SCRAM: Statistically Converging Recurrent Associative Memory

Sylvain Chartier
Department of psychology, UQO, Gatineau, Québec, J8X 3X7, Canada
chartier.sylvain@courrier.uqam.ca

Sébastien Hélie
Department of computer science, UQAM, Montreal, Quebec, H3C 3P8, Canada
helie.sebastien@courrier.uqam.ca

Mounir Boukadoum
Department of computer science, UQAM, Montreal, Quebec, H3C 3P8, Canada
Boukadoum.mounir@uqam.ca

Robert Proulx
Department of psychology UQAM, Montreal, Quebec, H3C 3P8, Canada
proulx.robert@uqam.ca

***Abstract*— Autoassociative memories are known for their capacity to learn correlated patterns, complete these patterns and, once the learning phase completed, filter noisy inputs. However, no autoassociative memory as of yet was able to learn noisy patterns without pre-processing or special procedure. In this paper, we show that a new unsupervised learning rule enables associative memory models to locally learn online noisy correlated patterns. The learning is carried out by a dual Hebbian rule and the convergence is asymptotic. The asymptotic convergence results in an unequal eigenvalues spectrum, which distinguish SCRAM from optimal linear associative memories (OLAMs). Therefore, SCRAM develops less spurious attractors and has better recall performance under noise degradation.**

I Introduction

Recurrent unsupervised associative memories are known for their ability to learn correlated patterns [1, 2]. Once trained, these networks can generalize, filter noise and complete patterns. Among the early models, Hopfield's network [1] is the most popular but, since its weights are updates by a simple Hebbian rule, the connections weights grow unbound [3]. One solution proposed to limit this growth is to train the network with a non-iterative learning rule. Thus, the weights are updated only once per pattern. This learning algorithm has the disadvantage of being offline and to require non-local information. Moreover, the storage capacity of the network is limited to approximately 15% of the units [1].

Many other solutions have been proposed to rectify this situation. The most popular solution develops a weight matrix that converges toward an optimal linear solution based on the pseudo-inverse [2]. This learning yields better results at the cost of remaining non–local and non-iterative. To overcome these difficulties many authors proposed iterative learning rules for Optimal Linear Associative Memories (OLAMs) that are locally implemented and still converge toward the optimal linear solution (e.g. [4 – 7]). However, these correlation models are not without problems: they still develop spurious attractors and have offline iterative learning rules. This is mainly due to the fact that the learning rule is finding the weights that approximate the following matrix form equation

$$\mathbf{X} = \mathbf{WX} \tag{1}$$

where $\mathbf{W}$ represents the weight matrix and $\mathbf{X}$ is the matrix resulting from the union of the stimulus-vectors. This, solution is linear and the resulting weight matrix is semi-definite, with all positive eigenvalues being equal. This indicates that all the network's attractors have the same radius of attraction. The equality of the radius results in a large effect of spurious attractors which causes suboptimal recall performance. Moreover, none of the OLAM networks are able to learn in a noisy environment, which limits their usefulness.

In this paper, we introduce a new online learning rule that uses the feedback from the transmission rule to correct the weight updates. As we will show, this simple time delayed learning allow the network to learn online correlated patterns with fewer spurious attractors. This reduction of spurious states increases the network's performance over pattern degradation. Moreover, contrarily to other OLAM type models, the proposed model is able to develop the correct stimulus-space even in a noisy environment.

II Statistically Converging Recurrent Associative Memory (SCRAM)

Like all others artificial neural networks, SCRAM can be entirely described by its architecture, transmission rule and learning rule.

A. Architecture

SCRAM's architecture is illustrated in Figure 1. It architecture is identical to the one proposed by Hopfield [1] and it key feature is the presence of a feedback loop that creates the dynamism essential to pattern completion.

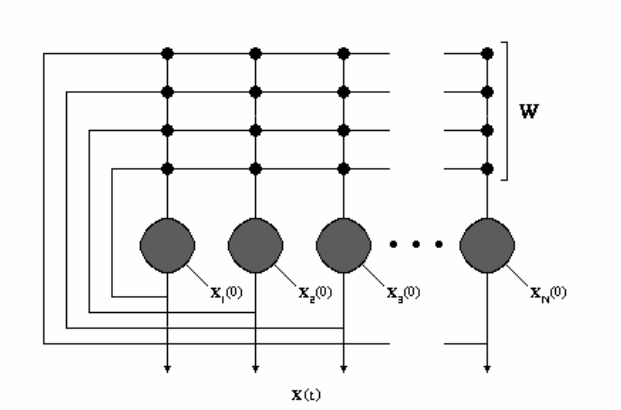


Fig. 1. SCRAM's architecture

B. Transmission Rule

The transmission in the network is a variant of Hopfield's network [8] and is expressed by

$$\mathbf{x}_{[t]} = Tanh(\phi \mathbf{W} \mathbf{x}_{[t-1]}) \tag{2}$$

where $\mathbf{W}$ represents the connection weight matrix, $\mathbf{x}_{[t]}$ the state vector at time t and ϕ (ϕ>0) is a transmission parameter that determine the slope of the function. The higher the value of ϕ, the higher the slope will be. This transmission function has the advantage of being continuous and differentiable in the real domain. Moreover this rule can be directly integrated into the learning rule whereas in OLAM models it was not possible without learning convergence problems.

C. Learning Rule

Most of the network's new properties are a result of the new learning rule. The innovative character of the proposed learning rule is a consequence of the network's dynamical structure. More precisely, weight updates are based on time-delayed dual Hebbian learning. This learning rule is expressed by the following equations

$$\Delta \mathbf{W}_{[k]} = \eta(\mathbf{x}_{[0]}\mathbf{x}_{[0]}^{\mathrm{T}} - \mathbf{x}_{[t]}\mathbf{x}_{[t]}^{\mathrm{T}}) \tag{3}$$

$$\mathbf{W}_{[k+1]} = \zeta \mathbf{W}_{[k]} + \Delta \mathbf{W}_{[k]} \tag{4}$$

where $\mathbf{x}_{[0]}$ represents the initial input vector, η is a general learning parameter ($\eta > 0$), ζ is a general memory efficiency parameter ($0 << \zeta \leq 1$) and k is the learning trial number. It is easy to see from eqn. 3 that, when the network's feedback corresponds exactly to the initial input, the weight update is null and the learning stops. It is noted that the feedback comes directly from the output (contrarily to [4]) thus making the rule online.

However, because we use an asymptotic transmission function, the weights will always be updated by a small amount, which will result in an unequal eigenvalue spectrum. The inequality of the eigenvalues will increase the network's performance by leading to unequal radius of attractions. Moreover, if ζ is smaller than 1, the learning will converge and the eigenvalue spectrum will remain unequal without loss of recall performance.

III Model Analysis

A. Convergence.

By simple analysis, one can easily see that the learning rule solves the following equation (instead of eqn. 1).

$$\mathbf{X} = Tanh(\phi \mathbf{W} \mathbf{X}) \tag{5}$$

To show the model learning convergence we decompose the weight matrix updates in its corresponding eigenvalue updates. To simplify the demonstration we assume that input are bipolar and orthogonal between each other. Then eqn. 3 and 4 becomes

$$\Delta \mathbf{W}_{[k]} = \eta(\mathbf{x}_{[0]}\mathbf{x}_{[0]}^{\mathrm{T}} - \mathbf{x}_{[t]}\mathbf{x}_{[t]}^{\mathrm{T}})$$
$$\Delta \mathbf{W}_{[k]} = \eta(\mathbf{x}_{[0]}\mathbf{x}_{[0]}^{\mathrm{T}} - \mathrm{Tanh}(\phi \mathbf{W} \mathbf{x}_{[t-1]})\mathrm{Tanh}(\phi \mathbf{W} \mathbf{x}_{[t-1]})^{\mathrm{T}})$$
$$\Rightarrow \Delta \lambda_{[k]} = \eta(R * R - \mathrm{Tanh}(\phi \lambda_{[k]} R)\mathrm{Tanh}(\phi \lambda_{[k]} R)),$$

$$\Delta \lambda_{[k]} = \eta(R^2 - \mathrm{Tanh}^2(\phi \lambda_{[k]} R)) \tag{6}$$

and

$$\lambda_{[k+1]} = \zeta \lambda_{[k]} + \Delta \lambda_{[k]} \tag{7}$$

where, λ represents the eigenvalue and R an input vector elements. The network reaches an equilibrium state when the input pattern remains unchanged after a single iteration.

$$\lambda_{[t+1]} = \zeta \lambda_{[t]} + \eta(R^2 - Tanh^2(\phi \lambda_{[t]} R)) = \lambda_{[t]}$$
$$\Rightarrow (1-\zeta)\lambda_{[t]} = \eta(R^2 - Tanh^2(\phi \lambda_{[t]} R)) \tag{8}$$

The right part describes the amount of eigenvalue update, as describe by eqn. 6, and the left part the effect of the memory efficiency parameter. The convergence of the network will depend on two variables: The value of the efficiency parameter and the value of R.

B. Condition R = ±1 and ζ = 1

In this condition, all the elements of an input vector are set to ±1. Example: x = [1, -1, 1, …, 1]$^{\mathrm{T}}$. In this case eqn. 8 simplifies to

$$0 = \eta(1 - \mathrm{Tanh}^2(\phi\lambda_{[k]})) \tag{9}$$

This equation is quadratic and thus yields two roots. However, those roots are -∞ and +∞, as shown by the following solution

$$\Rightarrow \lambda_{[k]} = \pm\frac{\mathrm{ArcTanh}(1)}{\phi}$$
$$\Rightarrow \lambda_c = \lambda_{[k]} = \pm\infty \tag{10}$$

Thus at convergence (λ_c), the eigenvalues of the weight connections will be infinite. However, the variation of the eigenvalue decreases asymptotically to zero, as suggested by Figure 2, contrarily to [3]. For example, the eigenvalue variation will be less than 4.5×10^{-7} for a small value of $\lambda > 4$ (η = 1, ζ = 1 and ϕ = 2). Thus this model clearly distinguishes it self compare to BSB model where the eigenvalue variation increases, instead of decreases, as the number of learning trial increase.

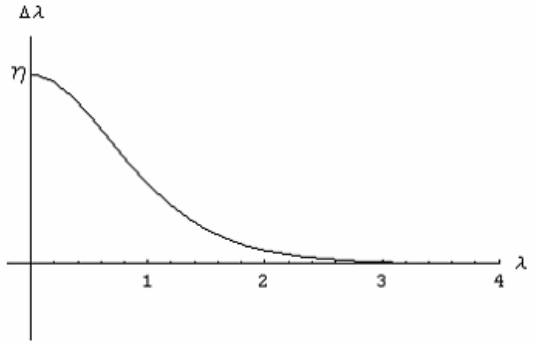


Fig. 2. Eigenvalue growth in function of its variation

This property has two consequences. First, because of the transmission rule, the output is never equal to the input. Second, in correlated pattern learning procedure, the eigenvalue spectrum will be unequal, without domination of the first eigenvalue. Moreover, the simulations will show that these particular values of SCRAM's parameters gives better recall performance than other popular OLAM networks.

Next, we analyze the learning parameter to determine what its optimal value is. To accomplish this we must find the derivative of eqn. 7 for a positive slope.

$$\frac{d\lambda_{[k+1]}}{d\lambda_{[k]}} = 1 - 2\eta\phi\,\mathrm{Sech}^2(\phi\lambda_{[k]})\mathrm{Tanh}(\phi\lambda_{[k]}) > 0 \tag{11}$$

However, when solved for η when the eigenvalue has converged ($\lambda_{[k]} = \lambda_c = \infty$), the solution is indeterminate. To overcome this problem the value of η must be considered when R is not bipolar but tend to. In this case the value of η tends to infinity. Consequently, there is no restriction on the learning parameter's value, but we utilize a conservative limit that has been used in most neural networks algorithms.

$$\eta < 1/N \tag{12}$$

where N represents the network's dimension. It is noted that this analytic solution is valid only if the prototype values are exactly at ±1 and ζ = 1.

C. Condition $|R| < 1$ and $\zeta = 1$

In this condition, all the elements of an input vector are set to equal magnitude bipolar real values. For instance, R = ±0.9: x = [0.9, -0.9, 0.9, … , 0.9]^T. In this case, eqn. 8 becomes

$$0 = \eta(R^2 - \mathrm{Tanh}^2(\phi\lambda_{[k]}R)) \tag{13}$$

Once again, this last equation is quadratic and yield two roots. Those roots are given by

$$\Rightarrow \lambda_{[k]} = \pm\frac{\mathrm{ArcTanh}(R)}{\phi R} \tag{14}$$

It is noted that the eigenvalue only converges to the positive root, because the weights are initially null and are incremented at each time step (as express by the eqn. 4). For example, if we set ϕ = 0.2 and R = ±0.9, we get a convergence value of 0.8179. Figure 3 shows some examples of converging curves as a function of different values of R.

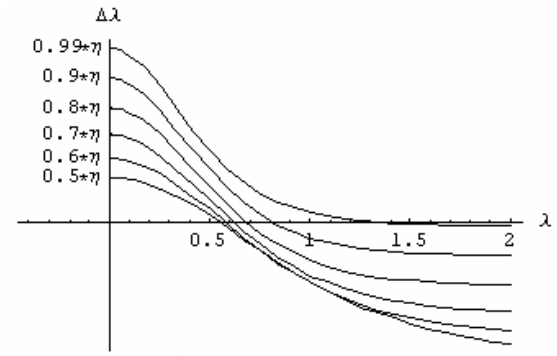


Fig. 3. Eigenvalue growth in function of its variation for different value of $|R|$ (0.99, 0.9, 0.8, 0.7 0.6 and 0.5)

Thus, if we use any bipolar value other that 1, the learning rule converges and, as a consequence, the non-null eigenvalues will be equal (and so will the radius of attraction).

To determine the optimal value of the learning parameter the derivative of the eqn. 7 for a positive slope must be found.

$$\frac{d\lambda_{[k+1]}}{d\lambda_{[k]}} = 1 - 2\eta\phi R\,\mathrm{Sech}^2(\phi\lambda_{[k]}R)\mathrm{Tanh}(\phi\lambda_{[k]}R) > 0 \qquad (15)$$

If we set $\lambda_{[k]} = \lambda_c = \frac{\mathrm{ArcTanh}(R)}{\phi R}$ and solve for η, we get the following general solution for a N units network.

$$\eta < \frac{1}{2\phi R^2(1-R^2)N},\ R \neq \pm 1 \qquad (16)$$

As seen, this last constraint yields larger value for η than the conservative constraint of eqn. 12. Next we analyses the role of the efficiency parameter.

D. Condition R = ±1 and $0<\zeta<1$

If we set the value of the efficiency parameter (ζ) between 0 and 1, from the eqn. 7 the network will reach equilibrium when

$$(1-\zeta)\lambda_{[t]} = \eta(1 - Tanh^2(\phi\lambda_{[t]})) \qquad (17)$$

There is no exact solution to eqn. 17 but Figure 4 gives a general idea of the action of this parameter. For a value of $0 < \zeta < 1$, the two functions intersect at a given λ yielding an equilibrium state. On the other hand, if $\zeta = 1$, the functions do not share any intersection point; therefore, the weights will converge to infinitydiverge (see section III-B).

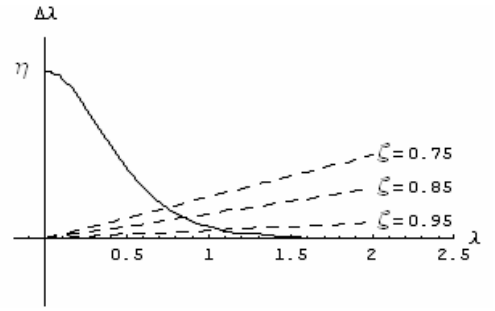


Fig. 4. Variation of the eigenvalue spectrum for a value of ζ smaller than 1

E. Condition $|R| < 1$ and $0<\zeta<1$

In this last condition, the network always converge. More precisely, this condition is close to the model of [4]. However, in our case, the model learns online without inputs normalization whereas it is not the case in [4]. Thus, SCRAM can be seen as a generalization of [4].

IV SIMULATIONS

SCRAM was tested in two conditions. The aim of the first condition was to compare it performance to popular OLAMs on correlated patterns learning. The aim of the second condition was to show the network's ability to learn in a noisy environment. The detailed simulations' methodology follows.

A. First Condition: Noise-Free Learning

The patterns used for the simulations are shown in Figure 5. Each pattern consisted of a letter placed on a 5 x 7 pixel grid, where the white and black pixels were assigned the corresponding values of -1 and +1 (or -0.9 and 0.9 for SCRAM ±0.9 condition). These patterns were chosen because they represent a wide range of correlations. The correlation between the patterns varied from 0.03 to 0.83. It is noted that this stimulus set represents a 34 % memory load, which is more than twice Hopfield's network storage capacity [1].

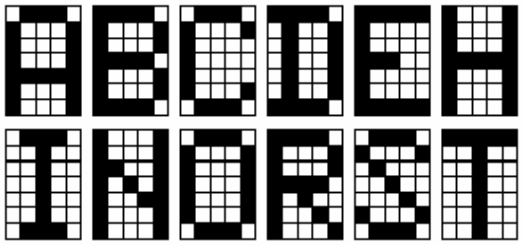


Fig. 5. The twelve stimuli used in the simulations

In the first simulation, the free parameters were set as follow: $\phi = 2$, $\zeta = 1$, $\eta = 1/N = 1/35$ and the number of learning trials was set to 10 000. Those parameters respect the constraint given by eqns. 12 and 16. For the second simulation, the parameters remained unchanged except for ζ which was set to 0.999. This aimed at showing the convergence of the learning when ζ is smaller than 1 and that SCRAM's eigenvalue spectrum would still be unequal. The network's performance was measured by its capacity to correctly recall a given pattern under the addition of a random noise vector. Each noise vector was randomly normally distributed with a mean of 0 and a standard deviations varying from 0 to 2, which gave a noise proportion varying from 0 to 200%.

The other dependant variable was the number of spurious attractors developed by each network. This variable was estimated by calculating the proportion of random vectors that stabilized in spurious states. To accurately evaluate the number of spurious memories, 1 000 random vectors, in which each element is a real number between -1 and 1, were generated. SCRAM's performance was compared with other networks based on the Hopfield type architecture. Those models were proposed by Bégin and Proulx [4],

Diederich and Opper [5], Kanter and Sompolinsky [6], Storkey and Valebregue [7].

B. Second Condition: Noisy Learning

The methodology used in the second condition's simulations was the same as the one previously described. However, a noise vector was added to each stimulus before it presentation to the network. Each noisy vector was normally distributed with a mean of 0 and a standard deviation of 0.2. Fig. 7 shows some examples of noisy Ss. The free parameters were set as follow: $\phi = 2$, $\zeta = 0.999$, $\eta = 1/350$ and the number of learning trials was set to 10 000. It is noted that the learning parameter has been decreased to make sure that the network is not overwhelmed by a bad noise sequence.

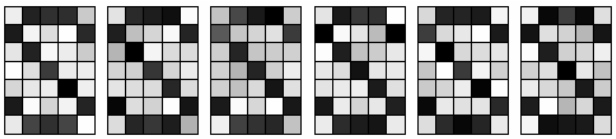


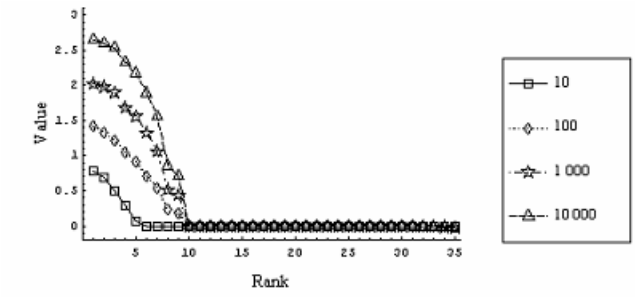
Fig. 6. Examples of noisy inputs used to train the network

V. RESULTS

A. Noise Free Learning

Figure 7a shows the network's eigenvalues development during the learning process. As the number of learning trial increases, the growth of the eigenvalue spectrum decreases. Thus, after 10 000 learning trials, the average weights update was less than 2×10^{-6}. Also, we can see that the network developed only 12 positive eigenvalues, whereas the rest are zero or negative, indicating the memorization of 12 patterns. Figure 7b shows that the eigenvalues did converge, as predicted in section III - *D,* after a finite number of learning trials for the condition $\zeta = 0.999$. The weight matrix remained almost unchanged from 1 000 to 10 000 learning trials.

a



b

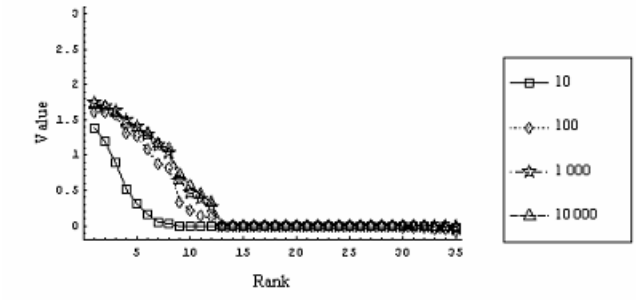


Fig. 7. Eigenvalue spectrum after 10 000 learning trial: a) $\zeta = 1$, b) $\zeta = 0.999$

For the recall task, Figure 8 clearly shows that the condition R=±1 had a better performance compared to the condition R=±0.9. Moreover, setting the efficiency parameter to a value close to 1 (ζ = 0.9999) slightly increased the overall model performance.

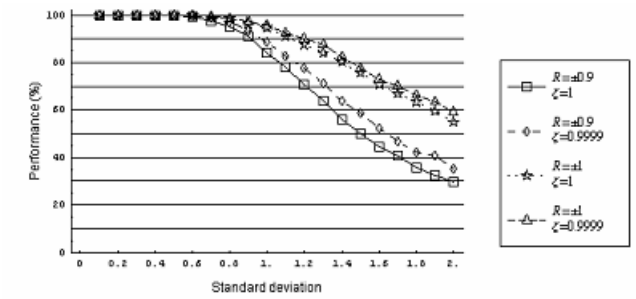


Fig. 8. Recall performances under the addition of normally distributed noise

If we compare SCRAM (condition: R=±1, ζ = 1) with other popular OLAM models (Figure 9) SCRAM's performance is much better. Also, the performance was unaffected by the value taken by the memory efficiency parameter; the model achieve the same performance.

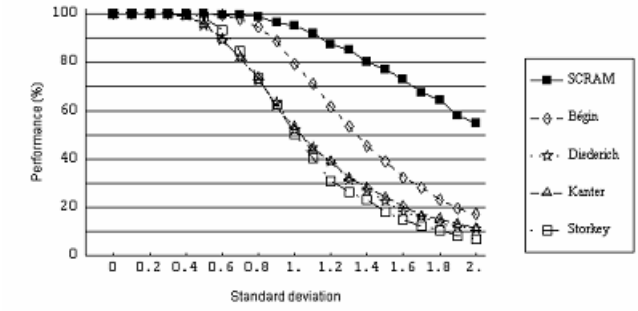


Fig. 9. Recall performances under the addition of normally distributed noise

Finally, the proportion of spurious attractors was greatly reduced by the proposed model. More precisely Figure 10 shows that SCRAM develops a spurious memory proportion of only 30% compared to 96% for Kanter's model, 97% for

Diederich's model and 99% for Bégin's model and Storkey's model!

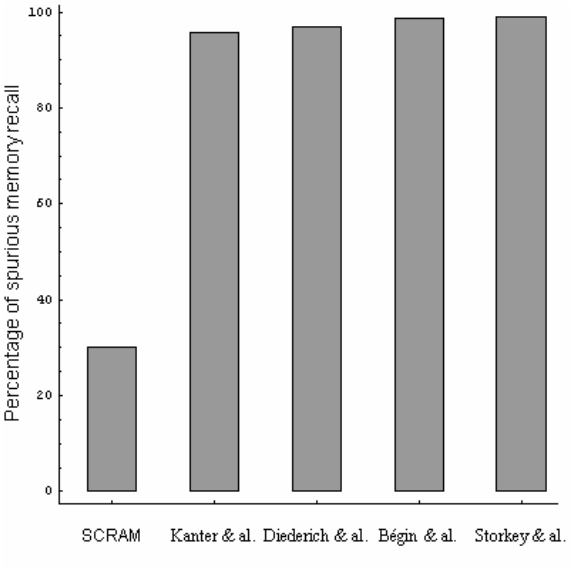


Fig. 10. percentage of spurious memory recall

B. Noisy Learning

Figure 11 shows the eigenvalue spectrum development for the first 10 000 learning trials ($\zeta = 0.999$). It is shown that the 12 first eigenvalues stands out from the 23 others, thus indicating that the network has stored 12 independent stimuli. The network was also able to correctly associate any given corrupted stimulus to the correct, corresponding, noise-free stimulus. In this condition, we were unable to compare SCRAM's performance with other associative memory models because SCRAM was the only model able to develop the correct attractors.

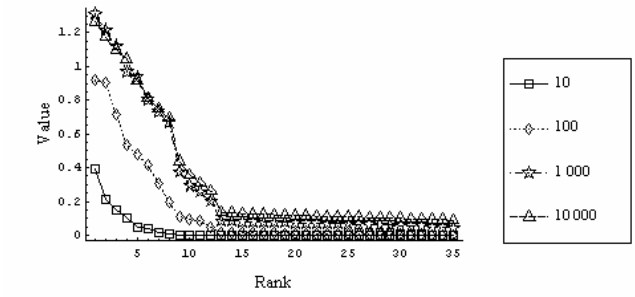


Fig. 11. Eigenvalue spectrum developed by SCRAM in a noisy environment.

VI. CONCLUSION

In this paper, we have shown that incorporating the feedback from the transmission rule into the learning rule enables a network to correctly learn online correlated patterns. Moreover, by using an asymptotic transmission rule, the network developed unequal radius of attraction which resulted in substantially less spurious attractors while maintaining a higher performance over noise degradation. Also, the network was able to learn in a noisy environment without any pre-processing or particular procedure. Those properties make SCRAM unique in that it is well suited to be applied in a wide variety of contexts generally out of the reach of other unsupervised associative memory models.

These new findings broaden the field of possible applications for associative memories. Further studies are still required and should concentrate on the relation between the noise proportion during learning, the value of the learning parameter and the value of the efficiency parameter. Moreover, further work should study how the network behaves in a larger dimension network and how it can be modified to handle grey-level patterns.

REFERENCES

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the Natural Academy of Sciences (U.S.A.)*, vol. 79, pp. 2554-2558, 1982.

[2] L. Personnaz, L. Guyon, L. and G. Dreyfus, "Collective computational properties of neural networks: new learning mechanisms," *Physical Review A*, vol. 34, pp. 4217-4228, 1986.

[3] J. A. Anderson, J. W. Silverstein, S. A. Ritz and R. S. Jones, "Distinctive features, categorical perception, and probability learning: Applications of a neural model," *Psychological Review*, vol. 84, pp. 413-451, 1977.

[4] J. Bégin and R. Proulx, "Categorisation in unsupervised neural networks: The Eidos model," *IEEE Transactions on neural networks*, vol. 7, pp. 147-154, 1996.

[5] S. Diederich and M. Opper, "Learning of correlated pattern in spin-glass networks by local learning rules," *Physical review letters*, vol. 58, pp. 949-952, 1987.

[6] I. Kanter, and H. Sompolinsky, "Associative recall of memory without errors," *Physical Review A*, vol. 35, pp. 380-392, 1987.

[7] A. J. Storkey, and R. Valabregue, "The basins of attraction of a new Hopfield learning rule", *Neural Networks*, vol. 12, pp. 869-876, 1999.

[8] J.J. Hopfield, "Neurons with graded response have collective computational properties like those two-state neurons," *Proceedings of the National Academy of Sciences*, vol. 81, pp. 3088-3092, 1984.