Studies on the Memory Capacity and Robustness of Chaotic Dynamic Neural Networks

Igor Beliaev
Computer Science Division,
The University of Memphis
Memphis, TN USA
ibeliaev@memphis.edu

Robert Kozma
Computer Science Division,
The University of Memphis
Memphis, TN USA
rkozma@memphis.edu

Abstract - A dynamical neural model that is strongly biologically motivated is applied to learning and retrieving binary patterns. This neural network, known as Freeman's K-sets, is trained with Hebbian rule and habituation to memorize the input patterns by associating them with an attractors formed in the state space. After the patterns are memorized noisy input is given to the network to recover the original. We compare the results of this recall for a different number of memories and compare them with performance of the Hopfield model. We show capacity of the dynamical system exceeds that of the Hopfield network and the noisy recall degrades at a slower pace as the number of the patterns is growing. Experimental results indicate that the critical load paramter, which gives approximation of the network capacity, is higher in K-model than in the Hopfield network. Significant advantage of K-model is achieved at a larger training set size, when compared to Hopfield model.

I INTRODUCTION

Ability of the human brain to form long-term memories from a relatively few examples and enormous capacity of the memory fascinated many researchers and encouraged to seek for artificial neural models that would exhibit similar properties. Memory models find their use in many artificial intelligence systems applications and can provide an insight and help study the mechanisms of brain functioning and cognition in general. Many of the proposed systems suffer from a very limited capacity and ability to discriminate incomplete or noisy inputs is also often poor. One of the associative memory models, the famous Hopfield network introduced by Hopfield in 1982 [1, 2], have been carefully mathematically analyzed bounds on the storage capacity have been found. It is known that the load parameter $\alpha = M/N$; where M is number of fundamental memories, N is the number of neurons in the network or the size of the network, has a critical value $\alpha_c = 0.138$ [3]. The quality of memory recall in the Hopfield network significantly deteriorates for the values of load parameter grater than this critical value. Thus, the storage capacity with errors on recall of the Hopfield network is roughly 0.14N. McEliece et al. [4] found the upper limit on the training inputs is $N / 2 \log_2 N$ for exactly recoverable patterns. Storage capacity of neural networks models in more general way have been investigated by Gardner and Derrida in [5, 6]. They calculated the number of random N-bit patterns that an optimal neural network can store allowing a given fraction of bit errors in the models where each bit is stabilized by a local field at least equal to some parameter. More recently a lot of attention has been given to dynamical models with non-convergent dynamics and chaotic systems. In a number of works by Molter et al [7, 8] have developed a chaotic dynamical neural models together with supervised and unsupervised training algorithms, and reported on the storing information in such networks. The input patterns are represented in the quantized limit cycle attractors of different period, which the network forms in many iterations of Hebbian training. The authors show that such network has great memory capacity that goes far beyond that of any convergent dynamics neural networks. Another approach to storing information in oscillatory systems is described by Nishikawa et al. in [9]. They study the capacity of oscillatory associative memory network with error-free retrieval and show that networks of coupled periodic oscillators can form stable states and reach capacity comparable with Hopfield model. These associative memories are based on temporal coding of information, where the synaptic phase delays between units are adjusted.

In this work we report on initial study of the capacity of the novel type of the dynamical neural model, the Freeman K-models. These systems have been developed by Freeman [10, 11, 12] in the attempt to construct a biologically plausible computational model of olfactory system. The observation on the olfactory bulb in rabbits during cognitive tasks made Freeman think of spatio-temporal nature of the information

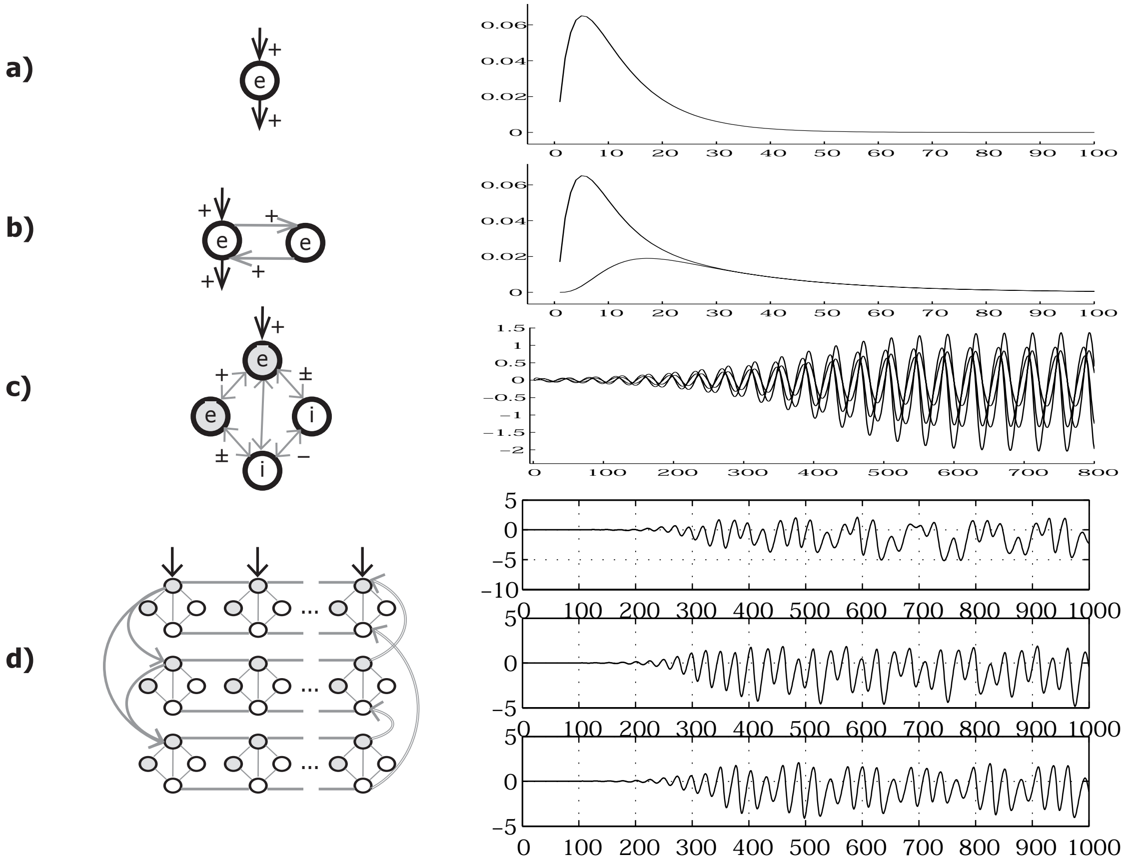


Figure 1. Hierarchy of K-sets with network structure and small perturbation response. a) K_0 simplest k-unit, models population of 10^4 brain cells, b) K_I – few basic K_0 units laterally connected provide prolonged response to small input perturbation, c) K_{II} – two excitatory and two inhibitory units with negative feedback connections result in oscillatory behavior given proper choice of weights [w_{ee}, w_{ei}, w_{ie}, w_{ii}], d) K_{III} – three double layers consisting of fully laterally connected K_{II} sets that are connected with no-delay feed-forward connections and delayed feed-back connections. This model can produce chaotic oscillation as a result of initial impulse.

processing in the brains. The brain systems and sensory cortices of animals always maintain very unstable basal state of chaotic oscillatory pattern in the EEG signal that reflects the magnitude of the neurons pulse density. The computational paradigm of the chaotic itinerary by Kaneko and Tsuda [13] states that the trajectory of highly complex attractor space jumps between quasi-stable attractor basins. In real brains the jumps happen in the time frames at a rate of 5-10 per second. The encoding of information for each frame is in the 2D spatial pattern of the amplitude modulation (AM) of an aperiodic carrier wave. These AM patterns are selected by the sensory stimuli and are shaped by synaptic connection weights. To form the proper attractors in the K-model's state space we use two simple training rules, one the Hebb rule and another the habituation, which is a diminished response to sensory stimuli that is not reinforced. We test the K-model's ability to learn and distinguish between binary patterns of English letters. The results of our experiments with K-models are compared with the capacity of the Hopfield network trained on the same data set.

II K-MODELS AS ASSOTIATIVE MEMORY

The architecture of artificial neural networks is inspired by the nervous system. It captures the information from data by learning and storing information in its weights. This computational model is especially good, considering generalization and error tolerance, compared to other symbolic computational models. In a series of works Freeman has developed biologically realistic computational model of the olfactory system [10, 11] – the K-models. The most prominent difference from other generally known NNs is that the activity of the K model is oscillatory, which typically lies in the chaotic regime.

Figure 1 introduces the hierarchy of K-set. The basic K-unit, called K_0 set, models behaviour of a population of about 10^4 biological neurons. As the diagram on the Figure 1a) shows the K_0 unit accepts the combination of the external input I(t) and internal connections from other units in the network F(t). Its own dynamics is governed by the second order ordinary differential equation as in the equation (1), and the output activation x(t) is then processed through the sigmoid function as in the equation (2).

For a general topology K-model we can write the system of second order ordinary differential equations in the following form:

$$ab \cdot \frac{d^2}{d^2 t} x_i(t) + (a+b) \cdot \frac{d}{dt} x_i(t) + x_i(t) = F_i(t) + I_i(t);$$
$$F_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^{N} w_{ij} \cdot Q\left(x_j(t - \theta_{ij})\right); \qquad (1)$$

where a and b are real-time constants, a = 0.22, b = 0.72, that are determined experimentally from brain tissue to best model the impulse response of the neural population on mesoscopic level [10]; $x_i(t)$ – is the ith unit activation at time t, $I_i(t)$ – is the external input the ith unit receives at time t, $F_i(t)$ – activation coming from all the internal connections, weighted by the weight w_{ij}; Θ_{ij} - is the time delay parameter for the link between units i and j; indices i and j run from 1 to N – the total number of the units in the K-network; no auto-feedback connections are allowed in the system, more details on formal description of k-models are found in [14, 15]. As it is shown on the Figure 1d., feedforward connections all have zero delay parameters unlike the feed-back connections drawn as the dotted arrows on the diagram. The asymmetric sigmoid transformation function is as in the equation (2):

$$Q(x) = q \cdot \left\{ 1 - \exp\left(-\frac{(e^x - 1)}{q} \right) \right\} \qquad (2)$$

where q is a parameter, which specifies the slope and maximal asymptote of the curve, and in all our experiments we use q = 5. This sigmoid function is also modeled from experiment on biological neural activation [12]. Given small initial perturbation from external input, this K_0 responds as shown on the plot in Figure 1.a.

Coupling two or more K_0 sets with excitatory connections, we create a K_I set whose structure and behaviour is depicted on Figure 1.b. The next step in the hierarchy is the K_{II} model. K_{II} is a double layer of excitatory and inhibitory units. In the simplest architecture there are 4 nodes: two excitatory, denoted e, and two inhibitory, denoted i; see Figure 1.c.

Given small external impulse at the top layer this model may produce sustained periodic oscillations. Frequency and magnitude of these are determined by the interconnection weights going between units. In order to achieve a certain level of stability on K_{II}, we conduct several experiments to search for the weight parameters, namely, w_{ee}, w_{ei}, w_{ie}, w_{ii}, so that K_{II} can sustain oscillatory activation in impulse-response tests. The selection of weight parameters is also based on the stability analysis by Ilin et al. in [16].

The K_{III} model is designed to be a dynamic computational model that simulates the sensory cortex. It can perform pattern recognition and classification. Based on the structure of the cortex, K_{III} consists of three layers connected by feedforward/feedback connections. Each layer has multiple K_{II} sets, connected by lateral weights between corresponding e_1 and i_1 nodes. Although our model uses full connectivity, different topologies are possible. Feed-forward connections join excitatory units from the first to second, first to third and second to third layers, denoted by double arrows on the figure 1.d. Feedback connections are delayed and drawn as dotted double arrows on the figure.

Each of the K_{II} layers alone is capable of producing periodic, or limit cycle, oscillations that do not allow for formation of complex attractor landscape necessary for storing large quantities of information. Only in combination of three layers, each oscillating at incommensurate frequency and in a different attractor region, there appears a complex aperiodic oscillation in the k-system as a whole. The first K_{II} layer, which accepts the external input signals, generates spatially coherent carrier wave that is propagated by the second K_{II} layer. The second layer plays a role of chaotic controller allowing the whole system to stay in balance. The third layer, by analogy with underlying biological model [14], provides the output and is adaptable through the Hebbian learning. The complex aperiodic dynamics of the well-balanced K_{III} model forms an attractor landscape that is shaped by the learning. Different attractor regions in this landscape correspond to different input stimuli received by the network, which it can distinguish by switching between the attractors. No K_I or K_{II} systems are capable of producing that type of behaviour.

Various criteria need to be satisfied in order to achieve a healthy K_{III} dynamical state, which exhibits aperiodic oscillations. Details of the parameter selection procedure are given in [14]. The present work uses the parameters determined in those studies. They guarantee a balanced chaotic regime for K_{III}, and sensitivity to external input that is projected on the top layer, which resides within the range of [-1, 1].

In order to use K_{III} as an associative memory, we apply Hebbian learning to the lateral weights between the excitatory nodes in the third layer. This treatment is biologically motivated, as the 3rd layer in our case models the sensory cortex. Next few paragraph give the details of the learning procedure.

A. Hebbian learning and habituation in K_{III} model

After the set up of K_{III} to a balanced state, its aperiodic oscillations are sustained if the inputs are constrained within a working range [-1, 1]. We use a combination of two learing techiniques. The first, the Hebbian updated. Since the system is always in a non-convergent dynamic state, there is no single value where the system activity converges. Instead, we use the quantity σ_i, which is computed according to the formula (3) from the activation over a given time window, and which is analogous to standard deviation.

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}\sigma_i;$$
$$\sigma_i = \left(\frac{1}{\tau_1}\sum_{t=1}^{\tau_1}\left(x_i(t) - \frac{1}{\tau_1}\sum_{t=1}^{\tau_1}x_i(t)\right)^2\right)^{1/2}. \qquad (3)$$

Let t be a time parameter that is measured according to discretization, which is used to solve numerically equations (1) by the 4th order Runge-Kutta method. This computation is performed for every node while computing next state of the K_{III} model. Let an excitatory node i in the third layer have activation $x_i(t)$. Take an input sample, binary vector $x \in \{\pm 1\}^N$, and present it to the top excitatory unit of the 1st layer for a $t_1 = 200$ time steps. This is one-to-one projection of the vector x to the sensory layer of the network. At the end of this period, we compute the quantities σ_i to measure the AM patterns.

Because of the oscillatory nature of the activation in the K_{III} model the standard Hebbian rule has to be modified. Details of the rationale are in [10], and here we introduce the formula:

$$\Delta w_{ij} = \alpha * (\sigma_i - \sigma) \cdot (\sigma_j - \sigma); \qquad (4)$$

where α is the learning rate; Δw_{ij} is the weight change between nodes i and j. In the same terms the habituation learning rule is as shown by equation (5):

$$\Delta w_{ij} = -\eta \cdot |\sigma_i - \sigma|; \qquad (5)$$

where η is the habituation learning rate. When all these changes are computed, the network is updated with new weights. Each update happens after the active phase, when the input is presented. There is a relaxation period of 100 time steps between any two inputs in order to let the network reach its basal oscillatory state. No changes happen during relaxation, and the network goes back to its balanced oscillations. Without this period affect of one training example would intervene with another, which would not allow the formation of distinct attractor regions for each sample and therefore distorting the training.

When the network receives input, the dynamical trajectory in the phase space changes and the network is forced out of the basal state into an attractor region corresponding to the presented input. We capture this attractor by enforcing the connections between the units with Hebbian rule.

The choice of learning rate is always an interesting question for neural network training. In our model we do not attempt to use adaptive learning rate and chose some fixed number. However we make the following argument on the selection of this number. As mentioned earlier, it is very important for a K_{III} model to maintain its balance between the layers' oscillations, otherwise the network would not be able to form distinct attractors in the state space and therefore would be unable to learn and store information. General observation about activity within one layer is that K_{II} units tend to synchronize their behaviour if lateral connectivity is too strong. This needs to be avoided so as to preserve sensitivity of the model to external input. Our layers are fully connected and therefore we have to set lateral connection weights to some values that would not lead to complete synchronization of units. During training these lateral connection weights are increased proportional to the learning rates and such updates may happen as many time as many there are training samples. The Hebbian update strenghtens weights, while the habituation brings the weights down at a slower rate. The idea is to balance the two learining rates so to make the most prominent connection grow while sligtly decaying all other. The relative change between weight before and after training should stay within the bounds in order to maintain the homeostatic balance in the system. Therefore we shall be very carefully to calculate the boundaries for the learning rate. From the experience we can estimate the values of the learning rates given the number of training samples, knowing that the relative change of weight should not exceed 20-30% in order to keep the system in good balance. In previous works [17, 18] we have applied K-models to time series prediction and some intial stury was done on the relationship between the learning rates and size of training set, as well as on other aspects of training in the K_{III} network.

The convergence of this training is not guaranteed and there is no explicit study on this for K_{III} model. We limit the growth/decay of the lateral weights during the training by limiting the number of epochs.

III EXPERIMENTAL RESULTS AND DISCUSSION

We conducted a series of experiments to test and compare ability of the K-models and Hopfield network to retrieve noisy binary patterns. The data set is binary representation of English alphabet as presented on the Figure 2. It consists of 26 letters in 6x6 arrays of -1's and 1's. The choice of bipolar samples is dictated by the Hopfield network general design and it also falls in line with the working range for the K-model, which can fit unipolar representation as well. There is no need to

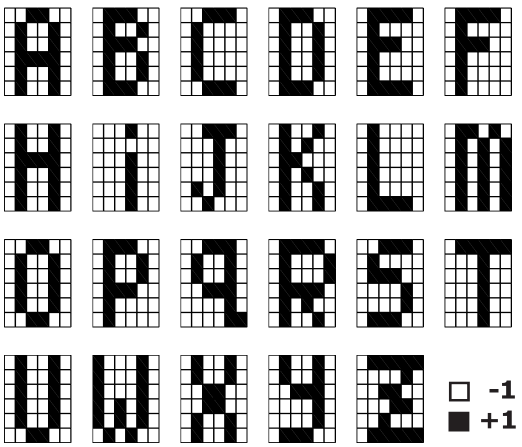


Figure 2. The set of binary English letters for forming fundamental memories.

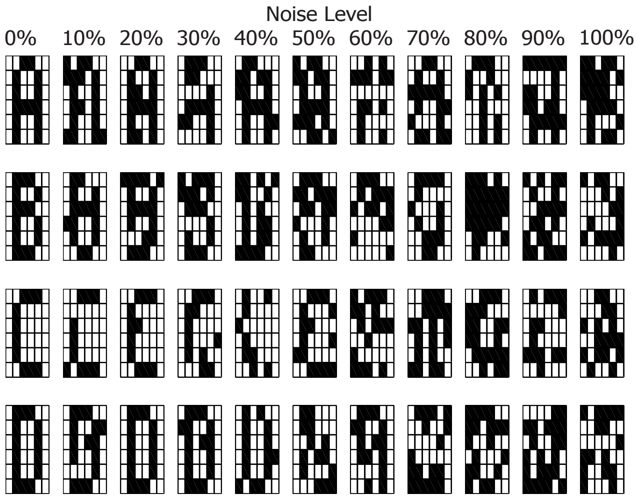


Figure 3. Example of noisy letters for testing the models at different level of noise ξ.

renormalize input patterns and hence unipolar inputs were not considered, both models are tested on the same types of inputs. We compose a set of samples by drawing at random n = 2..15 letters without repeat. Then, in the case of the K_{III} network, we train the network with Hebbian and habituation algorithms as described in the previous section for 3 epochs. In the process of training the lateral weights in the third layer are adjusted. After the training we simulate the network with the same inputs ones again in order to produce and collect activation patterns and associate them with the input samples in the reference table. Reference table thus contains the different attractor snapshots and the corresponding input samples. These represent the internal fundamental memories of the K-model that are not the fixed point attractors, like in the Hopfield network, but a limit cycle chaotic attractors. Strictly speaking, we cannot guarantee that these attractors of the K_{III} network are stable, however work on the analysis of stability of limit cycles in the K_{II} networks by Ilin and Kozma [16] may be extended to the case of K_{III} network. We present the network with noisy patterns, collect the activation pattern from the third layer, and find the closest match, in terms of simple Euclidean distance, in the reference table, from which the corresponding letter is taken for the network output. If this letter matches the target we count it as successful recovery, otherwise we count it as an error. We would like to point out, that despite storing the original patterns in the reference table, we do believe the simulation is valid and the comparison with the Hopfield model, where the patterns are only implicitly stored in the weight, is fair. The reference table for the K-model contains in internal representation of the incoming input stimuli, a "meaning", if one agrees, for the K-model this pattern carries. Therefore in the absence of actuators in our simulation we detect the internal state of the network and associate it with the appropriate experience of input letter patterns. This would make perfect sense when a K-model is embedded in a robotic system and can generate actions based on the internal state directly [19]. After repeating this experiment 10 times for every level of noise ξ = 0% .. 100% with step 10% we collect the statistics on the error. The noise level corresponds to the probability of randomly choosing a pixel in the pattern and setting it to -1 or 1 with probability 1/2. This experiment is repeated for all n. No other scheme for distorting the inputs has been tested.

Similarly we test the Hopfield network. Each experiment is repeated 10 times for all n, and the number of perfect recalls is counted. We simulate the Hopfield network for a fixed number of 50 iterations, that is nearly one and a half times greater than needed to let the network converge on average for the most severely damaged pattern. If the probability of changing any bit in the pattern is p, than on average it takes 36 / p iterations to correct all these corrupted bits. When p is 100% the network should take about 36 iterations. We compute the error is the number of imperfect recalls.

Figure 4 shows the results of these experiments, where each plot gives the percentage of the correct recalls for K_{III} model, Hopfield network, and the random guess of the letter. Each subplot has number of memories on the axis of abscissa and the error in per cent on the axis of ordinates. The ten subplots a-j correspond to the levels of noise in the input samples.

As one can see random guess is very rapidly vanishes to near-zero correct recognition with the growth of the number of letters in the test set. Intuitively it is clear that the probability of picking the right letter correctly is inversely proportional to the number of letters. The theoretical limit on the storage capacity for Hopfield network before recall start to deteriorate

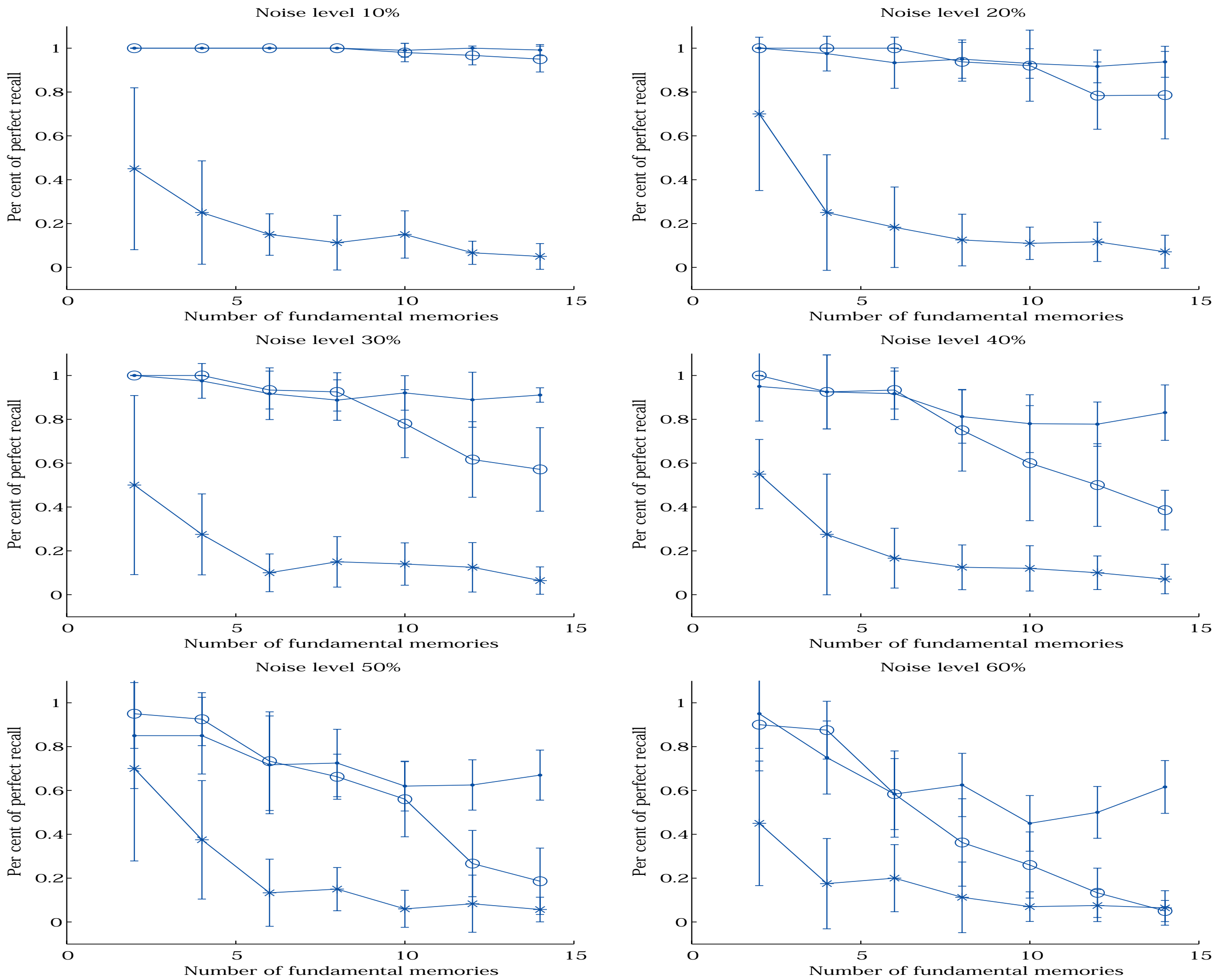
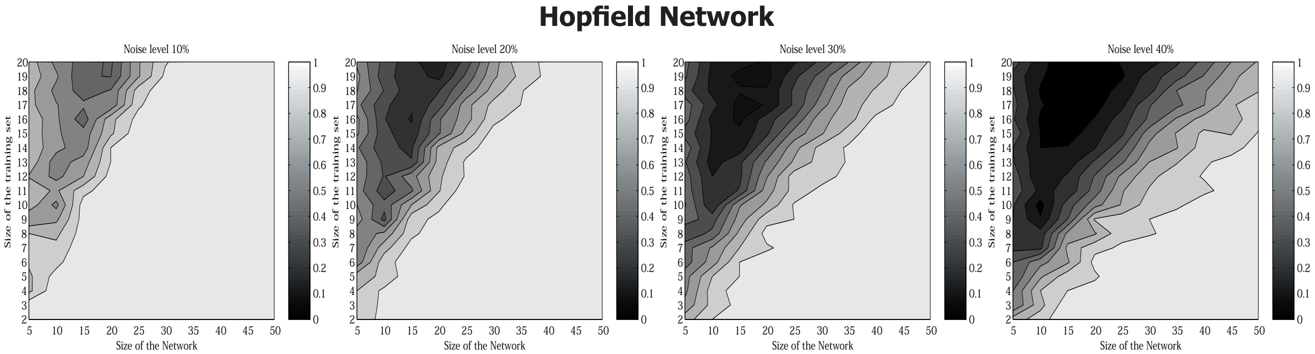


Figure 4. Comparison of the per cent of perfect recall from –○– Hopfield model, –•– KIII model, and –*– random guess. Horizontal axis - number of patterns to memorize, vertical the correct recall in per cent. Each subplot corresponds to a different level of noise running from 10% to 60%. For higher levels of noise, all models are practically indistiguishable from one another. In the case of zero noise both Hopfield and K-model give 100% accuracy.

for noisy inputs is 0.14 of the number of neurons. In our experiments of 6x6 = 36 neurons it corresponds roughly to 5 memories. It is observable on the figures that approximately at that level Hopfield model perfect recall rate decreases drastically. Most vividly this is seen on the Figure 4 and noise level 0.4 and 0.5. On the other hand, the K-models show very similar pattern for few sample set, n = 2..6, but are staying on top of the Hopfield for larger sets. We conclude that the K-models have greater potential in the storage capacity when measured against Hopfield network on the recall with no errors, i.e., the perfect match of the output and the target.

In order to estimate the critical value of alpha_c for the K-models, like it is known for the Hopfield to be 0.138, we conducted another experiment. Both models are tested on the random binary patterns for size of the network from the set {5, 10, 20, 30, 40, 50}, and the size of the training set n = 2..20. In a similar fashion we compute the average per cent of the perfect recall for noisy patterns with ξ = 0.1..0.5 and with 10 repeats for every experiment. Results of this experiment are presented on the Figure 5, where on horizontal axis there is size of the network, on the vertical number of patterns to store, and the recall is colorcoded contour plot. One can clearly see the boundary at which the performance for both networks starts to deteriorate given the noise level. It also clear that the slope of the line that separates the two regions, bright for good results and dark for poor, is noticeably higher in the K-models. In the case of Hopfield our results fall in line with the known estimate of 0.13-0.15 for the slope depending on how much error one allows to consider results satisfactory. It is premature to make a conjecture on the asymptotic behaviour of α_c for K-models from this experiment, but we are convinced,



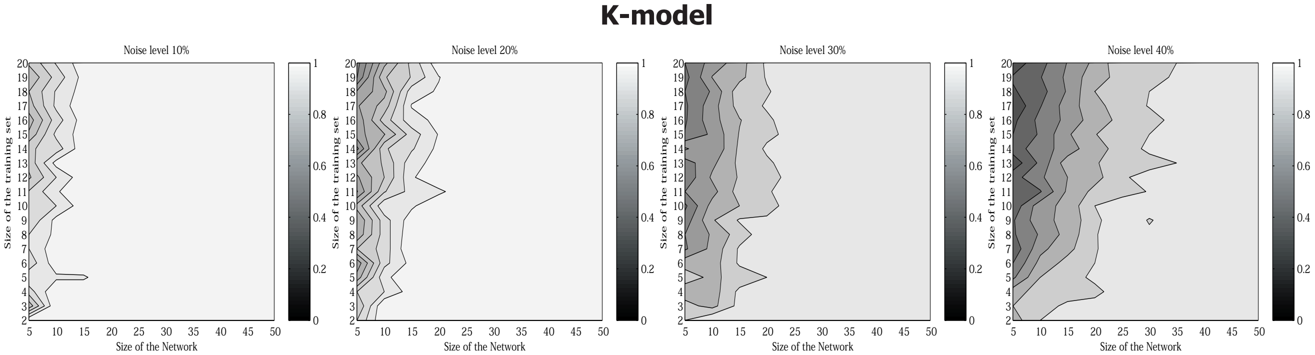


Figure 5. Comparison of the border line of good and poor performance in the perfect recall of random binary patterns by Hopfield network and KIII model. The slope of line separating bright region from dark gives estimate on the critical value of load parameter α_c, which is higher for K-models.

the performance of K-model scales at least linearly with network size and the critical value for load parameter is higher than that of Hopfield network.

The number of neurons, which is the principal factor for network capacity, in the K-model is greater than that in Hopfield network. K-model has three layers each with four times size of the input sample, which is 3x4x36 = 423 in the letters experiment, versus only 36 in the Hopfield network. The number of connections is also greater and so is the number of weights. K model can be viewed as an extended Hopfield network with additional layers, which generate nonconvergent oscillatory behavior. The encoding and recall happens based on the activations in a single layer (layer #3 in this case), which is comparable with the layer of nodes in the Hopfield network. All the additional nodes and connections act simply as surrogates and they are not modified during the learning process. Thus, the number of plastic weights, which we change during training is exactly the same as in the Hopfield network, i.e., order of N·(N-1). It is possible to reduce the overall number of units in a K-model without changing its dynamical properties, but we have not attempted this.

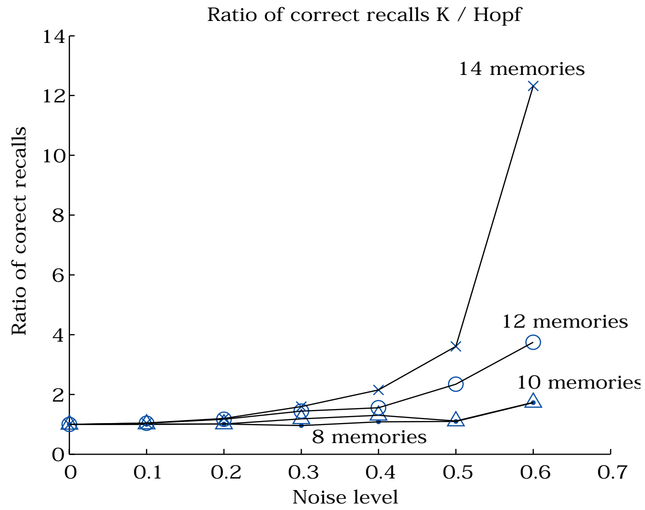


Figure 6. Ratio of number of correct recalls made by K-model to that of Hopfield for larger sizes of training set. K model has done drastically better.

Therefore, we attribute the greater capacity of K-models to the ability of the non-convergent dynamical systems to form attractor landscapes that can fit many more chaotic attractors than fixed point attractors like in convergent models as Hopfield, and not to the fact that K-model has more neurons and weights, in which it can store information. Illustration on the advantage the use of non-convergent dynamics gives to memory capacity is presented on the Figure 6, where we plot the ratio of the number of correct recalls from K-model to the number of correct recalls from Hopfiled model. For larger size sets, i.e., n = 8, 10, 12, 14, the improvement is rather significant.

IV CONCLUSION

In this work we have shown how a dynamical chaotic neural network can be trained to store binary patterns of letters of some alphabet. We used simple Hebbian rule and habituation to train the network. The results suggest that K-models have greater potential than the Hopfield model in the storage capacity. The tests we presented here clearly show superiority of K-model for retrieval of binary patterns for larger set sizes. The critical value of load parameter is also found to be greater for the K-model in comparison with the Hopfield networks. The achived improvement in the capacity is order of magnitude larger for the K-model than for Hopfield for large size of training set.

We have not fully researched the optimality of the learning rates for our training mechanisms of Hebbian rule and the habituation. This is an on-going effort and we plan to extend the research in this direction.

REFERENCES

[1] J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Natl. Acad. Sci. USA , 1982, volume 79, pp. 2554-2558.

[2] J.J. Hopfield, Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons, Proc. Natl. Acad. Sci. USA, 1984, volume 81, pp. 3088-3092.

[3] B. Mueller, J. Reinhardt (1990), Neural Networks: An Introduction, Springer.

[4] R.J. McEliece, E.C. Posner, E.R. Rodemich, and S.S. Venkatesh, "The capacity of the Hopfield associative memory," IEEE Trans. Inform. Theory, vol 33, pp. 461-482, July 1987.

[5] E. Gardner, B. Derrida, "Optimal storage properties of neural network models", Journal of Physics A: Mathematical and General, Volume 21, Issue 1, pp. 271-284 (1988) .

[6] E. Gardner, B. Derrida, Three unfinished works on the optimal storage capacity of networks J. Phys. A 22, 1983-1994 (1989)

[7] C. Molter and H. Bersini, "How chaos in small hopfield networks makes sense of the world," Proceedings of the IJCNN conference - Boston, July, 2003.

[8] C. Molter and U. Salihoglu and H. Bersini, "Introduction of an hebbian unsupervised learning algorithm to boost the encoding capacity of hopfield networks," Proceedings of the IJCNN conference - Montreal, July, 2005.

[9] T. Nishikawa, Y.-C. Lai, and F. C. Hoppensteadt, "Capacity of oscillatory associative-memory networks with error-free retrieval," Physical Review Letters 92, 108101(1-4) (2004).

[10] W.J. Freeman (1975), "Mass Action in the Nervous System" Academic Press, N.Y.

[11] W.J. Freeman, "Mesoscopic neurodynamics: From neuron to brain", J. Physiol., 2000, Sep-Dec; 94(5-6):303-22.

[12] H.Z. Chang, W.J. Freeman, B.C. Burke (1998) "Optimization of Olfactory Model in Software to Give 1/f Spectra Reveals Numerical Instabilities in Solutions Governed by Aperiodic (chaotic) Attractors", Neural Networks, 11, 449-466.

[13] K. Kaneko and I. Tsuda, "Chaotic itinerancy," Chaos: Focus Issue on Chaotic Itinerancy, vol. 13, no. 3, pp. 926–936, 2003.

[14] R. Kozma, W.J. Freeman, "Aperiodic Resonance - Methods and Applications of Noisy and Variable Patterns", Int. J. Bifurcation & Chaos, 11(6), 2307-2322, 2001

[15] J.C. Principe, V.G. Tavares, J.G. Harris, W.J. Freeman (2001), Design and implementation of a biologically realistic olfactory cortex in analog VLSI, Proceedings IEEE (2001) 89:1030-1051

[16] R. Ilin, R. Kozma, W.J. Freeman (2004, "Studies on the Conditions of Limit Cycle Oscillations in the KII Models of Neural Populations," IEEE/INNS 2004 Int. Joint Conference on Neural Networks IJCNN'04, July 25-29, 2004, Budapest, Hungary.

[17] R. Kozma, I. Beliaev, "Time Series Prediction Using Chaotic Neural Networks: Case Study of IJCNN CATS Benchmark Test", Proc. 2004 IEEE IJCNN, Budapest, July 2004.

[18] I. Beliaev, R. Kozma, "Time Series Prediction Using Chaotic Neural Networks on the CATS Benchmark," to appear in Neurocomputing, 2006.

[19] S. Muthu, R. Kozma, W.J. Freeman (2004), "Applying KIV Dynamic Neural Network Model for Real Time Navigation by Mobile Robot Aibo," IEEE/INNS 2004 Int. Joint Conference on Neural Networks IJCNN'04, July 25-29, 2004, Budapest, Hungary.