptosis/neurogenesis regulates learning and memory capabilities of adaptive neural networks. *Neuropsychopharmacology*, vol. 29, pp. 747-758.

[3] Erikson, P. S., Perfilieva, E., Bjork-Eriksson, T., Alborn, A. M., Nordburg, C., Peterson, D. A., Gage, F. H., (1998). Neurogenesis in the adult human hippocampus. *Nature Medicine*, vol. 4, pp. 1313-1317.

[4] Gazzaniga, M. S., Ivry, R. B., Mangun, G. R., (2002). *Cognitive Neuroscience.* W. W. Norton and Co.

[5] Gould, E., Beylin A., Tanapat, P., Reeves, A., Shors, T., (1999). Learning enhances adult neurogenesis in the hippocampal formation. *Nat Neurosci*, vol. 2, pp. 260-265.

[6] Haykin, S., (1999). *Neural Networks: A Comprehensive Foundation.* Prentice Hall.

[7] Kornack, D. R., (2001). The generation, migration, and differentiation of olfactory neurons in the adult primate brain. *PNAS*, pp. 4751-4757.

[8] Makakis, E. A., Gage, F. H., (1999). Adult-generated neuorons in the dentate gyrus send axonal projections to field CA3 and are surrounded by synaptic vesicles. *J Comp Neuro*, vol. 406, pp. 449-460.

[9] Wirth, S., Yanike, M., Frank, L. M., Smith, A. C., Brown, E. N., & Suzuki, W. A., (2003). Single neurons in the monkey hippocampus and learning of new associations. *Science*, v. 300, pp. 1578-1581.