The partitioning problem in unsupervised learning for nonlinear neurons

1993 J. Phys. A: Math. Gen. 26 7417

(http://iopscience.iop.org/0305-4470/26/24/016)

J. Phys. A: Math. Gen. 26 (1993) 7417–7426. Printed in the UK

The partitioning problem in unsupervized learning for nonlinear neurons

Adam Prügel-Bennett and Jonathan L Shapiro
Department of Computer Science, University of Manchester, Manchester, M13 9PL, UK

Received 16 August 1993, in final form 13 September 1993

Abstract. A closed form solution is found for the learning dynamics of a nonlinear Hebbian neuron presented with orthogonal patterns at different rates. The basins of attraction for each pattern are calculated as a function of the probability of the patterns being presented. A sample independent probability for learning a pattern is found in the limit of a large number of patterns. This function is, to a good approximation, proportional to the probability of the pattern being presented raised to a power, which depends strongly on the total number of patterns and on the nonlinearity of the response of the neuron to a stimuli. There is also a weak dependence on the distributions of the probabilities of the patterns being presented. The implications of this work for more realistic situations are discussed.

1. Introduction

Unsupervized learning is an important area of research in the field of neural networks. By using a Hebbian or Hebbian-like mechanism and allowing simple interactions between the neurons, a network can learn to form useful self-organized representations of its inputs. Models of unsupervized learning mechanisms have received considerable attention both because they provided plausible models for real neural systems and because they can be usefully exploited in artificial neural networks [1–5]. An example of the latter case is in a hybrid architecture where the inputs are 'pre-processed' by an unsupervized layer before being sent on to a normal back-propagation network [6, section 9.7]. The advantage of using a hybrid architecture is that, because the pre-processing layer does not require any feedback from later layers, it will learn relatively quickly.

The motivations for this paper are two-fold. First, neurons in various parts of the brain are found to respond to very specific input stimuli. The model neurons we examine here will learn to respond to particular patterns through a simple Hebbian learning rule. An important question is 'what is the probability that a neuron will learn a particular pattern?' or, equivalently, in a large group of neurons, 'how many neurons will, on average, learn to a particular input stimulus?' This could in principle be measured experimentally. This question is also important when using an artificial neural network to learn to clusters in some high-dimensional input space. The second motivation is to give an example of how partitioning of the input space, performed by an unsupervized network, can be calculated. Knowing this partitioning provides a complete description of the function performed by the network. The model analysed here gives a very clear illustration of how the partitioning can be obtained from the basins of attraction of the stationary points for the model, which in turn can be deduced by solving the dynamics. It also provides an illustration of how the functionality of the networks will depend on how it partitions the input space.

0305-4470/93/247417+10$07.50

The study of this partition has a long history. We give a very brief overview of some of this work. An important class of unsupervized networks are competitive networks [1–3, 7–9]. In these networks the neurons compete with each other to fire to the current input pattern. After many presentations of the input patterns, most of the neurons will have learned to fire either to a single pattern or to a cluster of patterns. The neurons thus partition the input space, hopefully finding some natural clustering of the input patterns. One aim in designing competitive networks has been to achieve a partitioning which properly reflects the structure of the input space. That is, to make the number of neurons that fire to patterns in a particular region of space proportional to the probability of such patterns being presented. To achieve this aim DeSieno [10] has proposed a 'conscience' mechanism to prevent neurons from learning too many patterns. An important elaboration on competitive networks is the feature map [2, 4, 11] in which the neurons try to preserve the topology of the input space. The partition problem for the Kohonen feature map has been studied by Ritter and Schulter [12–14]. A second important example of unsupervized learning is the 'linear Hebbian neuron' proposed by Oja [5]. Here a neuron learns to the maximal eigenvector of the pattern correlation matrix. Networks of linear neurons, with appropriately chosen interactions, can perform principal component analysis [15] or a similar decomposition of the input space [16]. Theses network can be viewed as partitioning the input space in the sense that the neurons pick out special directions, they will then respond strongly only to patterns aligned in these directions. These networks are useful in extracting important features from high-dimensional, noisy data. Recently a number of authors have studied unsupervized learning in networks with various different kinds of nonlinear neurons [17–19].

In this paper we consider the partitioning of the input space performed by a nonlinear Hebbian neuron [19]. The nonlinearity suppresses the firing of the neuron to weakly correlated patterns relative to more strongly correlated patterns. As mentioned above, for linear Hebbian neurons the partitioning problem is solved—the neuron learns the principal component of the pattern correlation matrix. For the nonlinear Hebbian neurons, the neuron will learn to individual input patterns, provided the response of the neuron to its inputs is sufficiently nonlinear. The probability of a particular pattern being learned will depend on the basin of attraction of the pattern, which will in turn depend on the size of the pattern and on the frequency with which it is presented. We will study the case, familiar in statistical mechanics, of a high-dimensional input space in which the patterns are uncorrelated (or weakly correlated). This provides a complementary view to the more frequently studied low-dimensional input spaces, where simulations can be used. It has the advantages that real data are usually high dimensional, but the disadvantage that they are usually highly structured. The problem we will consider is the partitioning achieved by the network when a set of random patterns is presented at different frequencies (that is, with different probabilities). The probability of a neuron learning a pattern, and hence the partitioning performed by a network of uncoupled neurons, will depend on the parameters of the neuron (in particular on the nonlinearity of the response of the neuron). By varying this parameter a variety of different partitionings can be achieved. The required behaviour will depend on the application. For example, it might be desired for the patterns to be learned with a probability proportional to the frequency with which they are presented. Alternatively, in other applications, it might be desirable to learn all the patterns with equal probability, or else to learn only the most frequently presented patterns.

The structure of this paper is as follows. In section 2 we will briefly describe the model of the nonlinear neuron. To solve the partitioning problem we proceed in two steps. We first calculate the basins of attraction for the fixed-point solutions by solving the learning dynamics. To do this we have approximated the update equations by a set of differential

equations which can be solved exactly for orthogonal patterns. This will be presented in section 3. This section can also be viewed as giving an analysis of the dynamics for a nonlinear neuron to complement the stationary point analysis [19]. From the solutions to the dynamics we obtain a simple condition for determining which pattern will win that depends only on the frequencies of occurrence of the patterns, the initial overlaps, and a single parameter of the model. In the second step we consider the case of a large number of patterns which allows us to calculate a sample-independent probability for a pattern to be learned. This calculation is given in section 4. We find that, to a good approximation, if a pattern is shown with a relative frequency proportional to r, with $0 < r \leqslant 1$, then the probability of it being learned is

$$p(r) \approx \frac{(1+x)}{P} r^x \tag{1.1}$$

where $x \approx 2\log(P)/(b-1)$, P is the number of patterns, and b is a parameter that controls the nonlinearity of the neuron's response. If b is chosen so that $x = 1$, then the probability of a neuron learning a pattern will be proportional to the frequency with which the pattern is presented.

In the final sections we discuss the implications of this work to more realistic situations. In particular we consider the behaviour of neurons with a sigmoid activation function, and we briefly outline what we expect to happen when the input patterns are more complex. We also discuss the importance of introducing inhibitory interactions in order to achieve a desired partitioning.

2. The model neuron

The model neuron we will consider can be viewed as a nonlinear extension to Oja's model [5]. The neuron receives N inputs through modifiable synapses w_i, where $i = 1, \ldots, N$. We will study the situation when a set of P patterns are presented to the neurons. We represent the patterns by a set of vectors $\boldsymbol{\xi}^\mu = (\xi_1^\mu, \ldots, \xi_N^\mu)$, where the superfix $\mu = 1, \ldots, P$ labels the different patterns.

The patterns produce a post-synaptic potential V^μ in the cell, given by

$$V^\mu = \sum_{i=1}^{N} \xi_i^\mu w_i = \boldsymbol{\xi}^\mu \cdot \boldsymbol{w}. \tag{2.1}$$

The cell is assumed not to fire when the post-synaptic potential, V^μ, is negative and to fire according to a simple power law when V^μ is positive. Denoting the activation function by $A(V^\mu)$, then

$$A(V^\mu) = (V^\mu)^b \Theta(V^\mu) = \begin{cases} (V^\mu)^b & V^\mu > 0 \\ 0 & V^\mu \leqslant 0 \end{cases} \tag{2.2}$$

where b measures the degree of nonlinearity in the response of the neuron.

On presenting a pattern $\boldsymbol{\xi}^\mu$, the synaptic weights, w_i, change according to the update rule

$$\delta w_i = r A(V^\mu)(\xi_i^\mu - V^\mu w_i). \tag{2.3}$$

The first term can be viewed as a 'linear Hebbian' term—the change in weight is proportional to the input activity and the cell activation. The second term can be thought of as a decay-like term which ensures that the weight vector w becomes normalized, since

$$w \cdot \delta w = \tfrac{1}{2}\delta|w|^2 = r(1-|w|^2)A(V^\mu)V^\mu \tag{2.4}$$

but $A(V^\mu)V^\mu \geqslant 0$, so when $|w| < 1$ then $|w|$ increases, while when $|w| > 1$ then $|w|$ decreases. Close to the fixed point $A(V^\mu)V^\mu \approx 1$, so that r must be less than one for the weight vector to converge.

When the learning rate, r, is sufficiently small the discrete dynamics (2.3) can be replaced by a differential equation. In this limit we can also make the adiabatic approximation of considering the updating to occur after presenting all the patterns. We will assume that the patterns are presented at different rates, so that the learning towards a pattern will be weighted by its frequency. We define the frequency of presentation of pattern μ to be proportional to r^μ. Using these assumptions the learning equation (2.3) becomes

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \sum_\mu r^\mu A(V^\mu)(\xi_i^\mu - V^\mu w_i). \tag{2.5}$$

Although this is strictly true only in the limit of an infinitesimal learning rate, it will be a good approximation provided δw_i is small. However, when the overlaps, V^μ, are small, δw_i will be small, even if r is of order 1. Thus, this approximation will be valid for the initial dynamics, and will only break down when V^μ becomes large, but when V^μ is large the pattern which will be learned has already been determined. Thus we expect that equation (2.5) will give a good prediction for which pattern is learned, even when r is large.

We note that, if the patterns have different lengths, then we would obtain the same dynamics by simultaneously rescaling the patterns $\xi^\mu \to \xi^\mu/|\xi^\mu|$ and the r^μ's, $r^\mu \to |\xi^\mu|^{b+1}r^\mu$. Therefore, the effect of having different length patterns can be absorbed into the r^μ's. In this paper we will assume that the patterns all have the same length. Clearly the generalization to patterns with different lengths is straightforward.

The final state learned by the neuron depends on the parameter b and on the correlation between the patterns. When b is less than some critical value, which depends on the intra-pattern correlation, the weight vector, w, will learn to some mixture of the patterns. When $b = 1$ (and there is no threshold) this model is identical to Oja's linear neuron. In this case, the neuron learns to the maximal eigenvector (or principal component) of the pattern correlation matrix $M_{ij} = \sum_\mu \xi_i^\mu \xi_j^\mu$. For b above its critical value the neuron learns to one of the input patterns. Which pattern it learns depends on where it started. (In the case when the weight vector starts negatively correlated with all the patterns the neuron will never learn, but the probability of this happening is 2^{-P}, and we will not discuss this possibility further.) A fuller analysis of this model is given in [19].

In this paper we consider the case when the stimuli ξ_i^μ are independent randomly chosen variables with $\langle(\xi_i^\mu)^2\rangle = 1/N$. Thus $\xi^\mu \cdot \xi^\nu = \delta^{\mu,\nu} + \mathcal{O}(1/\sqrt{N})$. We shall consider the case when N is sufficiently large that the intra-pattern correlations can be neglected; i.e. the patterns can be considered to be orthogonal. For orthogonal patterns the critical value of b is 1. In the rest of this paper we will consider only the case $b > 1$, so that the weight vector learns to align itself with one of the patterns. Although we have considered here only a single neuron, we can consider a network consisting of a set of uncoupled neurons. Provided the initial weight vectors are different from each other, the neurons are unlikely to learn the same pattern. Such a network, however, will not find a very even partitioning of the inputs due to random fluctuations. In section 5 we discuss briefly how including interactions between the neurons can improve the partitioning.

3. Solutions of the learning dynamics

For general patterns the set of differential equations (2.5) cannot be solved as they are coupled and nonlinear. However, for orthogonal patterns the equations decouple and are of a form that can be solved. To see this we start by multiplying equation (2.5) through by ξ_i^μ and summing over i, to obtain a set of equations for the evolution of the overlaps

$$\frac{dV^\mu}{dt} = \sum_\nu r^\nu A(V^\nu)(\boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}^\nu - V^\mu V^\nu) \tag{3.1}$$

where we have used $V^\mu = \boldsymbol{\xi}^\mu \cdot \boldsymbol{w}$. Since the patterns are orthogonal, this can be written as

$$\frac{dV^\mu}{dt} = r^\mu A(V^\mu) - V^\mu g(t) \tag{3.2}$$

where

$$g(t) = \sum_\nu r^\nu A(V^\nu) V^\nu.$$

Although we do not know the form of the function $g(t)$, it is the same for each pattern. Equation (3.2) can be solved exactly because, when V^μ is less than zero, $A(V^\mu) = 0$ and the equation is linear, while when V^μ is greater than zero, $A(V^\mu) = (V^\mu)^b$ and the differential equation has the form of a Bernoulli equation. This can be turned into a linear equation by making the substitution $u = (V^\mu)^{1-b}$ and then solved by introducing an integrating factor.

In order to write the final solution in a more elegant form it is useful to note that, from (3.2),

$$\sum_\mu V^\mu \frac{dV^\mu}{dt} = \frac{1}{2}\frac{d}{dt}\sum_\mu (V^\mu)^2 = \left(1 - \sum_\mu (V^\mu)^2\right) g(t) \tag{3.4}$$

so that

$$g(t) = -\frac{d}{dt}\left(\ln \sqrt{1 - \sum_\mu (V^\mu)^2}\right). \tag{3.5}$$

Substituting (3.5) into equation (3.2) we find the overlaps evolve according to

$$V^\mu(t) = \begin{cases} V_0^\mu Y(t) & V_0^\mu < 0 \\ \dfrac{V_0^\mu Y(t)}{\left(1 - r^\mu (b-1)(V_0^\mu)^{b-1} \int_0^t (Y(t'))^{b-1}\, dt'\right)^{1/(b-1)}} & V_0^\mu > 0 \end{cases} \tag{3.6}$$

where

$$Y(t) = \sqrt{\left(1 - \sum_\mu (V^\mu)^2\right) \Big/ \left(1 - \sum_\mu (V_0^\mu)^2\right)}. \tag{3.7}$$

It is easy to understand the dynamics of the overlaps from equation (3.6). Initially $Y(t) = 1$, and it remains close to 1 so long as the overlaps V^μ are all small. Thus for small

times, and for those patterns that started positively correlated with the neuron, the overlaps grow as

$$V^\mu(t) \approx \frac{V_0^\mu}{(1 - r^\mu(b-1)(V_0^\mu)^{b-1}t)^{1/(b-1)}}. \tag{3.8}$$

The overlaps all remain small until $t \approx 1/(r^\mu(b-1)(V_0^\mu)^{b-1})$ for one of the patterns. When this happens the overlap for the pattern in question grows rapidly and forces $Y(t)$ to zero, which then sends the overlaps for the other patterns to zero. Since the integral in the denominator of equation (3.6) is identical for each pattern, the pattern that will be learned is that for which

$$r^\mu(V_0^\mu)^{b-1} > r^\nu(V_0^\nu)^{b-1} \qquad \forall \nu \neq \mu. \tag{3.9}$$

This defines the basin of attraction for each pattern. For truly orthogonal patterns, and in the limit of infinitesimally small r, this condition is exact. For large random patterns and r of order 1, this should still be a good approximation. To calculate the probability of a particular pattern being learned we must average over all possible initial overlaps, V_0^μ. Since the patterns are high dimensional with components, ξ_i^μ, which are independent random variables, the overlaps will be Gaussian distributed. Thus the probability of learning a pattern, ξ^μ, which is presented with a probability $r^\mu / \sum_\nu r^\nu$ is

$$p(r^\mu) = \int_0^\infty \frac{\mathrm{d}V_0^\mu}{\sqrt{2\pi}} \mathrm{e}^{-(V_0^\mu)^2/2} \prod_{\nu \neq \mu} \left(\int_{-\infty}^\infty \frac{\mathrm{d}V_0^\nu}{\sqrt{2\pi}} \mathrm{e}^{-(V_0^\nu)^2/2} \Theta\left((r^\mu)^{1/(1-b)} V_0^\mu - (r^\nu)^{1/(1-b)} V_0^\nu \right) \right). \tag{3.10}$$

For a given set of frequencies, $\{r^\mu\}$, we can calculate $p(r^\mu)$ by numerically integrating equation (3.10).

4. Partitioning

In this section we consider the case when there is a large number of patterns, so that the probability of learning a pattern becomes sample independent. To make this more precise we assume that the r^μ's are drawn from some distribution $\rho(r)$. In the large-P limit the probability of learning a pattern self-averages, so that $p(r^\mu)$ does not depend on the other r^ν's (although it will depend on the distribution $\rho(r)$). Since only the relative frequencies of presentation are important, we are free to choose the scale of the r^μ's. In the following we will choose this scale so that r^μ has a maximum of 1.

If all the patterns are shown equally often (i.e. $\rho(r) = \delta(r-1)$), the integral in equation (3.10) can be performed exactly giving a probability of learning each pattern of $(1 - 2^{-P})/P$. When the r^μ's come from a more complicated distribution the integral cannot be performed exactly and we must resort to a saddle-point evaluation. The saddle-point equation gives an equation for $p(r^\mu)$. It turns out that for large P, and moderately large b, this is well approximated by a simple power law. We have used least-squares fitting to calculate this power law.

The first step in this calculation is to rewrite equation (3.10) as

$$p(r^\mu) = \int_0^\infty \frac{\mathrm{d}x}{2\pi} \exp\left\{ -\frac{x^2}{2} + \sum_{\nu \neq \mu} \log\left[\Phi\left(\left(\frac{r^\nu}{r^\mu} \right)^{1/(b-1)} x \right) \right] \right\} \tag{4.1}$$

where $\Phi(y)$ denotes the normal probability function, defined by

$$\Phi(y) = \int_{-\infty}^{y} e^{-x^2/2} \frac{dx}{\sqrt{2\pi}}. \tag{4.2}$$

Since there are a very large number of patterns we can replace the sum in equation (4.1) with its average

$$\sum_{\nu \neq \mu} \log\left[\Phi\left(\left(\frac{r^\nu}{r^\mu}\right)^{1/(b-1)} x\right)\right] \approx (P-1)\int_0^1 \rho(r) \log\left[\Phi\left(\left(\frac{r}{r^\mu}\right)^{1/(b-1)} x\right)\right] dr. \tag{4.3}$$

Making the substitution $u = (r^\mu/r)^{1/(1-b)}x$ we can write this as

$$-(P-1)(b-1)r^\mu x^{b-1} \int_{(r^\mu)^{1/(b-1)}x}^{\infty} \rho\left(r^\mu \left(\frac{x}{u}\right)^{b-1}\right) \log[\Phi(u)] \frac{du}{u^b}. \tag{4.4}$$

Equation (4.1) has a saddle point for $x \sim \mathcal{O}(\sqrt{\log(P)})$. For large P the integral above will be well approximated by its asymptotic expansion. Expanding $\log[\Phi(u)]$ for large u and performing the asymptotic expansion, equation (4.4) becomes

$$(P-1)(b-1)(r^\mu)^{-2/(b-1)} \frac{1}{\sqrt{2\pi}x^2} \rho(1) \exp\left\{-\frac{(r^\mu)^{2/(b-1)}x^2}{2}\right\} \tag{4.5}$$

where we have ignored terms of order $1/\log(P)$. Putting this back into equation (4.1) and differentiating with respect to x we find that the integrand is a maximum when

$$\frac{x^2}{2} = (r^\mu)^{-2/(b-1)}\left(\log(P-1) + \log\left(\frac{(b-1)\rho(1)}{4\sqrt{\pi}}\right) - \tfrac{3}{2}\log(\log(P-1)) + \cdots\right). \tag{4.6}$$

Substituting back into equation (4.1), we find to leading order in r

$$p(r) = C \exp(-Br^{-2/(b-1)}) \tag{4.7}$$

where the normalization constant, C, is given by

$$C = \frac{2}{P(b-1)\Gamma((b-1)/2, B)B^{(b-1)/2}} \tag{4.8}$$

where $\Gamma((b-1)/2, B)$ is the incomplete gamma function, and where B is, ignoring terms smaller than those shown,

$$B = \log(P) + 1 + \log\left(\frac{(b-1)\rho(1)}{4\sqrt{\pi}}\right) - \tfrac{3}{2}\log(\log(P)). \tag{4.9}$$

From equations (4.7)–(4.9) we see that, in the large-P limit, the leading term in B is $\log(P)$ and all the patterns which occur with a rate $r < 1$ will be very strongly suppressed. The leading corrections will be of order $\log(\log(P))$. However, for a realistic number of patterns $\log(P)$ never becomes huge and $\log(\log(P))$ will be of the same order as the constant terms. For example, when $b = 2$ and $P = 100$ then $B \approx 2.7$, when $b = 2$ and $P = 1000$ then $B \approx 4.9$, and when $b = 10$ and $P = 100$ then $B \approx 4.4$. The precise value

of B will depend on the distribution of the frequencies, $\rho(r)$, although this dependence is only slight. If, for example, $\rho(r^\mu) = (a+1)r^a$ then $\rho(1) = (a+1)$ and the dependence of $p(r)$ on a occurs only in $B = B' + \log(1+a)$.

Equation (4.7) is somewhat cumbersome and it turns out that, for moderate values of b, this function is well approximated by a simple power law, $p(r) \propto r^x$. To calculate x we perform a least-squares fit. We define an error

$$E(x) = \int_0^1 (p(r) - (x+1)r^x)^2 \, \mathrm{d}r. \tag{4.10}$$

By an appropriate change of variables we can write $E(x)$ in terms of incomplete gamma functions. We find that $E(x)$ is minimized when x satisfies

$$\frac{B^{(b-1)(x-1)/2}\Gamma(-(b-1)x/2, B)}{\Gamma(-(b-1)/2, B)} = \frac{1+x}{2x}. \tag{4.11}$$

To solve for x we use the (truncated) continued fraction expansion for the incomplete gamma function

$$\Gamma(a,x) \approx \frac{\mathrm{e}^{-x}x^{-a}}{x+1-a}. \tag{4.12}$$

Using this, the optimum value for x is $2(B+1)/(b-1)$, while the error $E(x)$, for this optimum value, is given by $2(2B+1+b)^2/(b-1)(4B+1+b)(4B+3+b)$, which for large B is approximately $1/[2(b-1)]$. Thus $p(r)$ is approximately given by

$$p(r) \approx \frac{1}{P}\left(1 + \frac{2B+2}{b-1}\right) r^{2(B+1)/(b-1)} \tag{4.13}$$

where the approximation becomes increasingly good as b is increased. If we wish $p(r)$ to be proportional to r then, using equations (4.9) and (4.13), we find that for $P = 100$ the nonlinearity b should be ≈ 6.5, while for $P = 1000$, $b \approx 8.5$.

5. Conclusion

We have seen that, for neurons with a simple power-law activation function, and with effectively orthogonal input patterns, we can calculate the probability of it learning a particular pattern, and by altering the degree of nonlinearity, we can achieve a variety of different behaviours. In this section we discuss what happens when we relax some of these conditions.

We consider first the effect of using a sigmoid activation function. For highly nonlinear responses using a sigmoid activation function can greatly increase the speed of learning. The reason for this is clearly seen by considering the approximate solutions to the dynamics equation (3.8). We see that (within this approximation) the neuron will learn when

$$t = \frac{1}{r^\mu(b-1)(V_0^\mu)^{b-1}}. \tag{5.1}$$

But V_0^μ is of order $1/\sqrt{N}$, therefore, as we increase b, the time it takes for the neuron to learn grows as $N^{(b-1)/2}$. To compensate for this we would like to increase the learning rate

r in equation (2.3). However, as we have already noted, r must be less than 1 in order for the weight vector to converge to a pattern. To overcome this problem we can use a sigmoid activation function, for example,

$$A(V^\mu) = \frac{c(V^\mu)^b}{1 + c(V^\mu)^b}\Theta(V^\mu). \tag{5.2}$$

This is clearly always less than 1, but for small V^μ it is well approximated by the simple power law

$$A(V^\mu) \approx c(V^\mu)^b\Theta(V^\mu).$$

Thus we have effectively increased the learning rate by a factor c. Provided $c(V^\mu)^b$ is negligible compared with 1, the sigmoid function is essentially a simple power law and the analysis given in this paper will apply to the sigmoid function. This approximation will break down only when V^μ becomes macroscopic, but the neuron spends a negligible amount of time in this region before it learns. Thus using a sigmoid function (with not too large a c) should not significantly alter the final state that is learned.

The effect of introducing correlated patterns is much more complicated. If the input patterns are random but low dimensional so that the intra-pattern correlation is significant, then some patterns might be preferentially learned. If the patterns are not single points in input space but extended (for example, they may be clusters of points), but otherwise random, then the neuron would learn to the extended patterns just as they learned to a single pattern. The fixed point in this case would be close to the 'centre of gravity' of the patterns. To see this we assume that the patterns are distributed according to some distribution $P(\boldsymbol{\xi})$. Again, using equation (2.3) and making an adiabatic approximation, the change in weight vector will be given by

$$\langle \delta \boldsymbol{w} \rangle = \int P(\boldsymbol{\xi})A(V)(\boldsymbol{\xi} - V\boldsymbol{w})\,\mathrm{d}\boldsymbol{\xi} \tag{5.4}$$

where the integral is over the space of the patterns. At a fixed point, $\boldsymbol{w}^*$ say, $\langle \delta \boldsymbol{w} \rangle = 0$. Writing $\boldsymbol{\xi} = \boldsymbol{w}^* + \boldsymbol{x}$, the fixed point is then given by

$$\boldsymbol{w}^* = C \int P(\boldsymbol{w}^* + \boldsymbol{x})A(1 + \boldsymbol{w}^* \cdot \boldsymbol{x})\boldsymbol{x}\,\mathrm{d}\boldsymbol{x} \tag{5.5}$$

where C is a normalization constant. If the patterns are normalized $\boldsymbol{w}^* \cdot \boldsymbol{x} = -|\boldsymbol{x}^2|/2$, and around the fixed point, $A(1 + \boldsymbol{w}^* \cdot \boldsymbol{x})$ is very nearly constant. Further from the fixed point $A(1 + \boldsymbol{w}^* \cdot \boldsymbol{x})$ falls to zero. Thus the neuron learns to the 'centre of gravity' of the patterns multiplied by a local weighting function. For sigmoid functions this local weighting function is closer to 1 for small $|\boldsymbol{x}|$, and then falls off more rapidly to zero.

Although we have shown that we can control the probability with which a neuron will learn a pattern by altering b, we should note that this does not guarantee that a group of neurons will learn a set of input patterns with the probability that we would desire. This is because, by chance, some patterns might be learned by several neurons while another pattern, which has an equal chance of being learned, is not learned at all. To overcome these random fluctuations we can introduce a small inhibitory interaction between the neurons so that when a pattern is learned by one neuron the probability of another neuron learning the same pattern is reduced. This kind of competitive network is more complicated to analyse because of the coupling between the neurons. We discuss the partition problem for this network elsewhere [20].

References

[1] von der Marlsburg C 1973 Self-organization of orientation selective cells in the striate cortex *Kybernetik* **14** 85–100

[2] Willshaw D J and von der Marlsburg C 1976 How patterned neural connections can be set up by self-organization *Proc. R. Soc.* **194** 431–45

[3] Grossberg S 1976 Adaptive pattern classification and universal recording: I. parallel development and coding of neural feature detectors *Biol. Cybern.* **23** 121–34

[4] Kohomen T 1982 Self-organized formation of topologically correct feature maps *Biol. Cybern.* **43** 59–69

[5] Oja E 1982 A simplified neuron model as a principal component analyzer *J. Math. Biol.* **15** 267–73

[6] Hertz J A, Krogh A S and Palmer R G 1991 *Introduction to the Theory of Neural Computation* (London: Addison–Wesley)

[7] Fukushima K 1975 Cognitron: a multilayered neural network *Biol. Cybern.* **20** 121–36

[8] Amari S A 1980 Topographic organization of nerve fields *Bull. Math. Biol.* **42** 339–64

[9] Rumelhart D E and Zipser D 1985 Feature discovery by competitive learning *Cognitive Sci.* **9** 75–112 (Also printed in *Parallel Distributed Processing* vol 1, ed Rumelhart *et al*, ch 5)

[10] DeSieno D 1989 Adding a conscience for competitive learning *Proc. Int. Conf. on Neural Networks I* (New York: IEEE) pp 117–24

[11] Kohonen T 1988 *Self-Organization and Associative Memory* 3rd edn (Berlin: Springer)

[12] Ritter H and Schulter K 1986 On the stationary state of Kohenen's self-organizing sensory mapping *Biol. Cybern.* **54** 99–106

[13] Ritter H and Schulter K 1988 Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimensional selection *Biol. Cybern.* **60** 59–71

[14] Ritter H and Schulter K Kohonen's self-organizing maps: Exploring their computational capabilities *IEE Int. Conf. on Neural Networks (San Diego 1988)* vol I (New York: IEEE) pp 109–16

[15] Sanger T D 1989 Optimal unsupervized learning in a single-layer linear feedforward neural network *Neural Networks* **2** 459–73

[16] Oja E 1989 Neural networks, principal components, and subspaces *Int. J. Neural Syst.* **1(1)** 61–8

[17] Softky W R and Kammen D M 1991 Correlations in high-dimensional or aymmetric data sets: Hebbian neuronal processing *Neural Networks* **4** 337–47

[18] Schuster H G 1992 Learning by maximizing the information transfer through nonlinear noisy neurons and 'noisy breakdown' *Phys. Rev.* A **46** 2131–8

[19] Prügel-Bennett A and Shapiro J L 1993 Statistical mechanics of unsupervized Hebbian learning *J. Phys. A: Math. Gen.* **26** 2343–69

[20] Prügel-Bennett A and Shapiro J L 1993 The partitioning problem in competitive networks (in preparation)