Boolean Factor Analysis by Hopfield-Like Autoassociative Memory: Factors Search

Frolov, A. A.
2006

Institute of Computer Science
Academy of Sciences of the Czech Republic

Boolean factor analysis by Hopfield-like autoassociative memory: factors search

A. A. Frolov, D. Husek, P. Y. Polyakov, I. P. Muraviev

Technical report No. 962

February 2006

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: (+420 266 053 230, fax: +420 286 585 789,
e-mail:dusan@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Boolean factor analysis by Hopfield-like autoassociative memory: factors search[1]

A. A. Frolov[2], D. Husek[3], P. Y. Polyakov[4], I. P. Muraviev[5]

Technical report No. 962

February 2006

Abstract:

In our previous paper [6] we have shown that Hebbian learning in Hopfield-like neural network is a natural procedure for Boolean factor analysis. Hebbian learning forms a connection matrix of the network as a co-variation matrix of input signal space. Neurons that represent one common factor are more correlated and thus create attractor of the network dynamics. In this paper we describe a procedure of factors retrieval. According to this procedure, network dynamics evolve forward starting from a random initial state and stabilize in the attractor which corresponds to one of the factors (a true attractor) or one of the spurious attractors. Separation of true and spurious attractors is based on calculation of their Lyapunov function and activation threshold. We studied efficiency of the retrieval procedure by computer simulation.

Keywords:
Boolean factor analysis, Hopfield neural network, unsupervised learning, recall procedure, data mining.

[1] This work was supported by grant from Grant Agency of Czech Republic, Prague No.201/05/0079 and by grant 1ET100300414 of the research program "Information Society" granted by GA AS CR.

[2] Institute of Higher Nervous Activity and Neurophysiology of the Russian Academy of Sciences; e-mail: aafrolov@mail.ru

[3] Institute of Computer Science Academy of Science of the Czech Republice; E-mail: dusan@cs.cas.cz

[4] Institute of Optical Neural Technologies of the Russian Academy of Sciences; e-mail: pavel@8ka.mipt.ru

[5] Institute of Higher Nervous Activity and Neurophysiology of the Russian Academy of Sciences; E-mail: muravevi@mail.ru

1 Introduction

The theoretical analysis and computer simulations performed in our previous paper [6] have revealed that Hopfield-like neural networks are capable of performing Boolean factor analysis of signals of high dimension and complexity. Factor analysis is a procedure which maps original signals into the space of factors. Linear factor analysis implies that each original signal can be presented as

$$\mathbf{X} = \mathbf{F}\mathbf{S} \tag{1.1}$$

where

F is a matrix $N \times L$ of factor loadings and **S** is a vector of factor scores. Each component of **S** gives contribution of the corresponding factor in the original signal. Each column of loading matrix **F** gives presentation of the corresponding factor in the signal space. Below these vectors are termed factors. Mapping of original space to the factor space means that signals are represented by vectors **S** instead of original vectors **X**. Boolean factor analysis implies that a complex vector signal has a form of the Boolean sum of weighted binary factors:

$$\mathbf{X} = \bigvee S_l \mathbf{f}^l. \tag{1.2}$$

In this case, original signals, factor scores and factor loadings are binary and the mapping of original signal to the factor space means citation of factors that were mixed in the signal.

There are many examples of data in the sciences when Boolean factor analysis is required [4]. However, there is no general method to perform it except the fit of the learning set by an exhaustive search. This method evidently requires an exponentially increasing number of trials when the dimension of the pattern space increases. Thus it can be used only when this dimension is relatively small.

For the case of large dimensionality it was a challenge [6] to utilize for Boolean factor analysis the Hopfield-like neural network with parallel dynamics. Due to the correlational Hebbian rule, neurons which represent one factor and therefore tend to fire together, become more tightly connected than neurons belonging to different factors, constituting an attractor of network dynamics. Hence the ability of Hopfield network to perform factor search is determined by two aspects: first, by probability that network activity converges to one of the factors starting from the random state, second, by possibility to distinguish between the factors and the spurious attractors. As shown in [6], this ability depends mainly on two network parameters: sparseness of factors encoding and relative informational loading L/N where L is the total number of factors and N is the network size (which is the dimensionality of signal space). Sparseness is determined by ratio $p = n/N$ of active neurons n in the factor to their total number N. We restrict the study by the case of sparsely encoded factors, i.e. by the case $p \ll 1$. There is a following a priori reason to do this. Boolean superposition of an even relatively small number of densely encoded binary factors, results in input signal vectors with almost components having value 1. It is clear that in the extreme case when all components have value 1, decomposition of signals in factors is evidently impossible and this is the reason why it is impossible to expect successful factor analysis of densely encoded factors. And this why almost of our computer simulations were performed for $p = 0.02$.

In the present study we suggest and investigate a two-run procedure for factor search. The separation of true and spurious attractors is based on calculation of their Lyapunov function.

We treat the mean number of factors mixed in each signal as signal "complexity" C. In the limit case $C = 1$, the considered network reduces to ordinary Hopfield network. As shown in [6], the increase the of complexity drastically decreases the size of the attraction basins around factors. However, this decrease can be completely suppressed by adding one inhibitory neuron to the principal neurons of Hopfield network. We suggest that this procedure allows excluding generally the influence of complexity to network dynamics.

The paper is organized as follows. The pProcedure for factors search is described in Section 2. The limit case of Boolean factor analysis when the input signals are pure factors ($C = 1$), i.e. the ordinary Hopfield network, is investigated in Section 3. The case of a large complexity is investigated in Section 4. The results are discussed in Section 5.

2 Description of learning and recall procedures

The binary patterns of the signal space are treated as activities of N binary neurons (1 - active, 0 - nonactive) with gradually ranging synaptic connections between them. Each pattern of the learning set $\mathbf{X}^m$ is stored in the matrix of synaptic connections $\mathbf{J}'$ according to the correlational Hebbian rule

$$J'_{ij} = \sum_{m=1}^{M} (X_i^m - q^m)(X_j^m - q^m),\ i{\neq}j,\ J'_{ii} = 0, \tag{2.1}$$

where M is the number of patterns in the learning set and bias $q^m = \sum\limits_{i=1}^{N} X_i^m/N$ is the total activity of the m-th pattern. This form of bias corresponds to the biologically plausible global inhibition being proportional to overall neuronal activity.

Additionally to the the N principal neurons of Hopfield network described above we introduced one special inhibitory neuron activated during the presentation of every pattern of the learning set and connected with all principal neurons by bidirectional connections. The patterns of the learning set are stored in the vector $\mathbf{J}''$ of the connections according to Hebbian rule

$$J''_i = \sum_{m=1}^{M} (X_i^m - q^m) = M(q_i - q), \tag{2.2}$$

where $q_i = \sum\limits_{m=1}^{M} X_i^m/M$ is a mean activity of the i-th neuron in the learning set and q is a mean activity of all neurons in the learning set. It is also supposed that the excitability of the introduced inhibitory neuron decreases the inversely proportionally to the size of the learning set being $1/M$ after storing all its patterns. In the recall stage its activity is then

$$A(t) = (1/M) \sum_{i=1}^{N} J''_i X_i(t) = (1/M)\mathbf{J}''^T \mathbf{X}(t)$$

where $\mathbf{J}''^T$ is transposed $\mathbf{J}''$. Respectively, the inhibition produced in all principal neurons of the network is given by vector $\mathbf{J}''A(t) = (1/M)\mathbf{J}''\mathbf{J}''^T\mathbf{X}(t)$. Thus, the inhibition is equivalent to the substraction of $\mathbf{J}''\mathbf{J}''^T/M = M\mathbf{q}\mathbf{q}^T$ from $\mathbf{J}'$ where $\mathbf{q}$ is a vector with components $q_i - q$. As shown in [6], the substraction first completely suppressed two global attractors which dominate in network dynamics for $C \gg 1$, and second, made the size of attractor basins around the factors to be independent of signal complexity C. In fact, the adding of the inhibitory neuron is equivalent to the replacement of common connection matrix $\mathbf{J}'$ by the matrix $\mathbf{J} = \mathbf{J}' - M\mathbf{q}\mathbf{q}^T$.

To reveal factors we used a two-run recall procedure. Its initialization starts by presentation of random initial pattern $\mathbf{X}^{in}$ with $k_{in} = r_{in}N$ active neurons. Activity k_{in} is supposed to be much smaller than the activity of factors, i.e. $r_{in} \ll p$. On presentation of $\mathbf{X}^{in}$, network activity $\mathbf{X}$ evolves to an attractor. The evolution is determined by the synchronous dynamics equation in discrete time. At each time step:

$$X_i(t+1) = \Theta(h_i(t) - T(t)),\ i = 1, \cdots, N, \qquad X_i(0) = X_i^{in} \tag{2.3}$$

where h_i are components of the vector of synaptic excitations

$$\mathbf{h}(t) = \mathbf{J}\mathbf{X}(t), \tag{2.4}$$

$\mathbf{J} = \mathbf{J}' - M\mathbf{q}\mathbf{q}^T$, Θ is the step function, and $T(t)$ is activation threshold.

At each time step of the recall process, the threshold $T(t)$ was chosen in such a way that the level of the network activity was kept constant and equal to k_{in}. Thus, on each time step k_{in} "winners" (neurons with the greatest synaptic excitation) were chosen and only they were active on the next time step. To avoid uncertainty in the choice of winners when several neurons had synaptic excitations at the level of the activation threshold, small random noise was added to the activation threshold of each individual neuron. The amplitude of the noise was put to be less than the smallest increment of the synaptic excitation given by formula (2.4). This ensured that neurons with the highest excitations were kept to be winners even if the random noise was added to the neurons' thresholds. The noise to individual neurons was fixed during the whole recall process to provide its convergence. As shown in [6], this choice of activation thresholds ensures stabilization of the network activity in point or cyclic attractor of length two.

When the activity stabilizes, at the initial level of activity k_{in}, $k_{in}+1$, neurons with maximal synaptic excitation are chosen for the next iteration step, and network activity evolves to some attractor at the new level of activity $k_{in}+1$. Then the level of activity increases to $k_{in}+2$, and so on, until the number of active neurons reaches the final level $r_f N$ with $r_f > p$. Thus, one trial of the recall procedure contains $(r_f - r_{in})N$ external steps and several steps inside each external step to reach an attractor for fixed level of activity.

At the end of each external step Lyapunov function was calculated by formula

$$\Lambda = \mathbf{X}^T(t+1)\mathbf{J}\mathbf{X}(\mathbf{t}), \tag{2.5}$$

where $\mathbf{X}^T(t+1)$ and $\mathbf{X}(t)$ are two network states in cyclic attractor (for point attractor $\mathbf{X}^T(t+1) = \mathbf{X}(t)$). Selection of true and spurious trials was based on the analysis of the change of Lyapunov function and activation threshold along each trajectory of network dynamics. In the following we ordinary use relative Lypunov function $\lambda(r) = \Lambda(r)/rN$ that is a mean synaptic excitation of neurons belonging to some attractor at the end of the external step with $k = rN$ neurons.

Computer simulation revealed that sizes of attraction basins around factors are distributed in a large range. When initial states are chosen randomly network activity tends to converge to factors with larger attraction basins. To suppress dominance of these factors we deleted them from network memory according to Hebbian unlearning rule by substraction ΔJ_{ij} from synaptic connections J_{ij} where

$$\Delta J_{ij} = \eta[(X_i(t)-r)(X_j(t+1)-r) + (X_i(t+1)-r)(X_j(t)-r), j \neq i \tag{2.6}$$

where $\mathbf{X}(t)$ and $\mathbf{X}(t+1)$ are successive patterns of network activity in the attractors and η is an unlearning rate. Due to this unlearning, factors that dominate in network dynamics and attract network activity the most often are gradually deleted from the network memory. Then factors with a little smaller attraction basins become to dominate which are deleted in turn until all factors would be revealed.

3 Simple signals of a learning set

To clarify peculiarities of the recall procedure we initially consider ordinary Hopfield network, i.e. "Boolean factor analysis" in the extremity of $C = 1$ when signals of a learning sets are factors themselves. Moreover according to results obtained in [6] we suggest that due to additional inhibition, network dynamics become independent of signals complexity at all. Thus the results obtained for $C = 1$ are expected to be valid for any complexity. We verify this expectation in the next Section.

In computer simulations we used factors that were uniformly distributed in B_n^N [6] (i.e. each factor contained exactly pN ones and $(1-p)N$ zeros), every pattern of the learning set was one of the factors and each factor appeared in learning set only once, i.e. network learning was performed by the same way as for ordinary Hopfield network.

3.1 Separation of true and spurious attractors

As an example, Fig. 1 demonstrates changes of relative Lyapunov function λ along the trials of the recall procedure for $p = 0.02$, $N = 3000$ and $L = 0.7N$. Each trial was initiated by random pattern with $r_{in} = 0.005$.

[6] $B_n^N = \{\mathbf{X}|X_i \in \{0,1\}, \sum_{i=1}^{N} X_i = n\}$

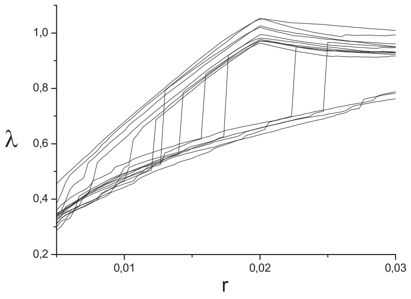

Fig. 1 *Relative Lyapunov function λ in dependence on the relative network activity r for common Hopfield network, $N = 3000$, $L = 0.7N$. The data were normalized to the mean of Lyapunov function for true attractors at $r = p$.*

Trajectories of network dynamics form two separated groups. As shown in Fig. 2, the trajectories with higher values of Lyapunov function are true and with lower ones are spurious. This Figure relates values of Lyapunov function for patterns of network activity at points $r = p$ to maximal overlaps of these patterns with factors. Overlap between two patterns $\mathbf{X}^1$ and $\mathbf{X}^2$ with p active neurons was calculated by formula

$$m(\mathbf{X}^1, \mathbf{X}^2) = \frac{1}{Np(1-p)} \sum_{i=1}^{N} (X_i^1 - p)(X_i^2 - p)$$

By this formula overlap between equal patterns is equal to 1 and mean overlap between independent patterns is equal to 0. Patterns with high Lyapunov function have high overlap with one of the factors, while patterns with low Lyapunov function are far from all the factors. It is shown that true and spurious trajectories are separated by the values of their Lyapunov functions. In Figs 1 and 2 and in the following the values of Lyapunov function are normalized by mean value of this function over true attractors at the point $r = p$.

The second characteristic feature of true trajectories is the existence of a kink at a point $r = p$ where the level of network activity coincides with that in factors (see Fig. 1). When $r < p$ the increase of r results in almost linear increase of the relative Lyapunov function. Increase of r occurs in this case due to joining of neurons belonging to the factor that are strongly connected with other neurons of factor. Then joining of new neurons results in proportional increase of mean synaptic excitation to the active neurons of factor that is just equal to their relative Lyapunov function. When $r > p$ the increase of r occurs due to joining of some random neurons that are connected with factor by week connections. Thus, the increase of the relative Lyapunov function for true trajectory sharply slows and it tends to the values of Lyapunov function for spurious trajectories.

The third characteristic feature of true trajectories which allows for their distinction from spurious trajectories, lies in the behavior of their activation thresholds. Activation thresholds T at the final states of external steps are shown in Fig. 3 for the same network parameters as in Fig. 1. Initially activation thresholds for true attractors are larger in comparison with spurious ones. However at the point $r = p$ they sharply drop to the level of spurious attractors. This behavior is originated from the well known fact (see, for example, [5]) that during activation of the fragment of one of the stored patterns the distribution of synaptic excitations has two separated high and low modes: high - for neurons belonging to the pattern and low - for neurons not belonging to it. The mean of the high mode increases proportionally to the size of the fragment and the mean of the low mode is close to zero. When $r < p$ fragments of stored patterns are activated along the true trajectories. Then activation thresholds are inside the high mode. The mean of high mode increases when r increases since r is the relative size of activated fragment of the stored pattern. Consequently the activation thresholds increase with r. However when the stored pattern is activated totally ($r = p$), the activation threshold has to jump to the low mode to activate additional neurons not belonging to the pattern as r continues to increase.

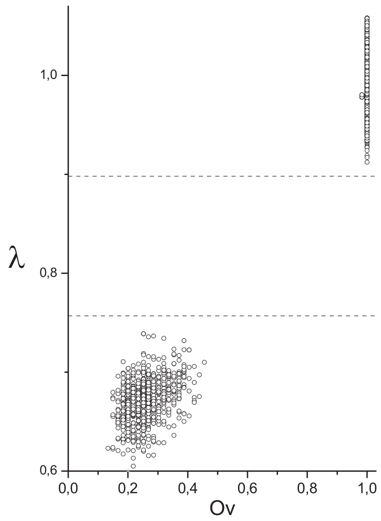

Fig. 2 *Values of normalized Lyapunov function in relation to overlaps with the closest factors for common Hopfield network, $N = 3000$, $L = 0.7N$. It is shown that true and spurious attractors can be easily separated due to a large gap between distributions of values of Lyapunov function.*

The use of these three features of true trajectories provides reliable recognition of patterns stored in Hopfield network.

3.2 Probability of true trials during random search

Another requirement to the procedure of factors search concerns its ability to reveal all factors (stored patterns in the case of common Hopfield network). Two properties of network dynamics could disturb this ability: the dominance of spurious attractors that prevents activation of true attractors at all and the dominance of several true attractors that prevents activation of all others. We show in this subsection that the first problem is actually principal and there exists the critical loading L that restricts network ability of factors search. However, as shown in the next subsection, the second problem can be easily solved by the unlearning of dominant factors.

There are two reasons why spurious attractors become to dominate when L increases: first, due to the increase of their Lyapunov function; second, due to the increase of their amount. It is well known (see, for example, [2], [1]) that Lyapunov function of spurious attractors increases when relative loading L/N increases, and amount of spurious attractors increases exponentially when network size N increases. Thus we can expect the existence of two limits that restrict network ability for factors search. One relates to the critical relative loading and another - to the critical network size under fixed relative loading.

To find the first limit we estimated the values of normalized Lyapunov function λ for spurious attractors in dependence on L/N when $N \to \infty$. The values of λ obtained by averaging across 10000 spurious trials are shown in Fig. 4 for $r = p$. For each ratio L/N experimental points were fitted by linear regression function. The regression intercept was treated as estimation of normalized Lyapunov function for $N \to \infty$ under fixed L/N. The obtained estimates are presented in Fig. 5.

Normalized Lyapunov function of spurious attractors reaches the value 1 that is the value of normalized Lyapunov function for true states at $L \simeq 2.8N$. Thus, the critical loading L_1 when spurious attractors become dominate due to their large Lyapunov function amounts to about $L_1 = 2.8N$.

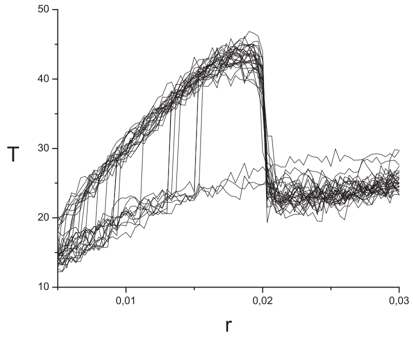

Fig. 3 *The change of activation threshold T along the recall process for common Hopfield network, $N = 3000$, $L = 0.7N$. Notice the drop of activation threshold at the point $r = p$.*

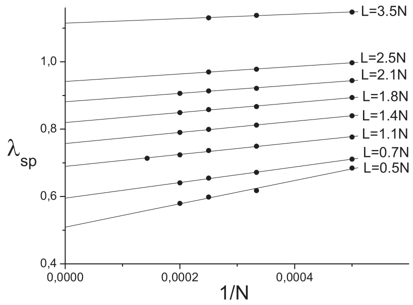

Fig. 4 *Normalized Lyapunov function of spurious attractors for common Hopfield network for $r = p$ in dependence on L/N and $1/N$. Each point was obtained by averaging over 10000 trials. Experimental points are fitted by straight lines.*

To estimate the second limit we analyze the probability of transitions from spurious to true trajectories along the recall process. As shown in Fig. 1 the most trajectories start as spurious but many of them transform to true ones. Transitions from spurious to true trajectories occur during the whole recall process and the probability P_{spur} that the trajectory is spurious monotonically decreases when r increases. As an example, probability P_{spur} in dependence on r and N is shown in Fig. 6 for $L = 0.7N$. Each point in Fig. 6 was obtained over 10000 trials. It quickly drops when network size N is relatively small and remains high for large N. Thus, actually, when network size is relatively small the most trajectories become true during the recall process and they can be used for factors recognition. However, when network size increases almost all trajectories are spurious and the recall procedure becomes incapable of factors search.

Fig. 7 demonstrates the probability of transition from spurious to true attractor P_{trans} in dependence on r and N. Probability P_{trans} was calculated as

$$P_{trans} = (P_{spur}(k) - P_{spur}(k+1))/P_{spur}(k)$$

It has maximum around $r = 0.01$. As shown in Fig. 1 the point of maximum corresponds approximately to the point where the values of Lyapunov function for true trajectories become markedly higher then those for spurious trajectories and, thus, true trajectories become more attractive. For $r > 0.01$ $\ln P_{trans}$ linearly decreases when r increases. The slope of this decrease appeared to be independent of N. However totaly, P_{trans} decreases when N increases. The obtained dependence of

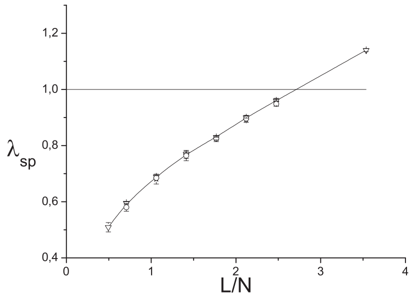

Fig. 5 *Normalized Lyapunov function of spurious attractors for* $r = p$. *The points are obtained as intercepts of regression lines as shown in Fig. 4.* $\bigtriangledown$ *- common Hopfield network* ($C = 1$), $\star$ *-* $C = 10$, $\circ$ *-* $C = 20$. *Thin solid line - B-Spline approximation of experimental points. Horizontal line gives the mean value of Lyapunov function for true attractors.*

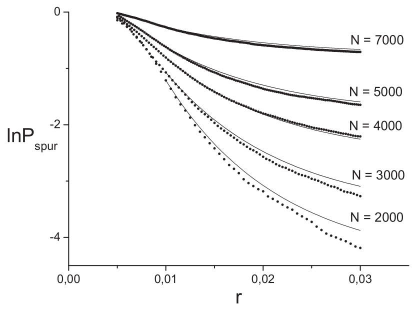

Fig. 6 *Probability* P_{spur} *in dependence on* r *and* N *for common Hopfield network,* $L = 0.7N$. $P_{spur}(r)$ *is a probability that a trajectory remains to be spurious until given* r. *Each experimental point was obtained over 10000 trials of computer simulation. Solid lines are approximations of experimental data by formula (3.2).*

P_{trans} on r and N can be fitted by the following regression model:

$$\ln P_{trans} = -aN - br \tag{3.1}$$

where $a = (6.2 \pm 0.1) \cdot 10^{-4}$ and $b = 82 \pm 5$.

For sufficiently large N when r can be considered as continuous variable, the probability P_{spur} has a form

$$P_{spur}(r) = \exp(-N \int_{r_{in}}^{r} P_{trans}(x)dx).$$

Then according to (3.1)

$$P_{spur}(r) = \exp[-\frac{N}{b} \exp(-aN)(\exp(-br_{in}) - \exp(-br)))]. \tag{3.2}$$

Fig. 6 demonstrates the accuracy of the used approximation. It is less accurate for smaller N due to the effect of discontinuity of r along the trials.

Coefficient a in regression model (3.1) happened to be proportional to L/N and was presented as $a = c_1 L/N$, while coefficient b could be presented as $b = c_2 + c_3 L/N$. Thus as a whole, dependence of P_{trans} on L, N and r could be presented as

$$\ln P_{trans} = -c_1 L - c_2 r - c_3 r L/N. \tag{3.3}$$

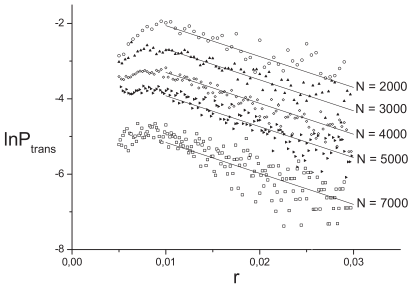

Fig. 7 *Probability of transition P_{trans} from spurious to true attractor for common Hopfield network, $L = 0.7N$. Each experimental point was obtained over 10000 trials of computer simulation. Solid lines are approximations of experimental data by formula* 3.1.

Coefficients c_i were found by the best fit over the whole set of tested network parameters: $L/N = 0.5$, 0.7, 1, 1.4 and $N = 2000$, 3000, 4000, 5000, 7000. Each value P_{trans} was calculated over 10000 trials. Coefficients c_i were estimated as $c_1 = (8.8 \pm 0.1) \cdot 10^{-4}$, $c_2 = 25.5 \pm 3$ and $c_3 = 80 \pm 2$. According to (3.2) the probability that a trial finally happened to be true, i.e. $1 - P_{spur}(r_f)$, is mainly defined by the term $aN - \ln(N/b) = c_1 L - \ln(N/(c_2 + c_3 L/N))$. The probability is relatively high when this term is small. Thus the probability of true trials is relatively high when

$$L < L_2 = \ln(N/(c_2 + c_3 L/N))/c_1 \tag{3.4}$$

and it drops to zero when this condition breaks. Due to proportionality of coefficient a to L/N, the critical condition (3.4) happened to depend mainly on absolute loading L and only logarithmically on the network size N. Therefore for large N critical absolute loading L_2 is reached for small relative loading. In this case one can ignore the term $c_3 L/N$ comparing with c_2 in (3.4) and rewrite this condition as $L < L_2 \simeq \ln(N/c_2)/c_1 \simeq \ln(0.04N) \cdot 10^3$. The random search of factors is possible when L satisfies both conditions $L < L_1$ and $L < L_2$. For $N < 2000$ the first condition restricts the network ability for factors search, and for $N > 2000$ - the second condition.

3.3 Hebbian unlearning

Even in the case when the probability of true trials is rather high, the search of all factors could be impossible due to dominance of some of them. Fig. 8 illustrates this statement: thin line shows the number of found different factors in dependence on number of trials for $N = 3000$, $L = 1.4N$. According to (3.2) for these network parameters the probability of true trial amounts to around 0.1. Initially the number of new found factors increases proportionally to the number of trials with proportionality coefficient equal to the probability of true trials, i.e. 0.1. However later the search of new factors becomes slower because all trajectories are attracted by the fraction of 0.135 of all factors.

This difficulty can be easily overcome by Hebbian unlearning when attractors appeared during the recall process are deleted from the network memory. The deletion was performed by the rule (2.6). Attractors were deleted by substraction of network states obtained for true trials at $r = p$ with unlearning rate $\eta = 1$. Fig. 8 shows the effect of unlearning (thick line). The points are values of Lyapunov function obtained for $r = p$. The densities of points with high and low values of Lyapunov function correspond to the probabilities of true and spurious trajectories. At the initial stage of the recall process the density of first points is initially smaller, that corresponds to the dominance of spurious trajectories. As shown in Fig. 8 initially the fraction of true trajectories amounts to about only ten percents that is the same as in the case without unlearning. However soon the rate of the search of new factors speeds up. This process is accompanied by the decrease of Lyapunov function for both true and spurious trials. For true trials it decreases because of deletion of factors with high values of Lyapunov function. For spurious trials it decreases because deletion of factors is equivalent

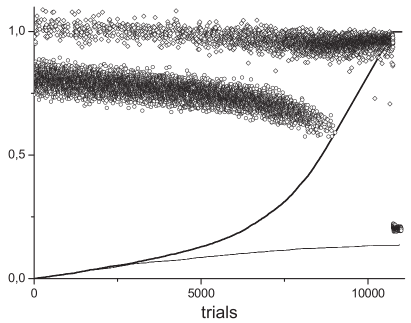

Fig. 8 *Relative Lyapunov function (points) and the fraction of found factors (lines) in dependence on number of recall trials for common Hopfield network, $N = 3000$, $L = 1.4N$. Values of relative Lyapunov function are shown only for the recall with unlearning. In the recall without unlearning, distribution of experimental points does not depend on number of trials and correspond to initial stage of the recall with unlearning. $\diamond$ - true attractors, $\circ$ - spurious attractors. Thick and thin solid lines are fractions of found factors obtained with and without unlearning, respectively.*

to the decrease of relative loading L/N and as shown in Fig. 5, the values of Lyapunov function for spurious attractors monotonically decrease when L/N decreases. Since values of Lyapunov function decreases faster for spurious trajectories, the fraction of true trajectories increases and at the final stage of the recall process the probability of spurious trials falls to zero and each trial results in the retrieval of new factor. Only about 11000 trials are required to reveal all 4200 factors. When almost all factors are found the probability of spurious trials increases again, but they have now very small Lyapunov function.

4 Large complexity of input patterns

In this series of computer experiments the network was trained by a set of M patterns of the form $\mathbf{X}^m = \bigvee_{l=1}^{L} S_l^m \mathbf{f}^l$, where $\mathbf{f}^l \in B_n^N$ are L factors and for every m-th pattern $\mathbf{S}^m \in B_C^L$ is a corresponding vector of factor binary scores. As follows from the definition every factor contains exactly $n = Np$ 1-s. Every complex pattern $\mathbf{X}^m$ contains, in turn, exactly C factors. We assumed factors and factor scores to be statistically independent. Computer experiments were performed for $C = 10$ and 20.

All characteristic properties of true and spurious trajectories in the case of complex input patterns completely coincide with those for simple input patterns. Particularly, Lyapunov function λ is higher for true trajectories, they have specific kink at the curve $\lambda(r)$ and specific drop of activation threshold T at the point $r = p$. The values of normalized Lyapunov function for spurious attractors for $C = 10$ and $C = 20$ are compared in Fig. 5 with those for $C = 1$. The complete coincidence of the results confirms our prediction that the use of the additional inhibitory neuron in the network architecture not only completely suppresses two global attractors but also suppresses the dependence of network properties on C. Thus all three specific properties of true trajectories can be used for their separation from spurious trajectories and the first limit L_1 which restricts the network ability to factor search due to the increase of Lypunov function for spurious attractors, coincides with that obtained above for common Hopfield network.

To find the second limit we used the same approach as in the previous section. Particularly, we obtained the probability of transition from spurious attractors to true attractors in dependence on N and α. This probability was approximated by the same regression model (3.3) as for common Hopfield network. Coefficients c_i of the regression model were found by the best fit of experimental data obtained for $L = 0.7N$ ($N = 2000, 3000, 4000$) and $L = N, 1.4N$ ($N = 2000, 3000$). For $C = 10$ they were estimated as $c_1 = (9.6 \pm 0.6) \cdot 10^{-4}$, $c_2 = 690 \pm 30$ and $c_3 = 254 \pm 18$ and for $C = 20$ $c_1 = (9.0 \pm 0.4) \cdot 10^{-4}$, $c_2 = 680 \pm 20$ and $c_3 = 367 \pm 14$. Coefficient c_1 which mainly defines the critical information loading $L_2 \sim \ln(N/c_2)/c_1$ happened to be close to that of Hopfield network. Hence, the second critical loading L_2 also happened to be independent of C. This is the second argument in support of our prediction that additional inhibitory neuron allowed to exclude the dependence of network properties on input patterns complexity.

To suppress the dominance of several factors in the recall process we used the same procedure of factors unlearning as for common Hopfield network. Fig. 9 demonstrates the effect of unlearning for $N = 3000$, $L = 1.4L$ and $C = 20$. Thin line gives the fraction of factors recalled without unlearning and thick line with unlearning. Only around 600 of 4200 factors were revealed after 16000 trials without unlearning and all of them were revealed with unlearning. It is shown that as for the common Hopfield network the search of new factors speeds up during the recall process with unlearning due to the primary decrease of Lyapunov function for spurious attractors.

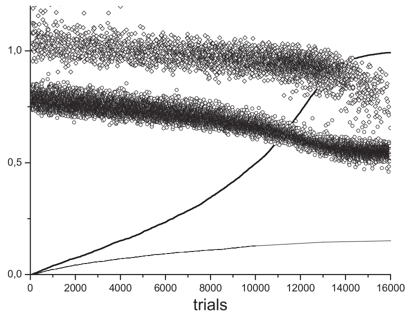

Fig. 9 *Relative Lyapunov function (points) and the fractions of found factors (lines)in dependence on number of recall trials for $C = 20$, $N = 3000$, $L = 1.4N$. ◇ - true attractors, ○ - spurious attractors. Values of relative Lyapunov function are shown only for the recall with unlearning. Thick and solid lines are fractions of found factors with and without unlearning, respectively.*

5 Discussion

In our previous paper [6] we have shown that Hopfield-like neural network is capable of performing Boolean factor analysis of the signals of high dimension and complexity. This ability is based on the fact that, due the to correlational Hebbian rule, factors become attractors of the network dynamics. Adding of one inhibitory neuron to the principal Hopfield network allows for suppression of two global spurious attractors that dominate in network dynamics when signals of learning set have large complexity ($C > 10$). This modification of network architecture allows also to avoid the reduction of sizes of attraction basins around factors which is observed in the network of common architecture. In the present paper we show that it allows to avoid the worsening of network properties in response to complexity increase, in general.

We suggested the simple recall procedure for factors search. Its ability is restricted by two critical high limits of loading L_1 and L_2. The first limit L_1 corresponds to loading L that provides equality of values of Lyapunov function for spurious and true attractors. If $L > L_1$ this value for spurious attractors is larger and they dominate in network dynamics. If $L < L_1$ this value is larger for true attractors, however spurious attractors can continue to dominate because of their huge amount. To avoid this kind of dominance, loading L must be less than L_2. Thus the suggested procedure is capable of factors search when both conditions $L < L_1$ and $L < L_2$ are satisfied. For the most interesting case of large signal dimensionality (N is of the order $10^3 - 10^4$) both limits are rather high. They are also of the order 10^3. The estimates of limit loadings L_1 and L_2 were presented here only for $p = 0.02$. They happened to be even higher for $p = 0.004$. Thus, we believe that for most practical tasks these limits cannot be reached. However the restriction to network performance due to their existence has to be taken into account.

We considered here only a special case of binary signals to be factorized, namely the case when factors were uniformly distributed in the signal space and in the learning set, the number of factors presented in each learning pattern was fixed, the number of active neurons in each factor was fixed and factors were mixed in patterns of learning set without distortion. These constrains seem to be unimportant for the proposed procedure of factors search. According to [6], the value of Lyapunov function for each factor is almost proportional to the number of input patterns which contain it. Thus factors with high probability of appearance in input patterns should have high Lyapunov function and hence high probability to be revealed by random search. Thus, in the case when factors are not uniformly distributed in the learning sets, the proposed procedure should be also valid and the only difference is that the distribution of values of Lyapunov function for factors would be determined by the properties of their distribution in the learning set.

Similarly one cannot expect any difficulty in the search of factors with different level of activity which is unknown in advance. As Fig. 1 and 3 show, specific kink at the curve $\lambda(r)$ and drop at the curve $T(r)$ when $r = p$ are characteristic features of factors that allow for their identification. One can even expect that the proposed procedure could be used to reveal hidden hierarchical organization of input signals when factors with high level of activity represent classes and factors with low level of activity represent objects of classes.

There are many examples of data in the sciences when Boolean factor analysis is required [4]. However Boolean factor analysis with the use of neural network approach seems especially efficient for processing textual data because texts are good example of signals of very large dimensionality that is equal to number of words in the used dictionary. We believe that each topic has some specific set of words that appear together as the document contains this topic. Then this set of words has to create factor. The efficiency of document terms factor analysis has been demonstrated by the method of Latent Semantic Indexing [3]. However, this method is based on linear factor analysis which seems less adequate for textual documents than the binary attempt described in this paper. The neural network approach based on ART was recently applied to text documents classification [7] and showed high quality. This method also implies the existence of some specific combination of words in documents containing specific topic. We believe that the suggested method of Boolean factor analysis should be more efficient in recognizing documents containing several topics. The results of Boolean factor analysis of Reuters Corpus Volume I (RCV1) will be presented in the accompanying paper.

6 Figure legends

Bibliography

[1] Amari S.: Mathematical fondation of neurocomputing. Proceedings of IEEE, 78(9), 1990, pp. 1443-1463.

[2] Amit D. J., Gutfreund H., & Sompolinsky H.: Statistical mechanics of neural networks near saturation. Annal of Physics, **173**, 1987, pp. 30-67.

[3] Berry, M. W., Browne, M. (1999). *Understanding search engines: mathematical modeling and text retrieval.* Philadelphia, PA: SIAM.

[4] De Leeuw J. (2003) Principal component analysis of binary data. Application to roll-call analysis. http://gifi.stat.ucla.edu

[5] Frolov A.A., Husek D., & Muraviev I.P.:Informational efficiency of sparsely encoded Hopfield-like autoassociative memory. Optical Memory & Neural Networks, **12**, 2003, 3, pp.177-197.

[6] Frolov A.A., Husek D., Muraviev I.P. & Polyakov P.Y.: Boolean factor analysis by Hopfield-like autoassociative memory.

[7] Massay L.: On the quality of ART1 text clustering. Neural Networks, v 16, 2003, pp. 771-778.

Contents

1 Introduction . . . 1

2 Description of learning and recall procedures . . . 2

3 Simple signals of a learning set . . . 3

3.1 Separation of true and spurious attractors . . . 3

3.2 Probability of true trials during random search . . . 5

3.3 Hebbian unlearning . . . 8

4 Large complexity of input patterns . . . 9

5 Discussion . . . 10

6 Figure legends . . . 12