

Symmetry processing in neural network models

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1989 J. Phys. A: Math. Gen. 22 3379

(<http://iopscience.iop.org/0305-4470/22/16/027>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 129.252.86.83

The article was downloaded on 31/05/2010 at 11:45

Please note that [terms and conditions apply](#).

Symmetry processing in neural network models

Javier Mas[†] and Eduardo Ramos[‡]

[†] Departamento de Física Teórica, C-XI, Universidad Autónoma de Madrid, 28049 Madrid, Spain

[‡] Institute for Theoretical Physics, State University of New York at Stony Brook, Stony Brook, New York 11794-3840, USA

Received 1 June 1988, in final form 11 April 1989

Abstract. We propose a simple way to preprocess symmetries based on the construction of invariant functions over the configuration space of the neural network.

1. Introduction

Some years ago Hopfield [1] proposed a very simple model for pattern recognition that despite its simplicity possesses one of the remarkable properties addressed by the human brain, its flexibility. This network is able to recognise patterns that are similar but not identical to the ones stored. This is accomplished by storing the memories as fixed points of a certain dynamics, for example in the Hopfield model it is a spin glass with a 'Monte Carlo' time evolution. If the input is close to one of these fixed points that act like attractors in phase space, the time evolution will eventually drive it there.

Due to this fact, the introduction of the concept of a 'distance' seems unavoidable. We need it in order to measure the basins of attraction of the fixed points or, in simple words, to estimate the amount of error beyond which the network will not recognise a given input. It is believed that these basins of attraction are very complex and may even have a fractal structure. Nevertheless it is known that a convenient measure of the distance between two different states is given by the Hamming distance which counts the number of different entries in the two states to be compared.

In our opinion, one of the serious drawbacks of this distance is the fact that it 'looks' at the vectors entry by entry. If we take a vector and perform a transformation that keeps its structure, the Hamming distance will in general be very large. It is unlikely that a network, where such a notion of distance is a natural one, would recognise it.

Hence, if we want a truly flexible network, one that recognises structures by identifying vectors that are related to each other by a 'symmetry transformation', we are driven to introduce a new concept of distance that will be invariant under such transformations.

The aim of this paper is to study how symmetries can be introduced in the configuration space of the neural network, in such a way that patterns related by a symmetry transformation to a memory can be recognised as such. For this reason, we introduce a preprocessing mechanism which can be regarded as a new distance among equivalence classes under the symmetry group, and is implemented by the construction of a particular kind of invariant functions (which may not be the most efficient ones). For different approaches to this problem we refer the interested reader to [2].

The plan of the paper is as follows: in § 2 we recall certain basic features about neural networks and the Hopfield model in particular, and we introduce the concept of an invariant distance for a particular choice of symmetry group. We choose to start with the group of cyclic permutations (or shifts with periodic boundary conditions) because of its interest and simplicity, but afterwards we give a prescription to generalise this construction for any subgroup of the whole permutation group. Hence the methods developed here are not restricted to the case of shifted, or translated, patterns.

In § 3 we study how errors, which are introduced at random, propagate via the preprocessing algorithm. We use a random walk approximation that nicely fits the 'experimental data' coming from several computer runs. We will see that errors are magnified, implying the reduction of the basins of attraction. This is the price we have to pay for greater flexibility and a larger number of recognised patterns.

In § 4 we analyse the more interesting question of the robustness of the system with respect to errors that are not random, but involve the permutation of blocks of entries, and we find (not surprisingly) that the system is more resistant to the latter, due to the fact that the distance we are using is 'not local' and only cares about the structure, which is partially preserved by non-random errors.

In § 5 we briefly discuss the viability of such a system as a model of a real biological one, and we argue that the kind of connections used for the preprocessing could in principle be found in a nervous system.

Finally, we conclude by recapitulating the results and speculating about the possible construction of a network, based on principles similar to the ones used to process the symmetries.

2. The invariant distance

In what follows we are going to recall some general features about neural networks [2] that will be used later. The network is a set of N 'neurons' that can be regarded as Ising spins, so they can only be in two states, which we have chosen to be 1 and 0 corresponding to the firing or not-firing state in biological terms. Let us define the configuration space of the network Γ_N as the set of all possible states of the system with N neurons

$$\Gamma_N = \{S \mid S \in \mathbb{Z}_2^N\} \quad (2.1)$$

where \mathbb{Z}_2 is the Abelian group of integers mod 2.

The next step is to define a dynamics in Γ_N in such a way that there are certain stable states ξ^α called memories. From now on we are going to assume for simplicity that these memory vectors are uncorrelated, and with entries that are statistically independent from each other (taking values 0 or 1 with probability $\frac{1}{2}$). Nevertheless, the methods developed here are not *a priori* restricted to that case.

The usual procedure is to define an energy functional that is a monotonically decreasing function of a 'time' defined through a Monte Carlo dynamics. Storing memory states is accomplished by constructing an energy functional with those states as minima, so if the input is 'near enough' to one of the memories, the time evolution will drive it to the corresponding minimum. Therefore such an input will be recognised as the stored memory even if it is not exactly identical to it.

We should ask what is meant by 'near enough'. Perhaps the best answer to this question is to reverse the argument above and say that two states are 'near enough'

if, under the influence of the chosen dynamics, they will eventually arrive at the same minimum. Of course, this will require exact knowledge of the time evolution of the system and for the moment it is far from our analytical and computational abilities to provide such an answer.

Nevertheless, it is known that the Hamming distance

$$\text{Ham}(\xi^\alpha, \xi^\beta) = \sum_{i=1}^N (\xi_i^\alpha + \xi_i^\beta) \bmod 2 \tag{2.2}$$

is a good measure of the distance in the known examples.

Now suppose (as is often the case) that we would like to recognise not just single patterns but whole equivalence classes under a certain symmetry group G . In general, the Hamming distance between any two elements of any given equivalence class will be very large. It is unlikely that if we store a certain pattern the network will recognise one that is symmetry related to it.

To keep things as simple as possible at the beginning, we will first fix our attention on the group of cyclic permutations C_P (or equivalently shifts with periodic boundary conditions). We think that this symmetry group is specially interesting, because it preserves the global ‘one-dimensional structure’ of the state. We will describe how to generalise our construction to other symmetry groups at the end of this section.

C_P is generated by just one element that can be taken to be the translation by a single step to the right, so $g^N = 1$ if we assume periodic boundary conditions. All the elements of the group are of the form g^r where $r \in \mathbb{Z}_{N-1}$.

If we compute the Hamming distance between a memory and its symmetry-related pattern, we obtain on average

$$\langle \text{Ham}(\xi^\alpha, g^r \cdot \xi^\alpha) \rangle = N/2 \quad \forall r \neq 0 \tag{2.3}$$

which implies that $g^r \cdot \xi^\alpha$ is uncorrelated to the associated memory.

If we want the network to recognise this pattern, there are two possible ways to proceed: store all patterns that are related by a symmetry transformation, or preprocess the input in such a way that the Hamming distance between the symmetry related patterns will be zero or at least much smaller than $N/2$.

The first option would require multiplying the number of memories by the order of the symmetry group, which is N in the case of C_P . This is not possible in the known examples; for instance in the Hopfield model the maximum number of stored patterns is of the order αN , where $\alpha \sim 0.14$. In any case this brute force procedure is not very appealing, so we will concentrate on the second possibility.

Our approach is the following: we construct a map γ from the configuration space Γ_N into a new configuration space $\hat{\Gamma}$ where the dynamics will take place, with the following constraints.

(i) In the new configuration space $\hat{\Gamma}$ we should be able to implement the same kind of neural networks based on a Monte Carlo spin dynamics.

(ii) This map is going to induce a distance[†] in Γ_N with the following properties:

$$\gamma^* \text{Ham}(s^\alpha, s^\beta) = \text{Ham}(\gamma \cdot s^\alpha, \gamma \cdot s^\beta) \quad \gamma \cdot s \in \hat{\Gamma}, s \in \Gamma_N \tag{2.4}$$

$$\gamma^* \text{Ham}(s^\alpha, G \cdot s^\beta) = \gamma^* \text{Ham}(s^\alpha, s^\beta) \quad \forall s^\alpha, s^\beta \in \Gamma_N \tag{2.5}$$

[†] Here we are using the concept of distance in a loose sense. Strictly speaking we will only require it to be a symmetric map from $\Gamma_N \times \Gamma_N$ into the non-negative integers.

where by $G \cdot s$ we mean the action of any element of the symmetry group, for example the action of g' for cyclic permutations. These relations clearly imply

$$\gamma \circ G = \gamma \tag{2.6}$$

so γ should be an invariant map under the action of G .

(iii) Finally, we require the map γ to be easily implementable in hardware. We will elaborate on this point in § 5.

In order to produce the invariant functions for the cyclic permutation group obeying the conditions stated above, we have proceeded as follows.

First we generate the set of vectors

$$\Omega_i^\theta = (s_i + s_{i+\theta}) \bmod 2 \quad \theta = 1, \dots, N/2 \tag{2.7}$$

so that for any $s \in \Gamma_N$ we have $N/2$ vectors labelled by θ ($(N+1)/2$ if N is odd). The reason θ takes values no bigger than $N/2$ is that the set consisting of all $\Omega_i^{N/2+\theta}$ for a fixed θ contains the same elements as the one obtained by taking all the elements of the form Ω_i^θ ; this is due to the fact that we are taking periodic boundary conditions, so the sum $i + \theta$ in (2.7) has to be understood modulo N .

Define ω as follows:

$$\omega^\theta = \sum_{i=1}^N \Omega_i^\theta. \tag{2.8}$$

Now we can regard ω as a vector with components labelled by θ and entries which take values in the positive integers. What is the information that this vector ω contains? Clearly ω only keeps track of the relative distribution of zeros and ones in the original vector s . This is the precise meaning of the ‘one-dimensional structure’ we referred to before. This information is invariant under cyclic permutations. So we have mapped a vector s to another vector ω in a way that is ‘blind’ to the action of C_P .

It is obvious that this map is also invariant under the following symmetry transformations in Γ_N : (1) the dual transformation, which consists of interchanging all the zeros and ones; (2) inversions, by which we mean the map $s_i \rightarrow s_{(N+1)-i}$. These invariances can be easily broken, but they preserve the ‘one-dimensional structure’ of the vectors as the cyclic permutations do, so we will keep them as a ‘bonus’.

We can regard equations (2.7) and (2.8) as defining a map π from the equivalence classes of Γ_N under the symmetry group into a space $\Psi = \{\omega\}$

$$\begin{aligned} \pi : \Gamma_N / G &\rightarrow \Psi \\ [s] &\leftrightarrow \omega. \end{aligned} \tag{2.9}$$

Obviously π does not obey the first of our requirements, because it will not be possible to implement a spin Monte Carlo dynamics in Ψ . Hence, the next step is to define the map

$$\eta : \Psi \rightarrow \Gamma_{N/2} \tag{2.10}$$

where

$$\eta(\omega^\theta) = \Theta(\omega^\theta - N/2) = \hat{s}^\theta \tag{2.11}$$

and Θ is the usual truth function. $\Theta(x) = 1$ if $x > 0$ and zero otherwise.

Now we define $\gamma = \eta \circ \pi$

$$\gamma : \Gamma_N \rightarrow \Gamma_{N/2}. \tag{2.12}$$

This map γ obeys all the requirements because it can be implemented in hardware with very simple connections, as we will see in § 5. In this case we can identify the space $\hat{\Gamma}$ with $\Gamma_{N/2}$.

A further test is needed if the network is to recognise only uncorrelated patterns. We have to check that the images of uncorrelated memory vectors are also uncorrelated†. This implies that

$$\langle \gamma^* \text{Ham}(\xi^\alpha, \xi^\beta) \rangle = N/4. \quad (2.13)$$

We have been unable to find an analytical proof for (2.13). Nevertheless we have checked with several computer runs the validity of this statement.

Since we have reduced the dimension of the configuration space from 2^N to $2^{N/2}$, the maximum number of stored patterns is also reduced; for example in the Hopfield method that number is proportional to N , so it decreases by a factor of 2. But, associated with each stored pattern there are $4N$ symmetry-related ones which will be recognised as a memory, so the final result is an increase by a factor of $2N$.

One of the main drawbacks of the map γ is that it maps all the vectors with less than $N/4$ ones (or equivalently less than $N/4$ zeros, because of the invariance under the duality map) into the zero vector in $\Gamma_{N/2}$. In any case, if the entries of the memory vectors are random bits, the probability that more than one of the memories belongs to that subspace is extremely low.

Another possible criticism of this preprocessing method is that many different vectors are mapped to the same one (by different in this context we mean vectors not related by a symmetry transformation), due to the fact that the dimension of the image of Γ_N is much smaller than the dimension of Γ_N itself. We do not consider this a drawback because if two different vectors are mapped to the same one in $\Gamma_{N/2}$, it means that statistically their Euclidean distance is smaller than the average distance between any two vectors in Ψ , so in some sense they are closer because they have similar structure (remember that the vectors ω were a measure of the structure of the vector they come from). In fact, we will speculate later that this can be a very interesting property, because 'similar' vectors can be recognised as the same one, without the need of any dynamics in the system.

The generalisation to different symmetry groups (that are subgroups of the whole permutation group of N elements) comes from the observation that in the set Y of cardinality $N(N-1)/2$, defined as

$$Y = \{\Omega_i^\theta\} \quad (2.14)$$

the action of the symmetry group defines a partition. We will denote the j th orbit by O^j and define the vectors ω as

$$\omega^j = \sum_{\alpha \in O^j} \alpha. \quad (2.15)$$

So the prescription goes through by substituting for $N/2$ in the argument of the Θ function one half the number of the elements in the orbit. It is easy to check that this reduces to the results obtained before for the particular case of the group of cyclic

† Notice that this is not true if we restrict ourselves to the map π . The ω^θ are not statistically independent; a simple way to see this is to consider the case where ω^1 is equal to N ; in this particular case we can reconstruct completely the vector in Γ_N it comes from (it has for components 0s and 1s alternatively), so it determines univocally the values of the other ω .

permutations. In equation (2.8) the sum over i can be interpreted as the sum over all the elements of the orbit

$$g^r \cdot \Omega_i^\theta = \Omega_{i+r}^\theta. \quad (2.16)$$

Notice that for an arbitrary symmetry group, θ will stop being a good label for the whole orbit, in general elements with different theta will belong to the same orbit. In general for symmetry groups of order much higher than N , we will have to introduce in Y elements that will measure more than the 'two-point correlation', those of the form

$$\Omega_i^{\theta\nu\dots} = s_i + s_{i+\theta} + s_{i+\theta+\nu} + \dots \quad (2.17)$$

where the sums have to be understood as modulo two.

If not, the configuration space $\hat{\Gamma}$ would be, in general, 'reduced to ashes'.

One of the peculiarities of this preprocessing mechanism is that the states that are retrieved are not the memories themselves but their γ -images. Of course that would be no problem for a computer system, because the information should not necessarily be encoded in the very same way as the input that recalls it. But a far more interesting question arises if we ask ourselves if the human brain can work in such a fashion. It is not still known how information is stored in our brains, but there are some clues that information about form and spatial orientation are connected with great difficulty in early childhood [4]. This can suggest that certain invariant properties of the objects are stored independently of the symmetry transformations that can be performed on them. For a specially interesting experiment along these lines we refer to Gibson [5]. This experiment shows that children from four to eight years old have no difficulty in distinguishing patterns with different shapes, but they show great difficulty in distinguishing patterns related by some particular kinds of symmetry transformations.

3. Random walk approximation for propagation of random errors

As already discussed, we would like to use a suitable Monte Carlo dynamics in the space $\hat{\Gamma}$. The fixed points of this dynamics will be the image via γ of the patterns that we want the network to recognise. So it is a question of major relevance to find how errors propagate from Γ_N to $\hat{\Gamma}$ via the map γ .

We will concentrate on the case of shifts or cyclic permutation invariance (the generalisation to other symmetries is obvious). We will assume that the errors are produced at random and uncorrelated, and the memories are also random (though it should be kept in mind that the methods developed in the last section are not restricted to this situation). The more interesting case of errors produced by permutation of blocks of entries will be studied numerically in the following section.

In order to see how these random errors propagate, we are going to assume that the entries of a vector in $\Gamma_{N/2}$ are statistically independent, so we will first compute the probability P_δ for each of the entries to change from 1 to 0 or vice versa when we modify a fraction δ of the entries in the memory, and later to predict the total number of changes we will use a random walk with probability P_δ of taking a step to the right and probability $1 - P_\delta$ of not moving at all (corresponding respectively to modifying or not one of the entries), so the average number of changes as well as its standard deviation will be easily computed. We have checked the agreement of the theoretical computation with the 'experimental' results obtained from several runs on the computer (figure 1).

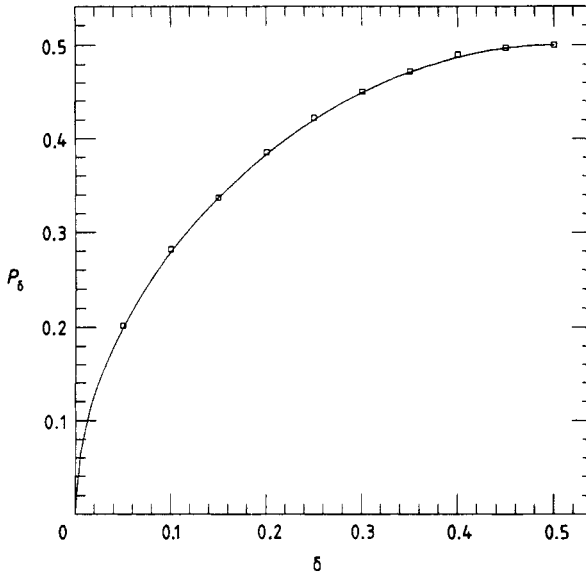


Figure 1. The fraction of modified entries in $\Gamma_{N/2}$ (P_δ) is plotted against the fraction of modified entries in Γ_N (δ). The full curve represents the theoretical prediction and the experimental points are represented by the squares. The error is approximately ± 0.05 .

The probability that an entry of \hat{s}^θ changes from one to zero is by its definition the probability for $\omega^\theta - N/2$ to change from positive to negative. For random memories, we will regard the map π as a random walk, where the probability for Ω_i^θ to be zero or one is $\frac{1}{2}$. This would be exact if all the Ω for a fixed θ were statistically independent, but this is not the case when θ and N are not relatively prime[†]. This can be clearly seen in an example with $N = 9$ and $\theta = 3$, if we know $s_1 + s_4$ and $s_4 + s_7$, that are respectively Ω_1^3 and Ω_4^3 , Ω_7^3 will just be $(\Omega_1^3 + \Omega_4^3) \bmod 2$. These correlations will be obviously not important when N/θ goes to infinity, but this will not *a priori* be true when N/θ remains finite in the limit N going to infinity. Nevertheless we will argue that this is indeed the case. In order to see that, we will fix our attention on the simple case where $\theta = N/3$ (here we are assuming N to be divisible by 3), in this case we will have a random walk with $N/3$ steps, but now we will have probability $\frac{3}{4}$ of moving two 'steps' to the right and probability $\frac{1}{4}$ of not moving at all; it is a simple exercise to show that in the limit N goes to infinity the probability distribution is a Gaussian with the same parameters that we would have obtain with N steps, with a probability of $\frac{1}{2}$ of giving a step to the right and $\frac{1}{2}$ of not moving at all. These results can be generalised, but perhaps the best and simplest justification for neglecting these correlations will be the results themselves.

In order to compute P_δ we proceed as follows.

(i) First, we compute the distribution of possible values for $\omega^\theta - N/2$ when it is the image via γ of one of the random memories (assuming statistical independence as explained above).

(ii) Next, we determine the average number of Ω of a specific level that are altered when we randomly change δN entries in the memory vector. We will refer to this number as βN .

[†] This means they have a common factor.

(iii) Finally, we compute the probability that in the random walk defined by (1.9), $\omega^\theta - N/2$ will go from a positive to a negative number when we randomly change β/N 'steps'. P_δ will be twice this number.

Let's define $\omega^\theta = \frac{1}{2}m + N/2$ where

$$\begin{aligned} m &= n_1 - n_2 & N &= n_1 + n_2 \\ n_1 &= \text{number of 'steps to the right'} \\ n_2 &= \text{number of 'steps to the left'} \end{aligned}$$

steps to the right and left correspond to Ω_i^θ taking values one and zero respectively.

$P_N(m)$ is the usual binomial distribution associated with a random walk with a probability $\frac{1}{2}$ of moving either right or left.

$$P_N(m) = \frac{N!}{[\frac{1}{2}(N+m)]![\frac{1}{2}(N-m)]!} \left(\frac{1}{2}\right)^N \tag{3.1}$$

If we change just one component of a memory vector ξ , it is obvious because of (2.8) that the Ω_i^θ and $\Omega_{i-\theta}^\theta$ associated to it will also be modified, so we would naively expect that varying δN components in ξ we would change $2\delta N$ of the Ω_i^θ for a fixed θ . This is not the case in general, because if we change two entries that are θ components apart we can see that only two, and not four of the Ω are altered. If we modify δN entries in ξ we will change $\delta N(N - \delta N) + O(N)$ elements in the set Y defined in (2.15), so the average number of altered Ω for fixed θ when N goes to infinity is given by $2N(\delta - \delta^2)$, therefore β is

$$\beta = 2(\delta - \delta^2). \tag{3.2}$$

We need to know how many 'steps' will be to the right and to the left if we pick βN of them randomly. Defining $\beta N = r_1 + r_2$, where r_1 and r_2 are the number of steps to the right and to the left respectively. The probability of taking r_1 and r_2 in a given order is†

$$\begin{aligned} &\frac{n_1}{N} \frac{n_1-1}{N-1} \cdots \frac{n_1-r_1+1}{N-r_1+1} \frac{n_2}{N-r_1} \frac{n_2-1}{N-r_1-1} \cdots \frac{n_2-r_2+1}{N-r_1-r_2+1} \\ &= \frac{n_1!}{(n_1-r_1)!} \frac{n_2!}{(n_2-r_2)!} \frac{(N-r_1-r_2)!}{N!}. \end{aligned} \tag{3.3}$$

This is clearly independent of the order so we have to multiply by the degeneracy

$$\frac{(\beta N)!}{r_1! r_2!}$$

Using the following constraints to eliminate r_1, r_2, n_1 and n_2

$$\begin{aligned} m &= n_1 - n_2 & n_1 &= \frac{1}{2}(N + m) \\ N &= n_1 + n_2 & n_2 &= \frac{1}{2}(N - m) \\ s &= r_1 - r_2 & r_1 &= \frac{1}{2}(\beta N + s) \\ \beta N &= r_1 + r_2 & r_2 &= \frac{1}{2}(\beta N - s). \end{aligned} \tag{3.4}$$

† This is the probability of picking up r_1 white balls and r_2 black balls in a given order from a bag with N balls, n_1 white and n_2 black.

We obtain:

$$P_{\beta N}(r_1) = \frac{(\beta N)!(N - \beta N)!}{[\frac{1}{2}(N - \beta N + m - s)]![\frac{1}{2}(N - \beta N - m + s)]![\frac{1}{2}(\beta N + s)]![\frac{1}{2}(\beta N - s)]!} \times \frac{1}{P_N(m)2^N} \tag{3.5}$$

where we have also used equation (3.1).

The probability P_δ that for any $m > 0$ we have changed enough steps (Ω) so that the ‘new random walk’ finishes at the left of the origin (or $s > \frac{1}{2}m$), or equivalently, the probability that ω^θ changes from one to zero if we vary δN components of the memory vector it comes from, is given by

$$P_\delta = 2 \sum_{m>0} \sum_{\{s|s>\frac{1}{2}m\}} P_N(m)P_{\beta N}(s). \tag{3.6}$$

Substituting the expressions for $P_N(m)$ and $P_{\beta N}$, and making the Gaussian approximation in the ‘thermodynamic limit’ ($N \rightarrow \infty$), the right-hand side of equation (3.6) can be written as

$$\frac{2}{\pi N \sqrt{\beta(1-\beta)}} \int_0^\infty dm \int_{\frac{1}{2}m}^\infty ds \exp \left\{ -\frac{s^2}{N\beta(1-\beta)} + \frac{2sm}{N(1-\beta)} - \frac{m^2}{N(1-\beta)} \right\}. \tag{3.7}$$

After some algebra we obtain

$$P_\delta = \frac{2}{\sqrt{\pi}} \frac{\sqrt{\beta(1-\beta)}}{1-2\beta} \int_0^\infty dt \left\{ 1 - \text{erf}(t) \right\} \exp -\frac{4\beta(1-\beta)t^2}{(1-2\beta)^2}. \tag{3.8}$$

Now we have all the ingredients that we need to compute the distribution of Hamming distances in $\Gamma_{N/2}$, between a memory vector and itself with δN randomly changed entries.

Denoting the distance by d , we obtain

$$P(d) = \frac{1}{\sqrt{2\pi} \sigma} \exp -\frac{(d - \mu)^2}{2\sigma^2} \tag{3.9}$$

where

$$\mu = \frac{N}{2} P_\delta \quad \sigma = \left(N \frac{P_\delta(1-P_\delta)}{2} \right)^{1/2}. \tag{3.10}$$

Notice that if we are interested only in how errors propagate in percentage, in the thermodynamic limit the finite width of the Gaussian is irrelevant.

The fraction of modified entries in $\Gamma_{N/2}$ is given by P_δ , which in figure 1 is plotted as a function of δ . There it can be compared with the experimental results obtained for a system of 400 ‘neurons’.

In figure 2 we plot the experimental distribution of distances for $N = 400$ and $\delta = 0.05$, and we compare it with the best Gaussian fit; the only free parameter P_δ agrees with the one obtained by equation (3.10) up to the fourth decimal place.

If the dynamics chosen for $\Gamma_{N/2}$ were the Hopfield model, we know that we could recognise inputs with up to $\sim 30\%$ of their entries differing from a memory without appreciable error. This would imply, as can be directly read from the plot, that with the preprocessing we would be able to almost certainly recognise vectors differing by 12%. For a complete study of the convergence properties of the Hopfield model we refer the interested reader to [6].

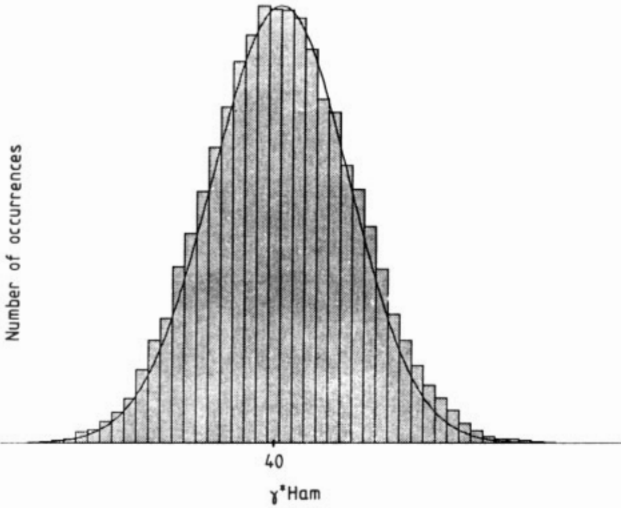


Figure 2. The experimental distribution of distances for $N = 400$ and $\delta = 0.05$ is compared with the best Gaussian fit; the only free parameter P_δ agrees with the one obtained by equation (3.10) up to the fourth decimal place. The height of the Gaussian has been normalised to one, and each bar represents a unit distance.

4. Propagation of correlated errors

We have already discussed the fact that the induced distance in Γ_N measures the ‘one-dimensional structure’. So it is interesting to ask how our system will behave when the error in the input preserves the structure of the state to some extent, for example the permutation of block of entries.

If we consider the vector as a ‘sentence’, a permutation of a block of entries will correspond to altering the order of the ‘words’ in it. In order to give a concrete example, consider the vector below thought of as a sentence with twenty words

$$(\underbrace{s_1, \dots, s_{20}}_A, \underbrace{s_{21}, \dots, s_{380}}_B, \underbrace{s_{381}, \dots, s_{400}}_C).$$

If we interchange the order of the blocks B and C, we obtain the vector

$$(\underbrace{s_1, \dots, s_{20}}_A, \underbrace{s_{381}, \dots, s_{400}}_C, \underbrace{s_{21}, \dots, s_{380}}_B).$$

In average the number of different entries between these two vectors (the average Hamming distance) would be 47.5%, and it would be impossible for the network to identify both of them.

However, if we use the distance induced by γ , we expect it to be much smaller, due to the fact that by a cyclic permutation ABC can go to CAB, and we have only to permute C and A which, on average, induces a change of only 5%.

Moreover, in a permutation many of the correlations between two entries are preserved, so we expect our distance to behave much better under such a change.

We have studied numerically how error propagates when we interchange two consecutive blocks of entries. The results are given in figure 3 where the fraction of modified entries in $\Gamma_{N/2}$ is plotted against the size of the permuted blocks for a system of 400 ‘neurons’.

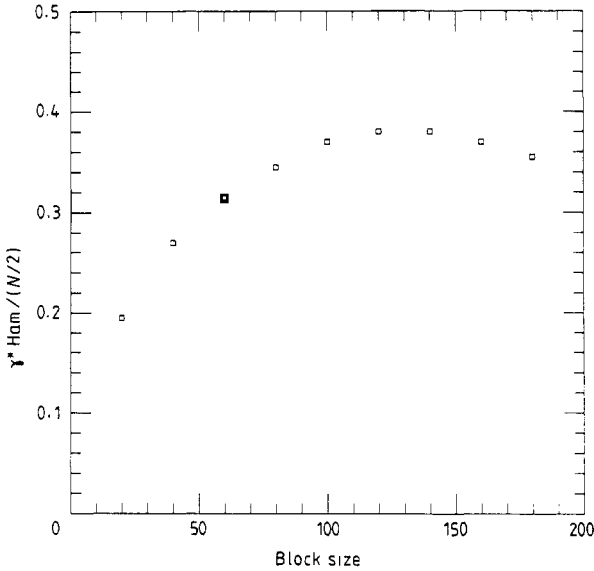


Figure 3. The fraction of modified entries in $\Gamma_{N/2}$, when we permute two neighbouring blocks of entries, is plotted against the size of the blocks for a system of 400 'neurons'. The error is approximately ± 0.05 .

Taking into account that a permutation of two blocks of size N corresponds (on average) to changing N entries of the vector, we can compare these results with those of the previous section (figure 1) and observe a significant improvement in the behaviour of the induced distance.

Of course, there are many different ways in which non-random errors can be introduced and there is no reason to restrict oneself to permutations of neighbouring blocks. We hope the results presented here are interesting enough to motivate further study.

5. Hardware implementation and biological systems

The only kinds of operations used in the preprocessing algorithm introduced in the previous sections are: addition modulo two and a threshold represented by the function Θ . All these operations are easily implementable in hardware, so our last condition for our map γ is fulfilled. A much more interesting question arises if we ask ourselves whether in principle the type of connections we have introduced to perform the logical operation XOR (which corresponds to the addition mod 2 used to construct the invariant functions) can be found in a biological system such as the human brain.

The answer is affirmative and, in order to explain why, we will fix our attention on a simplified model that retains all the properties we want to check. This is a single neuron whose dendrites are connected to two other neurons (figure 4).

The two operations

$$0 + 0 = 0$$

$$0 + 1 = 1$$

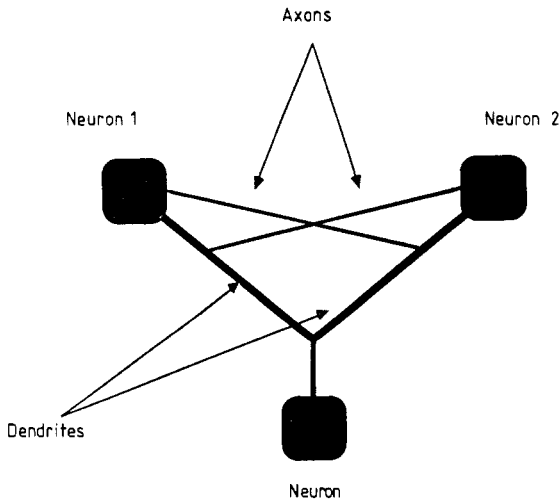


Figure 4. Neural circuit needed to perform the addition modulo two.

are implemented by adding the currents that flow in both dendrites. But in order to get $1 + 1 = 0$

we need a little more structure which is provided by the axons of neurons 1 and 2. We can connect the axon of neuron 1 to the dendrite of neuron 2 and vice versa with the interaction being of the inhibitory type, so that if both neurons are active these connections will kill the current flowing through the dendrites. Thus the current arriving to the neuron downstairs will be zero.

It is straightforward to check that such connections do not alter the outcome for the other possible cases.

These types of connections have already been considered for neural modelling in the biology literature. For a review see [7].

6. Conclusions and speculations

As we have seen, the introduction of different classes of distances can be easily obtained by a preprocessing mechanism. We have given a workable prescription for introducing distances which are invariant under arbitrary symmetry transformations acting in the neural network configuration space.

We have worked out explicitly the construction for C_p , and the results can be recapitulated as follows.

- (i) The network would recognise all the elements belonging to the equivalence class of a given memory, increasing the number of stored patterns by a factor of $2N$.
- (ii) Although the basin of attraction is shrunk, it would still be possible to recognise patterns with an appreciable amount of noise.
- (iii) The induced distance behaves much better under the permutation of a block of entries (that can be interpreted as highly correlated errors).

We would not like to finish without saying something about what we believe to be one of the most interesting features of the map γ . As we have already seen, vectors

that are close with respect to the Euclidean distance in Ψ can be mapped to the same state in $\Gamma_{N/2}$. We think it is an interesting question to ask how far this property can be pushed in order to obtain a network in which the recognition process could be carried by a map from the total configuration space Γ_N to a reduced one, where only the memories are represented. In that case the basin of attraction for a memory ξ would be given by $\gamma^{-1}(\xi)$, and no time evolution would be required at all.

Acknowledgments

We would like to thank especially R Shrock for his guidance, F Figueirido and J Figueroa for their continuous help and endurance, and B Razzaghe and S Sondhi for revising the final manuscript. We would also like to acknowledge many useful discussions with H Sah, J Salas and A Villarroel.

References

- [1] Hopfield J J 1982 *Proc. Natl. Acad. Sci., USA* **79** 2254
- [2] von der Marlsburg C and Bienenstock E 1987 *Europhys. Lett.* **3** 1243; **4** 121
Mikhailov A S 1988 *J. Phys. A: Math. Gen.* **21** L487
Kree R and Zippelius A 1988 *J. Phys. A: Math. Gen.* **21** L813
Dotsenko V S 1988 *J. Phys. A: Math. Gen.* **21** L783
- [3] Lautrup B 1988 The theory of the Hopfield model *Preprint* Nordita NBI-HE-88-06
Amit D J 1986 *Heidelberg Coll. on Glassy Dynamics* ed J L van Hemmen and I Morgenstern (*Lecture Notes in Physics* **275**) (Berlin: Springer)
- [4] Stern W 1926 *Psychology of Early Childhood* (Henry Holt)
- [5] Gibson E J 1965 *Science* **148** 1066-72
- [6] Kinzel W 1985 *Z. Phys. B* **60** 205; 1986 *Z. Phys. B* **62** 267
- [7] Koch C and Poggio T 1987 *Synaptic Function* ed G M Edelman, W Einar Gall and W Maxwell Cowan (New York: Wiley-Interscience)