ISSN 0006-3509, Biophysics, 2009, Vol. 54, No. 3, pp. 349–354. © Pleiades Publishing, Inc., 2009.
Original Russian Text © E.V. Budilova, M.P. Karpenko, L.M. Kachalova, A.T. Terekhin, 2009, published in Biofizika, 2009, Vol. 54, No. 3, pp. 500–507.

Familiarity Recognition and Recollection: A Neural Network Model

E. V. Budilova[a, b], **M. P. Karpenko**[a], **L. M. Kachalova**[a], **and A. T. Terekhin**[a, b]

[a] *Institute of Cognitive Neurology, Modern University for Humanities, Moscow, 109029 Russia*
[b] *Biological Faculty, Moscow State University, Moscow, 119899 Russia*
e-mail: terekhin@mail.ru

Received April 24, 2007; in final form, August 11, 2008

Abstract—The capacities of a specially designed neural network for familiarity recognition and recollection have been compared. Recognition is based on calculating "image familiarity" as a modified Hopfield energy function in which the value of the inner sum is replaced by the sign of this value. This replacement makes the calculation of familiarity compatible with the basic dynamic equations of the Hopfield network and is in fact reduced to calculating the scalar product of the neuronet state vectors at two successive time steps.

Key words: Hopfield network, energy function, familiarity recognition, recollection

DOI: 10.1134/S0006350909030178

INTRODUCTION

In the context of the memory blocking phenomenon, our preceding work [1] outlined the dramatic discrepancy between the awareness of the person trying to reproduce some information that it is contained in his/her memory and the inability to retrieve this information. This was explained by the difference in the processes of recognition and recollection in the human brain, which manifests itself on the neurobiological as well as on the psychological level. Observations of patients with lesions of the medial parietal lobe (entorhinal, perirhinal, and parahippocampal cortex) and/or of the hippocampus lead to a conclusion that the former is largely responsible for recognition and the latter, for recollection [2, 3]; further, they suggest a central role of the perirhinal cortex in recognition of familiar images (visual, auditory, verbal, etc.) [4, 5]. In experiments with monkeys, about one-fourth of perirhinal cortical neurons actively responded to presentation of a novel image but very weakly and briefly to a familiar one [6].

THE MODEL

Our neuronet model of recognition and recollection is based on a modified Hopfield energy function [7, 8] that we called "familiarity" and defined for a network of N neurons as

$$E^*(X_t) = \sum_{i=1}^{N} x_i(t)\,\mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij}(t)x_j(t)\right), \qquad (1)$$

where $x_i(t)$ and $w_{ij}(t)$ are respectively the state of neuron i and the synaptic weight of its connection to j at time moment t. The dynamics of the network proper is determined by the McCulloch–Pitts rule [9]:

$$x_i(t+1) = \mathrm{sgn}\left(\sum_{j=1}^{n} w_{ij}(t)x_j(t)\right) \qquad (2)$$

and the dynamics of synaptic weights for all $j \neq i$, by the Hebb rule [10]:

$$w_{ij}(t+1) = w_{ij}(t) + x_i(t)x_j(t). \qquad (3)$$

Applying (2), equation (1) converts to

$$E^*(X_t) = \sum_{i=1}^{N} x_i(t)x_i(t+1), \qquad (4)$$

i.e., the "familiarity function" simply equals the scalar product of neuron state vectors in two consecutive time steps. We have also proposed ([1], see also [11]) the architecture of a network calculating the familiarity with (4) and sending the result to an additional, $(N+1)$th "recognition neuron." Two assumptions have been made for the latter: (i) the earlier weights of connections from input neurons to $N+1$ are forgotten, and (ii) $N+1$ is in active state when the new weights form. The second assumption can be re-formulated: the weight for j to $N+1$ increases if j is active, regardless of the state of $N+1$; i.e., for the recognition neuron rule (3) converts to

$$w_{N+1,j}(t+1) = x_j(t). \qquad (5)$$

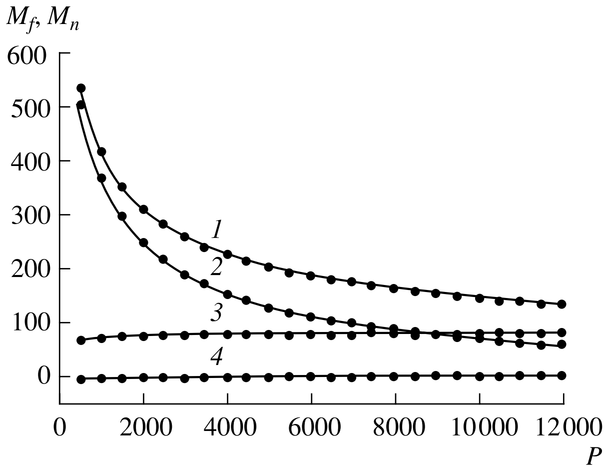

Fig. 1. Means M_f (*1*) and M_n (*4*) and 99% normal bounds $M_f - 2.33S_f$ (*2*) and $M_n + 2.33S_n$ (*3*) of E^* as functions of P for a network of 700 neurons.

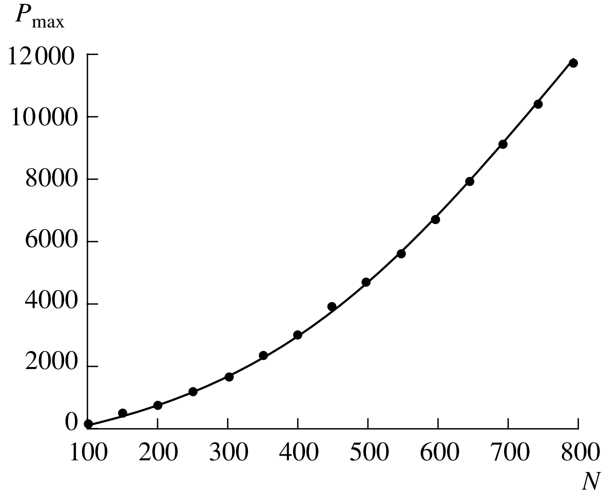

Fig. 2. Dependence of P_{max} on N (approximating curve, $P_{max} = 0.0185N^2$).

However, the question whether such neurons trained by the modified Hebb rule exist in the real brain network as a whole, and in the perirhinal cortex in particular, is clearly beyond the scope of the present work.

Here we assess and compare the possibilities of this neuronet in image recognition and recollection. The Appendix explains the designations and presents the pertinent Matlab programs for calculation.

RECOGNITION OF A FULLY PRESENTED UNDISTORTED IMAGE

The means M_f and M_n with standard deviations S_f and S_n of the familiarity function E^* are calculated for familiar (f) and novel (n) images; then M_f, M_n, $M_f - 2.33S_f$ and $M_n + 2.33S_n$ are plotted versus the number of presented images P. The intersection of the 99% normal boundaries of the $M_f - 2.33S_f$ and $M_n + 2.33S_n$ curves determines the maximal volume whereat the recognition error does not exceed 1%. In Fig. 1, $P_{max} \approx 9000$.

Table 1 gives the estimates of P_{max} vs. N obtained with Program 2, and in Fig. 2 these data are plotted together with the regression curve $P_{max} = 0.0185N^2$.

RECOGNITION OF A DISTORTED IMAGE

At large N, the neuronet recognition capacity $0.0185N^2$ is much greater than its recollection capacity $0.145N$ [1]. However, this pertained to presenting a full image. How will the network perform in recognition when the image is incomplete?

Table 2 shows that at $N = 700$, $P = 100$, and $K = 10$, presenting ~15.5% of image information (*cue*) is sufficient for recognition within 1% error.

However, when P is increased to 4500 (i.e. about half the recognition memory of a 700-neuron network), 1% error in a single series ($K = 1$) is attained only when about 70% of the image is presented (Table 3).

Finally, with $P = 9000$ (at the memory limit) the image must be presented completely to get the same 99% recognition.

FULL IMAGE RECOLLECTION

For comparison, we assessed the possibilities of the same network in reproducing the full image upon presentation of its distorted. The program realized the Hopfield network trained in accordance with (2) and relaxing to the attractor in accordance with (1).

Table 5 lists the results for $N = 700$, $K = 10$, and $P = 10, 20, \ldots, 120$. For each number of presented images (P) and each fraction of the image presented intact

Table 1

N	100	150	200	250	300	350	400	450	500	550	600	650	700	750	800
P_{max}	173	459	774	1218	1667	2387	2987	3912	4677	5587	6657	7842	9087	10406	11599

(*cue* = 0, 0.1, 0.2, …, 1.0), four performance indices were computed: the fraction of completely recollected images among the $K \times P$ presented, the fraction of images recollected by 99%, the fraction of images recollected by 95%; and the fraction of cases when the attractor was reached in at most 15 steps (regardless of recollection accuracy).

One can see that full recollection is possible (in 99% of cases) when initially the network has memorized not more than 50 images; with 10 images memorized, about 20% of correct information must be presented; with 20 images, 25%; with 40 images, >40%; et seq. At $P > 50$ a significant portion of even fully correct images is recollected with distortion; thus at $P = 100$ only 10% of fully presented images are completely recollected, and at $P = 120$ there are none.

If we require that an image be reproduced in 99% of cases with only 99% accuracy, this is possible with complete presentation upon memorizing 75 images; 95% recollection is possible from 85 images.

Thus, it can be concluded that satisfactory recollection from a distorted image by a Hopfield network of 700 neurons is feasible when it memorizes a number of images that is about half of its maximal "recollection memory" ($0.145N \approx 100$). For example, with 50 memorized images, recollection with 99% accuracy in 99% of cases requires only 40% of correct presentation (note that lowering the recollection accuracy to 95% does not reduce the requirement for presentation accuracy, see Table 5). We have seen above that satisfactory recognition of distorted images is possible when the "recognition memory" is also half full (Table 3). However, memory for recognition of distorted images greatly exceeds that for recollection.

Notable is another distinction between the two processes. Recognition is always achieved in two steps of network operation: first the input image X is converted to image Y, then Y is compared with the initial X. If Y is close enough to X, this means that state X is close to one of the network attractors, i.e. one of memorized images (provided of course that the above-specified memory occupancies are not exceeded). In contrast, recollection usually requires 3–6 steps for undistorted familiar images and a greater, maybe indefinitely great, number of steps for unfamiliar or heavily distorted images. One can even suppose that the brain network has mechanisms that block further search in memory (and conversely, stimulate memorization) if an image is not recognized in two steps after presentation.

Table 2

a	*b*	*c*	*d*	*error*	*cue*
861	139	8	992	0.074	0.10
904	96	6	994	0.051	0.11
948	52	11	989	0.032	0.12
974	26	7	993	0.017	0.13
983	17	5	995	0.011	0.14
983	17	7	993	0.012	0.15
995	5	10	990	0.008	0.16
997	3	10	990	0.007	0.17
999	1	7	993	0.004	0.18
1000	0	8	992	0.004	0.19
1000	0	8	992	0.004	0.20

Table 3

a	*b*	*c*	*d*	*error*	*cue*
3785	715	35	4465	0.083	0.50
4341	159	33	4467	0.021	0.60
4468	32	40	4460	0.008	0.70
4499	1	37	4463	0.004	0.80
4500	0	35	4465	0.004	0.90
4500	0	41	4459	0.005	1.00

Table 4

a	*b*	*c*	*d*	*error*	*cue*
4121	4879	80	8920	0.276	0.50
5940	3060	85	8915	0.175	0.60
7393	1607	65	8935	0.093	0.70
8277	723	76	8924	0.044	0.80
8699	301	86	8914	0.022	0.90
8898	102	97	8903	0.011	1.00

APPENDIX

Matlab Programs for Neuronet Calculations

Designations used throughout:

N—number of neurons in the Hopfield network;

P—number of images presented for training and for recognition;

Q—number of random images presented only for recognition;

Table 5

P	*cue*	100%	99%	95%	*r* < 16	*P*	*cue*	100%	99%	95%	*r* < 16
10	0	0.04	0.04	0.03	0.96	50	0.1	0.01	0.01	0.02	0.03
10	0.1	0.56	0.56	0.72	0.98	50	0.2	0.23	0.26	0.29	0.28
10	0.2	0.98	0.98	1	1	50	0.3	0.85	0.89	0.88	0.89
10	0.3	1	1	1	1	50	0.4	0.94	0.99	0.99	0.99
10	0.4	1	1	1	1	50	0.5	0.93	1	1	1
10	0.5	1	1	1	1	50	0.6	0.94	1	1	1
10	0.6	1	1	1	1	50	0.7	0.94	1	1	1
10	0.7	1	1	1	1	50	0.8	0.97	1	1	1
10	0.8	1	1	1	1	50	0.9	0.94	1	1	1
10	0.9	1	1	1	1	50	1	0.97	1	1	1
10	1	1	1	1	1	60	0	0	0	0	0.02
20	0	0.02	0.02	0.02	0.41	60	0.1	0	0	0	0.02
20	0.1	0.40	0.41	0.40	0.57	60	0.2	0.07	0.10	0.08	0.12
20	0.2	0.97	0.97	0.95	0.97	60	0.3	0.47	0.57	0.57	0.58
20	0.3	1	1	1	1	60	0.4	0.79	0.96	0.96	0.96
20	0.4	1	1	1	1	60	0.5	0.81	0.99	1	1
20	0.5	1	1	1	1	60	0.6	0.82	0.99	1	1
20	0.6	1	1	1	1	60	0.7	0.84	1	1	1
20	0.7	1	1	1	1	60	0.8	0.84	1	1	1
20	0.8	1	1	1	1	60	0.9	0.81	1	1	1
20	0.9	1	1	1	1	60	1	0.85	1	1	1
20	1	1	1	1	1	70	0	0	0	0	0.01
30	0	0	0	0	0.10	70	0.1	0	0	0	0.02
30	0.1	0.16	0.16	0.14	0.21	70	0.2	0.09	0.02	0.02	0.03
30	0.2	0.82	0.82	0.83	0.82	70	0.3	0.18	0.28	0.28	0.30
30	0.3	1	1	1	1	70	0.4	0.47	0.74	0.79	0.77
30	0.4	1	1	1	1	70	0.5	0.59	0.93	0.97	0.97
30	0.5	1	1	1	1	70	0.6	0.62	0.95	1	0.99
30	0.6	1	1	1	1	70	0.7	0.65	0.99	1	1
30	0.7	1	1	1	1	70	0.8	0.61	0.97	1	0.99
30	0.8	1	1	1	1	70	0.9	0.64	0.99	1	0.99
30	0.9	1	1	1	1	70	1	0.64	0.99	1	1
30	1	1	1	1	1	80	0	0	0	0	0.01
40	0	0	0	0	0.04	80	0.1	0	0	0	0.02
40	0.1	0.04	0.04	0.05	0.07	80	0.2	0	0	0	0.02
40	0.2	0.58	0.59	0.57	0.6	80	0.3	0.04	0.07	0.09	0.11
40	0.3	0.97	0.98	0.98	0.98	80	0.4	0.19	0.39	0.51	0.46
40	0.4	0.99	1	1	1	80	0.5	0.36	0.73	0.82	0.82
40	0.5	1	1	1	1	80	0.6	0.41	0.87	0.97	0.95
40	0.6	1	1	1	1	80	0.7	0.37	0.85	0.99	0.93
40	0.7	0.98	1	1	1	80	0.8	0.38	0.93	0.99	0.96
40	0.8	1	1	1	1	80	0.9	0.4	0.92	1	0.95
40	0.9	0.99	1	1	1	80	1	0.42	0.95	1	0.99
40	1	0.99	1	1	1	90	0	0	0	0	0.01
50	0	0	0	0	0.02	90	0.1	0	0	0	0.02

Table 5. (Contd.)

P	*cue*	100%	99%	95%	$r < 16$	P	*cue*	100%	99%	95%	$r < 16$
90	0.2	0	0	0	0.01	110	0.1	0	0	0	0.01
90	0.3	0.01	0.01	0.03	0.03	110	0.2	0	0	0	0.01
90	0.4	0.05	0.14	0.18	0.2	110	0.3	0	0	0	0.01
90	0.5	0.14	0.38	0.59	0.55	110	0.4	0	0	0	0.03
90	0.6	0.17	0.59	0.87	0.78	110	0.5	0.01	0.03	0.07	0.08
90	0.7	0.16	0.64	0.93	0.82	110	0.6	0.01	0.07	0.26	0.21
90	0.8	0.19	0.72	0.95	0.87	110	0.7	0.02	0.16	0.46	0.36
90	0.9	0.19	0.78	0.97	0.88	110	0.8	0.02	0.23	0.59	0.49
90	1	0.18	0.78	0.97	0.94	110	0.9	0.03	0.31	0.65	0.58
100	0	0	0	0	0.01	110	1	0.03	0.35	0.68	0.6
100	0.1	0	0	0	0.01	120	0	0	0	0	0
100	0.2	0	0	0	0.01	120	0.1	0	0	0	0
100	0.3	0	0	0	0.01	120	0.2	0	0	0	0.01
100	0.4	0.01	0.02	0.04	0.07	120	0.3	0	0	0	0.01
100	0.5	0.03	0.12	0.27	0.24	120	0.4	0	0	0	0.01
100	0.6	0.05	0.27	0.59	0.49	120	0.5	0	0	0.01	0.02
100	0.7	0.08	0.39	0.76	0.63	120	0.6	0	0.02	0.08	0.06
100	0.8	0.06	0.44	0.83	0.69	120	0.7	0	0.03	0.19	0.12
100	0.9	0.08	0.54	0.85	0.72	120	0.8	0	0.07	0.33	0.24
100	1	0.1	0.62	0.9	0.84	120	0.9	0	0.12	0.40	0.31
110	0	0	0	0	0.01	120	1	0.1	0.20	0.48	0.41

X_{fam}—matrix of size $N \times P$, with P columns being randomly formed images presented first (step $t = 0$) for training and then (step $t = 1$) for recognition;

X_{nov}—matrix of size $N \times Q$, with Q columns being random images presented (step $t = 1$) only for recognition;

W—matrix of size $N \times N$ calculated (step $t = 0$) with the Hebb rule (2) on P images of X_{fam};

Y_{fam}—matrix of size $N \times P$, with P columns being images obtained (step $t = 1$) from the P images of X_{fam} by transformation according to (1);

Y_{nov}—matrix of size $N \times P$, with Q columns being images obtained (step $t = 1$) from the Q images of X_{nov} by transformation according to (1).

Program 3 additionally uses:

a—number of familiar images correctly classed as such;

b—number of familiar images wrongly classed as novel;

c—number of novel images wrongly classed as familiar;

d—number of novel images correctly classed as such;

error—fraction of wrong determinations (sum of *b* and *c* related to the total number of presented images);

cue—fraction of correct information in presentation of distorted familiar images given by X_{famerr};

X_{famerr}—matrix of size $N \times Q$, where $N \times P \times cue$ randomly chosen elements coincide with those of X_{fam} and the rest are generated at random;

K—number of times the recognition series is repeated (should exceed unity at small P for a more reliable estimate of *error*).

Program 1. Familiarity $E^*(t)$ is calculated at step $t = 1$ for each of the $P + Q$ familiar and novel images presented (X_{fam} and X_{nov}) at $N = 700$ and $P = 500$, 1000, 1500, …, 12000 (Q taken equal to P).

```
   N=700;
   i=0;
   for  P=500:500:12000
   i=i+1;
   Xfam=sign(rand(N,P)-.5);
   Xnov=sign(rand(N,P)-.5);
   W=Xfam*Xfam'-P*eye(N);
   Yfam=sign(W*Xfam);
   Ynov=sign(W*Xnov);
   for  p=1:P
   Efam(p) =Xfam(:,p)'*Yfam(:,p);
Enov(p)=Xnov(:,p)'*Ynov(:,p);
   end;
```

```
PP(i)=P;
Mf(i)=mean(Efam);
Sf(i)=std(Efam);
Mn(i)=mean(Enov);
Sn(i)=std(Enov);
end;
plot(PP,Mf,PP,Mf-2.33*Sf,PP,Mn,
PP,Mn+2.33*Sn);
```

Program 2. For 15 values $N = 100, 150, 200, \ldots, 800$, bipartitioning is used to find the corresponding $P_{\max}(N)$ as solutions of $M_f - 2.33S_f = M_n + 2.33S_n$.

```
n=0;
for N=100:50:800
n=n+1;
Pmin=N;
Pmax=round(N^2/30);
P=round((Pmin+Pmax)/2);
while Pmax-Pmin>2
P=round((Pmin+Pmax)/2);
  Xfam=sign(rand(N,P)-.5);
  Xnov=sign(rand(N,P)-.5);
  W=Xfam*Xfam'-P*eye(N);
  Yfam=sign(W*Xfam);
  Ynov=sign(W*Xnov);
for p=1:P
Efam(p)=Xfam(:,p)'*Yfam(:,p);
Enov(p)=Xnov(:,p)'*Ynov(:,p);
end;
  Mf=mean(Efam);
  Mn=mean(Enov);
f=Mf-2.33*std(Efam)-Mn-2.33*
std(Enov);
   if f>0  Pmin=P;  else  Pmax=P;
end;
end;
NN(n)=N;  PP(n)=P;
end;
format short g;
disp([NN',PP'];
```

Program 3 assesses the robustness of image recognition.

```
N=700;  P=100;  K=10;  s99=80;
for cue=0:0.01:0.2;
a=0;  b=0;  c=0;  d=0;
for k=1:K
Xfam=sign(rand(N,P)-.5);
Xfamerr=Xfam;
Xnov=sign(rand(N,P)-.5);
for i=1:N
for j=1:P
if rand > cue
Xfamerr(i,j)=sign(rand-.5);
end;
end;
end;
W=Xfam*Xfam'-P*eye(N);
Yfamerr=sign(W*Xfamerr);
Ynov=sign(W*Xnov);
for p=1:P
Efamerr(p)=Xfam(:,p)'*Yfamerr(:,p);
Enov(p)=Xnov(:,p)'*Ynov(:,p);
if Efamerr(p) > s99 a=a+1; end;
if Efamerr(p) <= s99 b=b+1; end;
if Enov(p) > s99 c=c+1; end;
if Enov(p) <= s99 d=d+1; end;
end;
end;  %  end of k-cycle
format short g;
error=(b+c)/(2*K*P);
disp([a b c d error cue]);
end;  %  end of cue-cycle
```

REFERENCES

1. E. V. Budilova, M. P. Karpenko, L. M. Kachalova, and A. T. Terekhin, Biofizika **54** (2), 287 (2009).
2. J. P. Aggleton and M. W. Brown, Behav. Brain Sci. **22**, 425 (1999).
3. K. A. Norman and R. C. O'Reilly, Psychol. Rev. **110**, 611 (2003).
4. M. W. Brown and J. P. Aggleton, Nat. Rev. Neurosci. **2**, 51 (2001).
5. R. Bogacz and M. W. Brown, Hippocampus **13**, 494 (2003).
6. J. Z. Xiang and M. W. Brown, Neuropharmacology **37**, 657 (1998).
7. J. J. Hopfield, Proc. Natl. Acad. Sci. USA **79**, 2554 (1982).
8. D. J. Amit, *Modeling Brain Functions* (Cambridge Univ. Press, Cambridge, 1989).
9. W. S. McCulloch and W. Pitts, Bull. Math. Biophys. **5**, 115 (1943).
10. D. O. Hebb, *The Organization of Behavior* (Wiley, New York, 1949).
11. R. Bogacz, M. W. Brown, and C. Giraud-Carrier, J. Comput. Neurosci. **10**, 5 (2001).