Recognition of topological features of graphs and images in neural networks

J. Phys. A: Math. Gen. 21 (1988) L813-L818. Printed in the UK

LETTER TO THE EDITOR

Recognition of topological features of graphs and images in neural networks

R Kree† and A Zippelius‡

† Institut für Theoretische Physik, Universität Düsseldorf, D-4000 Düsseldorf, Federal Republic of Germany

‡ Institut für Festkörperforschung der Kernforschungsanlage Jülich, Postfach 1913, D-5170 Jülich, Federal Republic of Germany

Received 19 April 1988

Abstract. We extend the architecture of the Hopfield network, such that it can recognise transformed versions of a set of learnt prototypes. As an example we construct a network which can generalise over all topologically equivalent representations of graphs or images. The construction is based on two coupled networks: a Hopfield network to store and retrieve patterns and a preprocessor to transform the input data.

It has been shown that simple models of neutral networks operate as efficient associative memories [1]. Therefore the question arises: whether these models also display some of the enormous computational abilities of natural neural networks in the brain of higher vertebrates. The error corrections during associative information retrieval involve only simple computations. In fact, a conventional computer needs a time of at most $O(N^2)$ to restore an incomplete or noisy input pattern with N bits in a memory with N stored patterns [2]. On the other hand, many of the higher cognitive functions must involve intrinsically hard computational problems.

As an example we discuss here a neural network which is designed to recognise the topological features of images. After the network has learnt p prototype graphs, it will be able to recognise an image which is topologically the same as one of the learnt graphs. To perform this task the network generates isomorphisms of the input graph and simultaneously tries to retrieve one of the learnt graphs. We first discuss how a network can perform either of these two tasks—graph matching and graph retrieval—seperately. We then go on and couple these two networks—one for each task—such that graph recognition is achieved by the combined system.

An image is represented on a model retina (or input layer) R with sites numbered $i = 1, 2, \ldots, N$. In the context of pattern recognition, the image may be any preprocessed version of the original picture. The topology of R defines the class of subgraphs under consideration. Two cases will be of particular interest in the following: a two-dimensional lattice and an infinite-dimensional lattice, i.e. a set of N fully connected points. The elementary pixel of the image consists of a black or white line segment which joins two sites. In this way every image becomes a subgraph of the model retina and will be represented by its $N \times N$ adjacency matrix G [3]. The element $G_{ij} = 1$ if i and j are neighbours (i.e. joined by a black link) and $G_{ij} = 0$ otherwise. To compare the topological features of two graphs G_1 and G_2 we map nodes of G_1 into those of G_2 by a homeomorphism, i.e. a map which conserves the neighbourhood relationships between nodes [3]. Two images are considered equal if their graphs are isomorphic, i.e. all their nodes can be mapped homeomorphically.

0305-4470/88/160813+06$02.50

The Euclidean distance d of the adjacency matrices of two graphs G_1 and G_2 is defined as

$$d^2(G_1, G_2) = \tfrac{1}{2}\,\mathrm{Tr}(G_1 - G_2)^2 = N(G_1) + N(G_2) - \omega(G_1, G_2)$$

where $N(G_1)$ is the total number of links in the graph G_1 and $\omega(G_1, G_2) = \mathrm{Tr}(G_1 G_2)$ denotes the overlap of the two graphs G_1 and G_2. The distance as well as the overlap depend on the labelling of the graphs on the lattice. For a lattice of N points there are $N!$ different labellings $\pi(i)$ of the lattice points $i = 1, 2, \ldots, N$. These transformations T can be represented by an $N \times N$ matrix $t_{i,\alpha=\pi(i)}$ with one element 1 in each row and each column. To compare the topology of two graphs we either use the maximum of the overlap $\Omega = \max_T \omega_T$, $\omega_T = \mathrm{Tr}(G_1, T^+ G_2 T)$ or the minimum of the distance $D^2 = \min_T d^2(G_1, T^+ G_2 T)$ taken over all lattice transformations T. Ω equals twice the maximal number of links, which can be mapped homeomorphically. G_2 is a subgraph of G_1 if $\Omega(G_1, G_2) = 2N(G_2) \leqslant 2N(G_1)$ and $D = 0$ if two graphs are isomorphic.

The task of subgraph matching is defined as follows. Find the transformation T, which either minimises the distance or maximises the overlap of two graphs G_1 and G_2. Subgraph matching is known to be a hard computational problem [4]. One possible approach to such problems is simulated annealing [5], which attempts to find near-optimal solutions. In this procedure one introduces a noise level λ^{-1}. The system evolves stochastically in the space of transformations T, such that it relaxes to a stationary distribution $\exp(\lambda \omega_T)$. The quantity of interest is the ground-state configuration.

The task of subgraph matching can also be formulated for a neural network [2, 6]. One identifies the elements of the transformation T with N^2 neurons $t_{i\alpha} \in \{0, 1\}$ whose activity is restricted by $\Sigma_i\, t_{i\alpha} = 1$ and $\Sigma_\alpha\, t_{i\alpha} = 1$. These constraints can be enforced by appropriate penalties, which have to be added to the cost function

$$H_p\{T\} = \mu \sum_i \left(1 - \sum_\alpha t_{i\alpha}\right)^2 + \mu \sum_\alpha \left(1 - \sum_i t_{i\alpha}\right)^2. \tag{1}$$

Hence the total cost function for subgraph matching [2, 6] is $H = -\omega(T^+ G_1 T, G_2) + H_p$. The synaptic couplings are determined by the input graphs G_1 and G_2.

We want to construct a Hopfield network, which can store and retrieve graphs, which are subgraphs of a set of N fully connected points. We associate a neuron $S_{\alpha\beta} \in \{0, 1\}$ to every *link* between two points α and β ($\alpha, \beta = 1, 2, \ldots, N$). The $N(N-1)/2$ neurons interact with the Hamiltonian

$$H_M = -J_0 \sum_{(\alpha\beta) \neq (\gamma\delta)} S_{\alpha\beta} J_{\alpha\beta\gamma\delta} S_{\gamma\delta} + \sum_{(\alpha\beta)} h_{\alpha\beta} S_{\alpha\beta}. \tag{2}$$

Here the summation over $(\alpha\beta)$ runs over distinct links and $h_{\alpha\beta}$ denotes the threshold field of neuron $S_{\alpha\beta}$. The synaptic couplings $J_{\alpha\beta\gamma\delta}$ are determined by the graphs to be learnt, for example

$$J_{\alpha\beta\gamma\delta} = \sum_{\nu,\mu=1} \xi^\nu_{\alpha\beta} \xi^\mu_{\gamma\delta} (C^{-1})_{\nu\mu}$$

with $\xi^\nu_{\alpha\beta} = 2G^\nu_{\alpha\beta} - 1 \in \{-1, -1\}$ and

$$C_{\nu\mu} = \frac{2}{N(N-1)} \sum_{(\alpha\beta)} \xi^\nu_{\alpha\beta} \xi^\mu_{\alpha\beta}$$

[7]. The network can store and retrieve $p = N(N-1)/2$ graphs.

In the problem of pattern recognition one may be interested in an ensemble of graphs, which are subgraphs of a two-dimensional lattice and hence are sparse. The mean activity of such a graph becomes extremely low for large graphs: $\Sigma_{(\alpha\beta)} G^{\nu}_{\alpha\beta} = O(N)$. It may then be useful to constrain the activity of the neurons accordingly [8] and make use of the threshold fields $h_{\alpha\beta}$ for storing purposes [9]. The coupling constant J_0 should be scaled, such that the cost function is proportional to the number of active neurons, $J_0 = 1/2N_a$ ($N_a = N(N-1)/2$ for an infinite-dimensional lattice and $N_a = Nz/2$ for a two-dimensional lattice with coordination number z). The global minima of this network are the learnt patterns [10].

The task is to construct a network, which learns p prototype graphs and is able to perform one of the following functions.

(i) Recognise isomorphic graphs. The network is presented with an image, which is not equal to any of the learnt ones, but a displaced, rotated and/or homeomorphically distorted version of a learnt one. The network shall retrieve the learnt graph, which is isomorphic to the presented image.

(ii) Recognise subgraphs. The presented image is an incomplete version of a learnt graph and may be additionaly distorted as above. The network shall recognise that learnt graph which contains the presented image as subgraph.

These tasks can be performed by three coupled networks: a receptor layer R, a memory layer M and a preprocessor P. The layer R is the input layer described above. An image is represented on R by the state of $N(N-1)/2$ neurons $G_{ij} \in \{0, 1\}$ which are not updated during the recognition process. The memory layer is a fully connected Hopfield network, defined by (2). The preprocessor consists of N^2 neurons $t_{i\alpha} \in \{0, 1\}$, which realise the transformation T between the lattice points of the receptor R and the memory layer M (see figure 1).

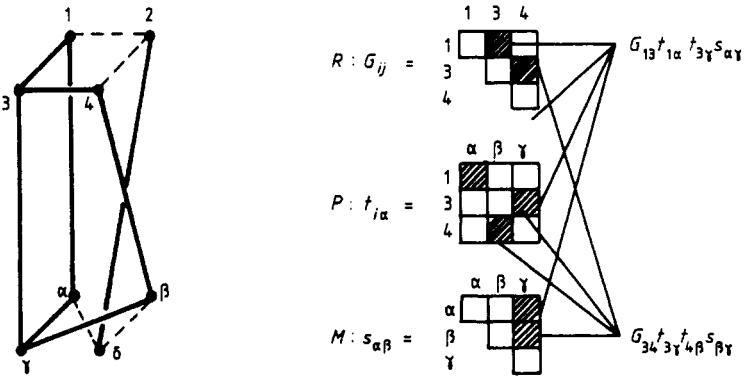


Figure 1. Network with receptor or input layer R, preprocessor P and memory M. Shown on the right are the neural activity patterns and the hardwiring.

The graphs to be learnt $\{G^{\nu}_{ij}\}$ are defined on the receptor layer R. The preprocessor transforms these into neural activity patterns $\xi^{\nu}_{\alpha\beta} = \Sigma_{i,j}\, t^{\nu}_{i\alpha} t^{\nu}_{j\beta}(2G^{\nu}_{ij} - 1)$ in the memory layer. These are stored in the synapses of M with an appropriate learning rule. The P layer has no special 'learning state'. Its configuration $\{t_{i\alpha}\}$ can be different for every learnt pattern if one wants to use the full storage capacity of the memory.

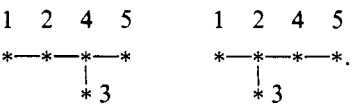
After the learning stage has been completed, an image G is presented on R and the network has to recognise it either as isomorphic to or as a subgraph of one of the learnt graphs. This can be achieved in a cooperation of the two networks: the preprocessor P for graph matching and the memory M for graph retrieval. Graph recognition is equivalent to finding the absolute minima of the total cost function

$$H = \lambda \left(\tfrac{1}{2} \sum_{\alpha,\beta} \left(\sum_{i,j} t_{i\alpha} t_{j\beta} G_{ij} - S_{\alpha\beta} \right)^2 + H_p\{T\} \right) + H_M\{S\} \tag{3}$$

for isomorphic graph matching. For subgraph matching the first term of H has to be replaced by $-\omega(T^+GT, S)$. The search for the global minima has to be done in the space of all $\{S_{\alpha\beta}, t_{i,\alpha}\}$, while the neurons of the input layer R are taken to be quenched in the configuration $\{G_{ij}\}$. The global minima of H_M are the learnt graphs. For isomorphic graph matching H is minimised if the distance between the input graph and a stored graph vanishes. Hence the global minima of H are $\Sigma_{i,j}\, t_{i\alpha} t_{j\beta} G_{ij} = S_{\alpha\beta} = (\xi^{\nu}_{\alpha\beta} + 1)/2$. For subgraph matching H is minimised if all links of the input graph G can be mapped homeomorphically onto one of the stored graphs.

The Hamiltonian for graph recognition contains interactions of up to four neurons (see figure 1). For technical realisations, this has the advantage that the hardwiring of the network is independent of the learnt and presented graphs [2].

The dynamic evolution of the network should be such that the global minima of H are stationary states in the absence of noise. The structures of metastable states and their barriers depend on the details of the dynamics. For example, if we choose the transposition of two lattice points as an elementary change of T, then one can easily convince oneself that metastable states with high barriers are present. A simple example is given below:



For the efficiency of the procedure it is important to exclude moves, which cannot change the cost function, i.e. transpositions of lattice points, which do not involve at least one node of G. To allow the system to escape from a metastable state one can either introduce noise [5] or relax the constraints to smooth the energy landscape [2]. Here we consider the first alternative and keep the constraints hard ($\mu \gg 1$). A possible dynamics consists of a four spin-flip of the neurons $\{t_{i\alpha}\}$ of P and a two spin-flip of the neurons $\{S_{\alpha\beta}\}$ of the memory M. In this way the activity of the networks can be dynamically constrained. If the flip rates are chosen according to detailed balance, then the system relaxes to the stationary distribution $\exp(-\beta H)$, where β^{-1} is the noise level. The question for a dynamics which leads to a satisfactory performance of the network cannot be answered in general. We expect it to be dependent on the ensemble of graphs under consideration. The tuning of flip rates, noise level and constraint parameters μ and λ has to be investigated numerically in the context of practical applications.

Nevertheless it is instructive to consider the limiting case of infinite λ for which the cost function takes on the form

$$H = -\frac{1}{2N_a} \sum_{(\alpha\beta)\neq(\gamma\delta)} \sigma_{\alpha\beta}\sigma_{\gamma\delta} J_{\alpha\beta\gamma\delta} + \sum_{(\alpha\beta)} h_{\alpha\beta}\sigma_{\alpha\beta} \tag{4}$$

with $\sigma_{\alpha\beta} = \Sigma_{i,j}\, t_{i\alpha} t_{j\beta} G_{ij}$. Equation (4) looks like a Hopfield Hamiltonian, but note that the configurations $\{\alpha_{\alpha\beta}\}$ are restricted to isomorphisms of the input graph G. To assess the effect of the noise we estimate the entropy and the energy for large graphs. The total number of transformations T is $N!$. Hence they contribute a term of $O(N \ln N)$ to the entropy. The total cost function is extensive, i.e. proportional to the number of links in the input graph. If the input layer R is a two-dimensional lattice, then the system will always be in the high-temperature phase, provided the noise level remains finite in the thermodynamic limit (a similar problem occurs in other optimisation tasks, see, e.g., [11]). To *balance* energy and entropy one has to choose a noise level $\beta = \tilde{\beta} \ln N$ with $\tilde{\beta} = O(1)$ as N goes to infinity. If the input layer is an infinite-dimensional lattice, the system will always be in the low-temperature phase unless the noise level is scaled as $\beta = (\tilde{\beta} \ln N)/N$ with $\tilde{B}$ again of order one. With the appropriate scaling we expect that the behaviour of the network with low noise level ($\tilde{\beta} > 1$) is qualitatively the same as without noise. In the low-temperature phase the overlap of the transformed input graph T^+GT with one of the learnt graphs G^μ scales with the number of active neurons:

$$\langle \omega \rangle_{\text{th}} = \sum_{\alpha\beta} G^\mu_{\alpha\beta} \sum_{ij} \langle t_{i\alpha} t_{j\beta} \rangle_{\text{th}} G_{ij} = O(N_a).$$

This implies that the order parameter for graph recognition is non-zero, i.e. the input graph has been recognised as one of the stored graphs. (Here $\langle \ldots \rangle_{\text{th}}$ denotes the average with the distribution $\sim \exp(-\beta H)$.)

In the case of a fully connected input layer the order parameter

$$m_\nu = \frac{1}{N_a} \sum_\mu C^{-1}_{\nu\mu} \sum_{(\alpha\beta)} \xi^\mu_{\alpha\beta} \langle 2\sigma_{\alpha\beta} - 1 \rangle_{\text{th}}$$

is the extremum of the free energy

$$f = \tfrac{1}{2} \sum_{\nu\mu} m_\nu C_{\nu\mu} m_\mu - \frac{1}{N_a \beta} \sum_{(\alpha\beta)} \ln \cosh(\beta \boldsymbol{m} \cdot \boldsymbol{\xi}_{\alpha\beta})$$

$$- \frac{1}{N_a \beta} \ln \operatorname{Tr}_{\{t_{i\alpha}\}} \prod_{(\alpha\beta)} [1 + (2\sigma_{\alpha\beta} - 1) \tanh(\beta \boldsymbol{m} \cdot \boldsymbol{\xi}_{\alpha\beta})]. \tag{5}$$

The threshold field has been chosen such that $h_{\alpha\beta} = \Sigma_{(\gamma\delta)} J_{\alpha\beta\gamma\delta}$. The first two terms are those of a standard Hopfield model with a transition temperature $\beta_c^{-1} = 1$. The constraint on $\sigma_{\alpha\beta}$ destroys the global inversion symmetry of H so that the expansion of f contains even as well as odd powers of m. Hence the transition is expected to be discontinuous.

For the purpose of illustration we consider a small system, where f can be calculated analytically. As an example we consider an input layer of four fully connected points. Suppose that the network has learnt three topologically different graphs and we use an isomorphism of one of them as input (see figure 2). The free energies f_ν for the pure states $m_\nu = m\delta_{\nu n}$ ($n = 1, 2, 3$) are shown in figure 2 for $\beta = 0.81$. For small β, $m = 0$ is the globally stable state and for $\beta \simeq 0.8$ there is a discontinuous transition to a phase, where isomorphic graphs are recognised. Note that the graph represented by $\{-\xi^1_{\alpha\beta}\}$ is isomorphic to G_2, whereas $\{-\xi^3_{\alpha\beta}\}$ is isomorphic to G_3. This implies $f_2(m) = f_1(-m)$ and $f_3(m) = f_3(-m)$.

Recently Bienenstock and von der Malsburg [6] have also suggested a network for invariant pattern recognition. They represent the image by a distribution of features on the nodes of a graph with fixed topology. We disagree with their Hamiltonian for graph retrieval because it possesses a large number of additional global minima which are mixtures of the learnt patterns. A further basic difference to our approach lies in

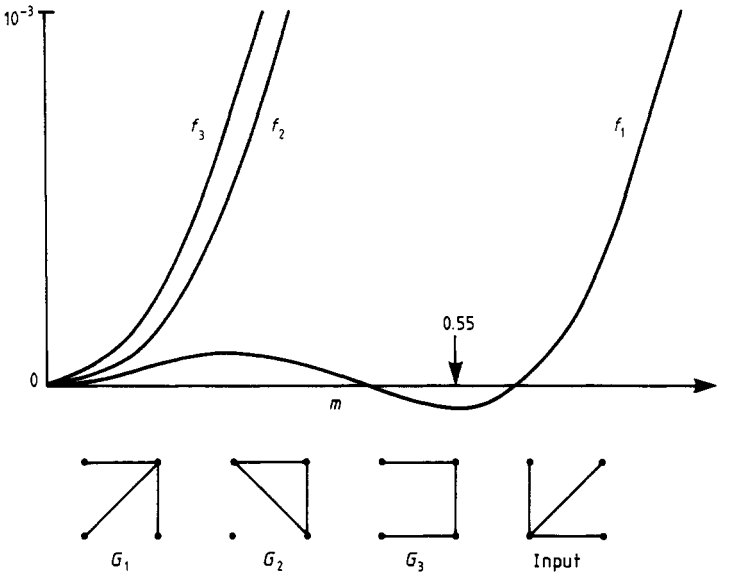


Figure 2. Free energies for $\beta = 0.81$. The learnt graphs (G_1, G_2, G_3) and the input graph are shown on the bottom.

the interpretation of the dynamic variables. Whereas we represent a graph as an activity configuration of neurons, their dynamic variables are interpreted as synapses. Their learning rule consists in local dynamic constraints on the synaptic activity which are not represented in the Hamiltonian.

In the problem of subgraph recognition, we have focused on topological features of images. In practical applications one often is also interested in a smaller group of transformations than homeomorphisms, for example translations and rotations. A two-dimensional retina does not only define neighbourhod relations but also metrical properties, like distances and angles. These can be mapped onto neural activity patterns if the transformations T—realised in the preprocessor P—are restricted to rotations and translations, too. Note that this does not require a change of the architecture of the network.

References

[1] Hopfield J J 1982 *Proc. Natl Acad. Sci. USA* **79** 2554; 1984 *Proc. Natl Acad. Sci. USA* **81** 3088
[2] Hopfield J J and Tank D W 1986 *Disordered Systems and Biological Organisation* ed E Bienenstock *et al* (Berlin: Springer)
[3] Harary F (ed) 1967 *Graph Theory and Theoretical Physics* (New York: Academic)
[4] Garey M R and Johnson D S 1979 *Computers and Intractability* (San Francisco: Freeman)
[5] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 *Science* **220** 671
[6] von der Malsburg C and Bienenstock E 1987 *Europhys. Lett.* **3** 1243
Bienenstock E and von der Malsburg C 1987 *Europhys. Lett.* **4** 121
[7] Kohonen T 1986 *Associative Memory* (Berlin: Springer)
Personnaz L, Guyon I and Dreyfus G 1985 *J. Physique Lett.* **46** L359
[8] Amit D J, Gutfreund H and Sompolinsky H 1987 *Phys. Rev.* A **35** 2293
[9] Tsodyks M D and Feigel'man M V 1988 *Europhys. Lett.* **6** 101
[10] Amit D J, Gutfreund H and Sompolinsky H 1985 *Phys. Rev.* A **32** 1007
Kanter I and Sompolinsky H 1987 *Phys. Rev.* A **35** 380
[11] Vannimenus J and Mézard M 1984 *J. Physique Lett.* **45** L1145