Increasing Attraction of Pseudo-Inverse Autoassociative Networks[1]

Dmitry O. Gorodnichy[a,b] and Alexandre M. Reznik[b]

a. Department of Computing Science, University of Alberta, Edmonton, Canada, T6G 2H1[2]
b. IPMMS, Glushkov Cybernetics Center of Ukrainian Ac.Sc., Kiev, Ukraine

Abstract— **We show *how* partial reduction of self-connections of the network designed with the pseudo-inverse learning rule increases the direct attraction radius of the network. Theoretical formula is obtained. Data obtained by simulation are presented.**

Keywords: pseudo-inverse learning rule, self-connections, attraction radius.

1 Introduction

1.1 The importance of knowing attraction radius

The pseudo-inverse (PI) learning rule (LR), which is also called the projection learning rule [1, 2], has become one of the classical learning rules used for autoassociative memories [3, 4]. PI LR is accredited for its high retrieval capability — it can retrieve from noise up to $50\%N$ prototypes. For comparison, Hebbian correlation LR retrieves only up to $14\%N$ prototypes (N is the size of the network).

The limit of $50\%N$, which is often referred to as a practical storage capacity, is obtained by calculating the attraction radius (AR) of prototypes $Hattr$ of the network. Personnaz *et al.* [1] provides the formula for the AR obtained for orthogonal prototypes

$$Hattr = \frac{N}{2M} \tag{1}$$

and in [2] we obtain the formula for the average AR

$$\langle Hattr \rangle = \frac{\frac{1}{2} - \frac{M}{N}}{\sqrt{\frac{M(N-M)}{N^3}}}. \tag{2}$$

M is the number of memorized prototypes.

As can be seen, when $M \geq N/2$, the AR becomes less than one, which means that the network will not be able to retrieve a prototype even with only one corrupted neuron.

1.2 The influence of self-connections

It is known [5] that self-connections severely restrict the size of attraction basins of prototypes. And in [2] a modification of the PI LR, which consists in partial reduction of self-connections, is suggested. More specifically, this modification is described by equation:

$$C'_{ii} = D \cdot C_{ii}, \quad (0 < D < 1); \quad C'_{ji} = C_{ji} \quad (i \neq j), \tag{3}$$

[1] To be published in *Neural Processing Letters*, volume 5, issue 2, 1997

[2] The address for correspondence. Email: dmitri@cs.ualberta.ca

which is applied after all weights have already been calculated according to PI LR (Eq. 5 below). It is shown that the network with partially reduced self-connections performs much better. In particular, it was illustrated by simulations that the network was able to retrieve from noise up to $75\%N$ prototypes, where prototypes were ordinary 10x10 letter patterns.

In this paper, we provide theoretical explanation of this result, which is based on calculation of the AR of this modified network.

2 Pseudo-Inverse Networks

Consider a fully connected network of N binary neurons, the evolution of which is determined by a synchronous *update rule*:

$$Y_i(t+1) = \mathrm{sgn}\left(\sum_{i=1}^{N} C_{ij} Y_i(t)\right), \ i = 1, ..., N \qquad \text{or in vector form}$$

$$\vec{Y}(t+1) = \mathrm{sgn}\big(\mathbf{C}\vec{Y}(t)\big), \tag{4}$$

where $\vec{Y}(t) \in \{-1,+1\}^N$ is a *state of the network* at time t. $\mathbf{C}$ is a *connection weight* $N * N$ matrix.

Embedding M *prototypes* $\vec{V^1}, ..., \vec{V^M}$ into the network memory is accomplished by adjusting weights C_{ij} according to the *Pseudo-Inverse learning rule*, which in matrix form is

$$\mathbf{C} = \mathbf{V}\mathbf{V}^+ \tag{5}$$

where $\mathbf{V}$ is the matrix made of column prototype vectors, $\mathbf{V}^+$ is the *pseudoinverse* of matrix $\mathbf{V}$.

This rule is obtained from the condition

$$\mathbf{C}\vec{V} = \vec{V}, \tag{6}$$

which ensures that every prototype $\vec{V}$ will be a stable state.

A neural network designed in the way described above will be referred to as the *Pseudo-Inverse (PI) network*. The relationship between the weight values of the PI network and the number of prototypes is obtained in [2]:

$$\langle C_{ii} \rangle = \frac{M}{N} \tag{7}$$

$$\langle C_{ij}^2 \rangle = \frac{M(N-M)}{N^3}, \tag{8}$$

where $\langle X_i \rangle$ designates the average (arithmetical mean) of values X_i.

3 Increasing of the Attraction Radius

3.1 Derivation of the formula

The *attraction radius* (AR) or, more exactly, the *direct attraction radius*, $Hattr$ of a prototype $\vec{V}$ is defined as the largest Hamming distance $H(\vec{V},\vec{Y})$ from within which all vectors $\vec{Y}$ are guaranteed to converge to $\vec{V}$ in a single iteration. As can be seen from Section 1.1, the AR is very important in evaluation of a network; and it is one of few performance criteria which can be estimated theoretically.

Let us calculate the average AR of the PI network with reduced (by Eq. 3) self-connections. For every state $\vec{Y}$ such that $H(\vec{V},\vec{Y}) \le Hattr$ the following must be true for each $i = 1..N$:

$$V_i = \mathrm{sgn}\left(\sum_{i=1}^{N} C'_{ij} Y_i\right) \quad \text{or}$$

$$V_i(\sum_{j=1}^{N} C'_{ij}Y_j) = V_i(\sum_{j=1}^{N} C'_{ij}V_j) - V_i\sum_{h=1}^{H} C'_{ih}(V_h - Y_h) > 0 \tag{9}$$

Noticing that $Y_h = -V_h$ and using Eqs. 3 and 6, we have

$$2V_i\sum_{h=1}^{H} C'_{ih}V_h < V_i(\sum_{j=1}^{N} C'_{ij}V_j) = V_i(\sum_{j=1}^{N} C_{ij}V_j) + (D-1)C_{ii} = 1 + (D-1)C_{ii}$$

Averaging this equation over all neurons, we obtain

$$\langle V_i\sum_{h=1}^{H} C'_{ih}V_h\rangle < \frac{1}{2}(1 + (D-1)\langle C_{ii}\rangle) \tag{10}$$

Denoting left-hand side of Eq. 10 as W, we have the following upper bound of W:

$$W \le \langle V_iC'_{ii}V_i\rangle + \langle V_i\sum_{h=1,h\ne i}^{H} C_{ih}V_h\rangle \le D\langle C_{ii}\rangle + \langle H-1\rangle\langle V_iC_{ih}V_h\rangle \le D\langle C_{ii}\rangle + (\langle H\rangle - 1)\langle|C_{ij}|\rangle$$

Using Eq. 10 for the upper bound of W yields

$$\langle H\rangle \le 1 + \frac{\frac{1}{2}(1-(D+1)\langle C_{ii}\rangle)}{\langle|C_{ij}|\rangle} \tag{11}$$

For every state $\vec{Y}$ such that Eq. 11 is true for $H(\vec{Y},\vec{V})$, Eq. 9 is also true for $H(\vec{Y},\vec{V})$. Hence, Eq. 11 gives the lower limit of the average attraction radius.

Using inequality $\langle C_{ij}\rangle^2 \le \langle C_{ij}^2\rangle$ and Eqs. 7 and 8, we obtain the final result:

$$\langle Hattr\rangle = \frac{\frac{1}{2}(1-(D+1)\frac{M}{N})}{\sqrt{\frac{M(N-M)}{N^3}}} \tag{12}$$

For $D = 1$ Eq. 12 reduced exactly to Eq. 2.

It should be noted that, as we use estimates for weights, which are more precise for large M [2], Eq. 12 is also more precise for large M.

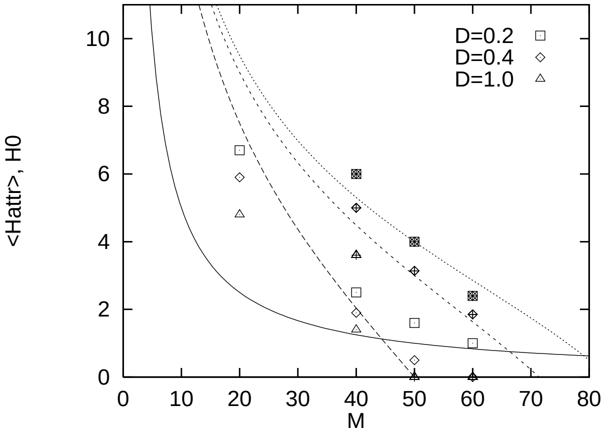
The dependency of Eq. 12 is illustrated in Figure 1, which shows the effect of self-connection reduction on the size of the AR. From the bottom up, the dashed lines correspond to the values $D = 1.0$, 0.4 and 0.2 in Eq. 12. For comparison, we plot the AR given by Eq. 1 by a solid line. As can be seen, decreasing of self-connections increases the AR, making it possible for the network to retrieve prototypes even for $M = 75$ (as $\langle Hattr\rangle > 1$ when $D = 0.2$).

3.2 Simulations

Figure 1 also shows the data obtained by simulation of the model. The network of size $N = 100$ is trained with M prototypes: $M = 20$, 40, 50 and 60. Prototypes are random vectors: 40 out of 100 neurons are clipped randomly to be in the "+1" state, the rest are in the "−1" state. For each M, 10 different prototype sets are created. During examination, $H0$ neurons of a prototype are randomly inverted and one update step (Eq. 4) is executed. Each prototype set is tested with 100 different implementations for each value of noise $H0$.

Dots in Figure 1 show the maximal value of noise $H0$, averaged over all prototype sets and noise implementations, which can be completely eliminated in a single iteration with a probability not less than 0.99 (white dots) and not less than 0.50 (black dots). As can be expected, white dots lie below the average values given by Eq. 2. Black dots show good correlation with the theoretical prediction.

Figure 1: *Attraction radius as a function of number of prototypes M for various values D ($C_{ii} := D \cdot C_{ii}$): estimate of the average value (dashed lines) and experimental data (dots).*



4 Discussion

We have shown that partial reduction of self-connections increases the attraction radius of the pseudo-inverse network. This provides the theoretical basis for the observed improvement of the performance of the network caused by the modification; yet this does not explain it completely. This is because the main property of this modification, as shown in [2], is that it allows the network to escape spurious attractors, which increases the *indirect* attraction radius (IAR). Therefore, knowing the increase of the IAR would give much more information about the improvement. However, as theoretical calculation of the IAR is hard [6], this can be done by simulations only. Another question that needs to be answered is how strongly self-connections can be reduced without the occurrence of cycles. Our simulations show that while $D > 0.2$ in Eq. 3 cycles do not occur. But a theoretical explanation is desirable.

These problems are currently under investigation.

References

[1] L. Personnaz, I. Guyon and G. Dreyfus. Collective computational properties of neural networks: New learning mechanisms, *Phys. Rev. A* vol 34, pp. 4217-4228, 1986.

[2] D.O. Gorodnichy. A way to improve error correction capability of Hopfield associative memory in the case of saturation, *HELNET 94-95 International Workshop on Neural Networks Proceedings* vol. I/II, pp. 198-212, VU University Press, Amsterdam, 1996.

[3] T. Kohonen. *Self-organization and associative memory*, Berlin:Springler, 1984

[4] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation*, Addison-Wesley, 1991.

[5] I. Kanter and H. Sampolinsky. Associative recall of memory without errors, *Phys. Rev. A* vol 35, pp. 380-392, 1987.

[6] P. Floreen and P. Orponen. Attraction radii in binary Hopfield nets are hard to compute, *Neural Computation* 5, pp. 812-821, 1993