On measuring similarity between different two-layered networks

1996 J. Phys. A: Math. Gen. 29 1581

(http://iopscience.iop.org/0305-4470/29/8/007)

J. Phys. A: Math. Gen. **29** (1996) 1581–1591. Printed in the UK

On measuring similarity between different two-layered networks

Marcelo G Blatt

Department of Physics of Complex Systems, The Weizmann Institute of Science, Rehovot 76100, Israel

Received 8 March 1995, in final form 8 November 1995

Abstract. In this paper we present a method for calculating ϵ_g, the *generalization error* of two-layered networks. ϵ_g is the fraction of the input space for which two networks yield different answers, therefore it is a good index to measure the similarity between them. The method presented here is an extension of work reported previously. It is applied here to the case of a single-layer perceptron (which can be regarded as a particular two-layered perceptron) that tries to imitate a two-layered network. The particular realizations of such a two-layered network that are analysed here are the 'parity machine', the 'and machine' and the 'committee machine'. We have also compared the input–output mapping of a committee and a parity machine.

1. Introduction

Feedforward neural networks can be viewed as input–output devices whose parameters are tuned to perform a given function. An index of similarity between two such mappings is the *generalization error*, i.e. the fraction of the input space for which the corresponding function value is different.

The generalization error, ϵ_g, as a function of the number of examples was studied in the framework of learning theory [1]. Different asymptotic behaviours were found for the cases in which the rule can or cannot be implemented [2]. Numerical simulations were also employed to understand how networks generalize when they 'try' to implement an unlearnable task [3].

There are many situations where it can be useful to calculate the similarity of the input–output map of two networks. For instance, one may want to evaluate the generalization ability of a learning algorithm. The method presented here could be applied if for a given learning algorithm it were possible to calculate the overlap between the weights of the 'teacher' and 'student' (as in the case of [4]).

More recently a method to calculate ϵ_g when the student and teacher are *two-layered perceptrons* having the same [5] and different [6] numbers of hidden units was introduced. In this case ϵ_g is a function of the two network parameters. Expressions for ϵ_g were provided in the thermodynamic limit, i.e. when the number of inputs is very large.

In this work we extend the method mentioned above in two directions. First, it is formulated in such a way that any mapping from the hidden units to the output is allowed. Second, a series expansion that enables us to calculate ϵ_g to any degree of precision replaces the multi-dimensional Gaussian integrals, in terms of which the results were previously given [5, 6].

0305-4470/96/081581+11$19.50

This new method is equivalent to the limit of zero temperature and infinite number of examples in the framework of the replica calculations. Therefore this technique can be useful to check symmetry or scaling assumptions for the order parameters. In particular, we compare with the case of a perceptron learning from a committee machine with three hidden units [7] and when the number of hidden units goes to infinity [8, 9].

This method is applied now to the case of a *single-layer perceptron* trying to learn the function realized by a *two-layered network*. In most of the cases this is an unlearnable task for the perceptron. We obtain expressions for the general case and also for some specific realizations of the two-layered network, like the *parity*, *and*, and *committee machines*. In particular, we are able to find the weights of the perceptron that minimize ϵ_g. Some of our results can be easily explained by geometrical arguments. We found that when the teacher is a committee machine of any number of hidden units, there exists a perceptron that is able to give the correct answer in almost 80% of the cases.

We have also compared the set of Boolean functions associated with committee and parity machines with tree-like architecture. We found that the intersection of the set of Boolean functions that can be implemented by a committee machine and the set of Boolean functions that can be implemented by a parity machine is empty.

The basic definitions are given in section 2. The method for calculating ϵ_g and a brief derivation of it is presented in section 3. In sections 4 and 5 we apply this technique to the cases mentioned above.

2. Definitions

Throughout this paper we will be concerned with feedforward networks composed of binary units, with N inputs and one output. Each input is described by an N-vector $\boldsymbol{x}$ with components $x_i \in \{-1, 1\}$, $1 \leqslant i \leqslant N$.

The simplest feedforward network is the *single-layer perceptron* (SLP) whose output is given by:

$$y_p(\boldsymbol{x}) = \text{sign}(\boldsymbol{W} \cdot \boldsymbol{x}) \tag{1}$$

where $\boldsymbol{W} \in \Re^N$ is called the *weight vector*; W_i denotes the strength of the connection of the ith input unit to the output.

We consider spherical perceptrons, i.e. $\boldsymbol{W}$ is normalized by requiring $\boldsymbol{W} \cdot \boldsymbol{W} = 1$. The SLP implements only the class of so-called *linearly separable* functions. That is, the input–output map (1) implemented by the SLP divides the input space into two regions, corresponding to the two sides of the hyperplane that passes through the origin and is normal to $\boldsymbol{W}$.

Two-layered networks (2LN) with one additional layer of hidden units have higher computational power than a perceptron. A 2LN is completely defined by the specification of the number of hidden units, K; the weight vectors $\boldsymbol{W}_l \in \Re^N$, $1 \leqslant l \leqslant K$ and by the Boolean function that maps the hidden layer to the output. Each hidden unit is connected to the inputs by its weight vector $\boldsymbol{W}_l$, performing the mapping

$$\sigma_l = \text{sign}(\boldsymbol{W}_l \cdot \boldsymbol{x}) \qquad \text{for } 1 \leqslant l \leqslant K\,.$$

The hidden units can be regarded as outputs of a single-layer perceptron. The state taken by the hidden layer in response to an input is called the *internal representation* (IR) corresponding to this input. The output of the network is determined by the IR:

$$y_{2LN}(\boldsymbol{x}) = B(\boldsymbol{\sigma})$$

where B: $\{-1, 1\}^K \longrightarrow \{-1, 1\}$ denotes the Boolean function that maps the hidden layer to the output unit. For instance the *parity machine* (PM) is a 2LN where the mapping from the hidden units to the output is

$$B(\boldsymbol{\sigma}) = \prod_{l=1}^{K} \sigma_l \,.$$

Another 2LN that was widely studied is the *committee machine*† (CM) whose output is determined by the votes of each of its hidden units; that is, the Boolean function is implemented by a SLP whose weight vector components have the same (positive) value,

$$B(\boldsymbol{\sigma}) = \operatorname{sign}\left(\sum_{l=1}^{K} \sigma_l\right).$$

The third 2LN that we consider in this work is the *and machine* (AM). In this case the output is $+1$ if and only if all the hidden units are equal to $+1$, otherwise the output is -1.

A particular case of a 2LN is the *ruler machine* (RM) where the output is determined by a *single* hidden unit. Clearly its computational capabilities are exactly the same as the SLP. The Boolean function $y_{2\mathrm{LN}} : \{-1, 1\}^N \to \{-1, 1\}$ is the *input–output map* of the 2LN.

3. The generalization error

The generalization error, ϵ_g, is an index of similarity of the input–output map implemented by two networks. ϵ_g is the fraction of the input space for which two networks give different outputs.

Let us now consider two 2LNs, $\mathcal{N}_1$ and $\mathcal{N}_2$, both with N inputs and, respectively, K_1 and K_2 hidden units. Use $W_{li}^{(1)}$ and $W_{li}^{(2)}$ to denote the weights of $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively. Similarly, use B_1 and B_2 with the same convention.

The generalization error between them is given by

$$\epsilon_g(\mathcal{N}_1, \mathcal{N}_2) = \langle\langle \Theta(-y_1 y_2) \rangle\rangle \tag{2}$$

where $\langle\langle \cdots \rangle\rangle = \frac{1}{2^N} \sum_{x_1=-1,1} \cdots \sum_{x_N=-1,1} \ldots$ indicates the average over input space, and $\Theta(\cdot)$ denotes the Heaviside step function. $\epsilon_g(\mathcal{N}_1, \mathcal{N}_2)$ can be expressed in terms of the IR as follows:

$$\epsilon_g(\mathcal{N}_1, \mathcal{N}_2) = \sum_{\mu_1=1}^{2^{K_1}} \sum_{\mu_2=1}^{2^{K_2}} \Theta\left(-B_1(\boldsymbol{\sigma}^{\mu_1})\, B_2(\boldsymbol{\sigma}^{\mu_2})\right) P(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}) \tag{3}$$

where $\{\boldsymbol{\sigma}^{\mu_a}\}_{1 \leqslant \mu_a \leqslant 2^{K_a}}$ is the set of possible IR for K_a hidden units, $a = 1, 2$. $P(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2})$ is the fraction of input space for which the two 2LNs get the IRs $\boldsymbol{\sigma}^{\mu_1}$ and $\boldsymbol{\sigma}^{\mu_2}$ simultaneously (i.e. in response to the same input):

$$P(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}) = \left\langle\left\langle \prod_{l=1}^{K_1} \Theta\left(\boldsymbol{W}_l^{(1)} \cdot \boldsymbol{x}\sigma_l^{\mu_1}\right) \prod_{m=1}^{K_2} \Theta\left(\boldsymbol{W}_m^{(2)} \cdot \boldsymbol{x}\sigma_l^{\mu_2}\right) \right\rangle\right\rangle \tag{4}$$

$P(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2})$ can be interpreted as the probability of getting the IRs $\boldsymbol{\sigma}^{\mu_1}$ and $\boldsymbol{\sigma}^{\mu_2}$ if an unknown vector $\boldsymbol{x}$ is fed as the input of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively.

Introducing the integral expression of the Θ-function [5, 6] in equation (4), we obtain at leading order in $1/N$

$$P\left(\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}\right) = \frac{1}{\sqrt{(2\pi)^{K_1+K_2} \det(R)}} \int_0^{\infty} \prod_{l=1}^{K_1+K_2} \mathrm{d}h_l \exp\left[-\frac{1}{2} \sum_{m,n=1}^{K_1+K_2} h_m \left(R^{-1}\right)_{mn} h_n\right] \tag{5}$$

† Also called *majority machine.*

where R is the (symmetric) correlation matrix given by

$$R = \begin{bmatrix} R^{11} & R^{12} \\ \left(R^{12}\right)^{\mathrm{T}} & R^{22} \end{bmatrix} \tag{6}$$

the elements of R^{ab} $(a, b = 1, 2)$ are the correlations between the weight vectors of $\mathcal{N}_a$ and $\mathcal{N}_b$

$$R_{lm}^{ab} = \sigma_l^{(a)} \boldsymbol{W}_l^{(a)} \cdot \boldsymbol{W}_m^{(b)} \sigma_m^{(b)} \qquad \text{for} \quad 1 \leqslant l \leqslant K_a \qquad 1 \leqslant m \leqslant K_b \qquad a, b = 1, 2 \; .$$

From expressions (3), (4) and (5) we observe that in the *thermodynamic limit* ($N \to \infty$) the generalization error is determined by the overlaps $\{R_{lm}\}_{l,m=1}^{K_1+K_2}$ and the Boolean functions B_1 and B_2. The remaining details of the networks are corrections of order $\frac{1}{N}$ to this result.

Equations (3) and (5) provide a constructive method to evaluate the generalization function for any pair of 2LNs. The generalization error is simply the sum of the probabilities $P\left(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}\right)$ over the pairs $(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2})$ for which $\mathcal{N}_1$ and $\mathcal{N}_2$ yield different answers. In the case that $\mathcal{N}_1$ and $\mathcal{N}_2$ are 2LNs with tree-like architecture, a simple expression for $P\left(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}\right)$ is obtained, because by definition the elements of the correlation matrix (6) are of the form

$$R_{lm}^{ab} = \delta_{lm}\delta_{ab} + \delta_{lm}\left(1 - \delta_{ab}\right) \boldsymbol{W}_l^{(1)} \cdot \boldsymbol{W}_l^{(2)} \qquad \text{for} \quad 1 \leqslant l, m \leqslant K \quad a, b = 1, 2$$

where K is the number of hidden units. Integrating (5) we obtain

$$P\left(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}\right) = \frac{1}{2^{2K}} \prod_{l=1}^{K} \left[1 + \sigma_l^{\mu_1}\sigma_l^{\mu_2}\left(1 - 2\epsilon_l\right)\right] \tag{7}$$

with

$$\epsilon_l = \frac{1}{\pi} \arccos R_{ll}^{12} \; . \tag{8}$$

The factorization of (5) is a consequence of the fact that the input seen by each hidden unit of the 2LP is decoupled from the others, hence each hidden unit acts as an independent perceptron. ϵ_l is no more than the generalization error of the lth hidden unit of $\mathcal{N}_1$ with respect to the lth hidden unit of $\mathcal{N}_2$.

In general, when the integration is not possible, the expression for $P\left(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}\right)$ can be evaluated to any degree of precision using Kendall's expansion [10, 11]. This is an expansion for the integral (5) in powers of the matrix elements $0 < |R_{lm}| \leqslant 1$, defined as follows. Assign an integer $n_{lm} \geqslant 0$ to every pair of indices $1 \leqslant l, m \leqslant K_1 + K_2$. Denote by $\{n\}$ a set of these integers. Further, denote

$$\begin{aligned} n_l &= \sum_{m=1}^{l-1} n_{ml} + \sum_{m=l+1}^{K_1+K_2} n_{lm} \\ n &= \sum_{l=1}^{K_1+K_2} n_l \; . \end{aligned} \tag{9}$$

Kendall's expansion is a sum over all possible sets $\{n\}$:

$$P\left(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2\right) = \sum_{\{n\}} (-1)^n \left[\prod_{l<m} \frac{{R_{lm}}^{n_{lm}}}{n_{lm}!}\right] \prod_{l=1}^{K_1+K_2} G_{n_l} \; . \tag{10}$$

The first product is over all pairs l, m with $l < m$; the argument of the second product is given by

$$G_n = \begin{cases} \frac{1}{2} & \text{if } n = 0 \\ \dfrac{(2m-1)!!}{\sqrt{2\pi}\,\mathrm{i}} & \text{if } n = 2m+1 \quad (m = 0, 1, 2, \ldots) \\ 0 & \text{if } n = 2m \quad (m = 1, 2, 3, \ldots) \end{cases} \tag{11}$$

where $(2m-1)!! = \prod_{l=1}^{m}(2l-1)$.

4. A perceptron learning from a two-layered network

In this section we use the proposed method for the case where one of the networks is a single-layer perceptron with weight vector $\boldsymbol{W}_{\mathrm{P}}$ and the second a 2LN of K hidden units. We find expressions for a general 2LN as well as for some particular realizations of it like the PM, AM and CM.

In order to simplify the problem we assume that the first layer weights of the 2LN are uncorrelated [8, 9], i.e. $\boldsymbol{W}_l \cdot \boldsymbol{W}_m = \delta_{lm}$. This can be considered a 'typical case' in the sense that there is a big probability of getting K (almost) orthogonal weight vectors if they are chosen at random in the large N limit. The 2LNs with tree-like architecture, also known as non-overlapping receptive fields 2LN, are a particular case whose weights satisfy exactly the orthonormal condition.

The overlap of $\boldsymbol{W}_l$ with the perceptron weight vector is designated ρ_l

$$\rho_l = \boldsymbol{W}_l \cdot \boldsymbol{W}_{\mathrm{P}} .$$

Denoting by $\boldsymbol{\sigma}$ the IR of the 2LN and by y_{P} the output of the perceptron, and introducing

$$z_l = y_{\mathrm{P}}\, \sigma_l \rho_l$$

the correlation matrix (6) takes the form

$$R = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 & z_1 \\ 0 & 1 & 0 & \cdots & \cdots & 0 & z_2 \\ 0 & 0 & 1 & \cdots & \cdots & 0 & z_3 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & 1 & z_K \\ z_1 & z_2 & z_3 & \cdots & \cdots & z_K & 1 \end{bmatrix} . \tag{12}$$

We have to take into account only the non-diagonal elements that are different from zero in expansion (10). Therefore the set $\{n\}$ is composed, in this case, only by elements of the form $n_{l,K+1}$, i.e.

$$\{n\} = \{m_l\}_{l=1}^{K} \qquad \text{with} \quad m_l = n_{l,K+1} .$$

Equation (9) becomes

$$n_l = \begin{cases} m_l & \text{if} \quad l \neq K+1 \\ \sum_{l=1}^{K} m_l & \text{if} \quad l = K+1 \end{cases}$$

$$n = 2\sum_{l=1}^{K} m_l$$

and expansion (10) can be expressed as

$$P\left(\boldsymbol{\sigma}^{\mu_1}, \boldsymbol{\sigma}^{\mu_2}\right) = \sum_{m_1,\ldots,m_K=0}^{\infty} G_{\Sigma m_l} \prod_{l=1}^{K} \frac{z_l^{m_l} G_{m_l}}{m_l!} .$$

Only terms with an odd number of z raised to an odd power can appear in this expansion because $G_n \neq 0$ only if n is odd or zero. Hence expansion (10) can be written as

$$2^K P\left(y_{\mathrm{P}}, \boldsymbol{\sigma}^{\mu}\right) = \tfrac{1}{2} + \sum_{l=1}^{K} f_1\left(z_l\right) + \sum_{l_1<l_2<l_3} f_3\left(z_{l_1}, z_{l_2}, z_{l_3}\right) + \cdots + \sum_{l_1<\cdots<l_{K^*}} f_{K^*}\left(z_{l_1}, \ldots, z_{l_{K^*}}\right)$$

where

$$f_{2m+1}(z_1,\ldots,z_{2m+1}) = \frac{1}{\pi}\left(\frac{-2}{\pi}\right)^m \sum_{t_1,\ldots,t_{2m+1}=0}^{\infty} \left(2\sum t_i + 2m - 1\right)!! \prod_{i=1}^{2m+1} \frac{z_i^{2t_i+1}(2t_i-1)!!}{(2t_i+1)!}$$

with $K^* = K$ if K is odd and $K^* = K-1$ if K is even. In particular, it can be shown that $f_1(z)$ is just the series expansion of $\frac{1}{\pi}\arcsin(z)$. Using

$$f_{2m+1}\left(y_{\mathrm{P}}\sigma_{l_1}\rho_{l_1},\ldots,y_{\mathrm{P}}\sigma_{l_{2m+1}}\rho_{l_{2m+1}}\right) = y_{\mathrm{P}}\left[\prod_{i=1}^{2m+1}\sigma_{l_i}\right] f_{2m+1}\left(\rho_{l_1},\ldots,\rho_{l_{2m+1}}\right)$$

the generalization error becomes

$$\epsilon_{\mathrm{g}} = \tfrac{1}{2} - \sum_{l_1=1}^{K} C_{l_1} f_1(\rho_l) - \sum_{l_1<l_2<l_3} C_{l_1 l_2 l_3} f_3\left(\rho_{l_1},\rho_{l_2},\rho_{l_3}\right) - \cdots$$
$$\cdots - \sum_{l_1<\ldots<l_{K^*}} C_{l_1\ldots l_K} f_{K^*}\left(\rho_{l_1},\ldots,\rho_{l_{K^*}}\right) \tag{13}$$

where $C_{l_1\ldots l_m}$ is the correlation of hidden units $l_1\ldots l_m$ with the output

$$C_{l_1\ldots l_m} = \frac{1}{2^K}\sum_{\mu=1}^{2^K} B(\boldsymbol{\sigma}^\mu)\prod_{i=1}^{m}\sigma_{l_i}^{\mu} \tag{14}$$

and $\{\boldsymbol{\sigma}^\mu\}_{\mu=1}^{2^K}$ is the set of all IRs for K hidden units.

It is interesting to observe that (13) is basically a sum of products, each of the form $C_{l_1\ldots l_m} f_m$ where the functions f_m depend only on the 2LN weight vectors and the perceptron weight vector, while the correlation coefficients C_m are completely determined by the Boolean function that maps the hidden layer to the output. In some cases, as we will see below, it is possible to draw many conclusions from the correlation coefficients without calculating the f_m explicitly.

In many cases all hidden units play equivalent roles in the hidden layer to output map B (like the PM, AM and CM). Hence the correlation depends only on the number of hidden units and not on the particular ones chosen. This motivates the following definition. We say that a 2LN is *symmetric* if

$$C_{l_1\ldots l_m} = \hat{C}_m$$

for all $l_1 < \cdots < l_m$ with $m = 1, 3, \ldots, K^*$.

Let us now evaluate (14) for some 2LN. In the case of the PM it is easy to show that

$$\hat{C}_l = \begin{cases} 1 & \text{if } l = K \\ 0 & \text{otherwise}. \end{cases} \tag{15}$$

Therefore the generalization error of any perceptron that tries to imitate a PM of an even number of hidden units is always $\frac{1}{2}$ because only the correlations of odd numbers of hidden units enter in (13). This result was expected because for every PM with an even number of hidden units we have that $y_{\mathrm{PM}}(\boldsymbol{x}) = y_{\mathrm{PM}}(-\boldsymbol{x})$ while for every SLP we know that $y_{\mathrm{P}}(\boldsymbol{x}) = -y_{\mathrm{P}}(-\boldsymbol{x})$ for any input $\boldsymbol{x}$. For a PM with an odd number of hidden units it is necessary to calculate only one term in the expansion (13), i.e. it is of the form $\epsilon_{\mathrm{g}} = \frac{1}{2} - f_K$.

Evaluation of (14) in the case of the AM is also immediate:

$$\hat{C}_l = \frac{1}{2^{K-1}}.$$

In the case of the CM we get, after some algebra and assuming that K is odd, that

$$\hat{C}_{2m+1} = \frac{(-1)^m}{2^{K-1}} \sum_{n=0}^{m} \left[(-1)^n \begin{pmatrix} 2m+1 \\ n+m+1 \end{pmatrix} \sum_{u=-n}^{n} \begin{pmatrix} K-2m-1 \\ u-m+\frac{K-1}{2} \end{pmatrix} \right]. \tag{16}$$

When the 2LN is a ruler machine, we can assume without loss of generality that its output is determined by the hidden unit 1. In this case we have that all the correlation coefficients (14) vanish except $C_1 = 1$. Therefore (13) yields $\epsilon_g = \frac{1}{\pi} \arccos(\rho_1)$, which is the well known expression for the generalization error between two SLPs.

4.1. The optimal perceptron

We turn now to address the issue of finding the minimal generalization error of an SLP that tries to imitate a symmetric 2LN. We consider first the general case and then we analyse the cases where the 2LN is a PM, CM and AM.

We have to minimize ϵ_g with respect to $\{\rho_l\}_{l=1}^K$ under the constraint $\sum_{l=1}^K \rho_l^2 \leqslant 1$. This can be done by minimizing

$$h = \epsilon_g + \lambda \left(1 - \rho_\perp^2 - \sum_{l=1}^{K} \rho_l^2 \right)$$

with respect to $\{\rho_l\}_{l=1}^K$; $\rho_\perp$, the projection of $\boldsymbol{W}_P$ on the subspace orthonormal to the 2LN weight vectors and the Lagrange multiplier λ. The condition $\partial h / \partial \rho_\perp = 0$ implies that $\rho_\perp = 0$, which means that the weight vector of the optimal perceptron must be contained in the subspace spanned by $\{\boldsymbol{W}_l\}_{l=1}^K$; i.e.

$$\sum_{l=1}^{K} \rho_l^2 = 1 \,. \tag{17}$$

Finally, taking the derivative with respect to ρ_l leads to

$$\sum_m \hat{C}_m \sum_{l_1,\ldots,l_m} \frac{\partial f_m}{\partial \rho_l} \left(\rho_{l_1}, \ldots, \rho_{l_m} \right) = 2\lambda \rho_l \qquad \text{for} \quad 1 \leqslant l \leqslant K^* \,. \tag{18}$$

Equations (17) and (18) are valid for any set of overlaps, ρ_l, except for the cases where the perceptron coincides with one of the first layer perceptrons of the teacher, because the norm of the gradient of ϵ_g diverges. Thus, in order to obtain the minimum of ϵ_g, not only must ϵ_g be evaluated at all solutions of (17) and (18) but also at these points. In the case where the 2LN is a CM, PM and AM the minima can be found explicitly.

4.1.1. Committee machine. Let consider the particular realization of a 2LN, where the mapping from the hidden layer of units to the output unit is made by a CM. Since the correlations $\hat{C}_m$ have the property that $\text{sign}[\hat{C}_m f_m(\rho_{l_1}, \ldots, \rho_{l_m})] = \text{sign}\left[\prod_{i=1}^m \rho_{l_i}\right]$ we have that the overlaps that lead to the minimum of the generalization error ϵ_g must be positive. Moreover, if we consider $\partial C_m f_m / \partial \rho_l$ as a function of ρ_l we observe that it is an even function and it is a monotonic function for $\rho_l > 0$. Hence the right-hand side of equation (18) is an even function of ρ_l and monotonic increasing for $\rho_l > 0$. On the other hand the left-hand side of equation (18) is a monotonic odd function of ρ_l. Therefore each of the K equations of (18) with $l = 1, \ldots, K$ possesses at most two solutions. By the symmetry of the problem we have that if $\left(\rho_1^\star, \ldots, \rho_K^\star\right)$ is a solution, then any permutation of it will still be a solution. So, the solutions of the set of equations (18) have the particularity that each of the overlaps $\rho_l^\star$, can take at most two values, say $\alpha > 0$ and $\beta > 0$. The constraint (17)

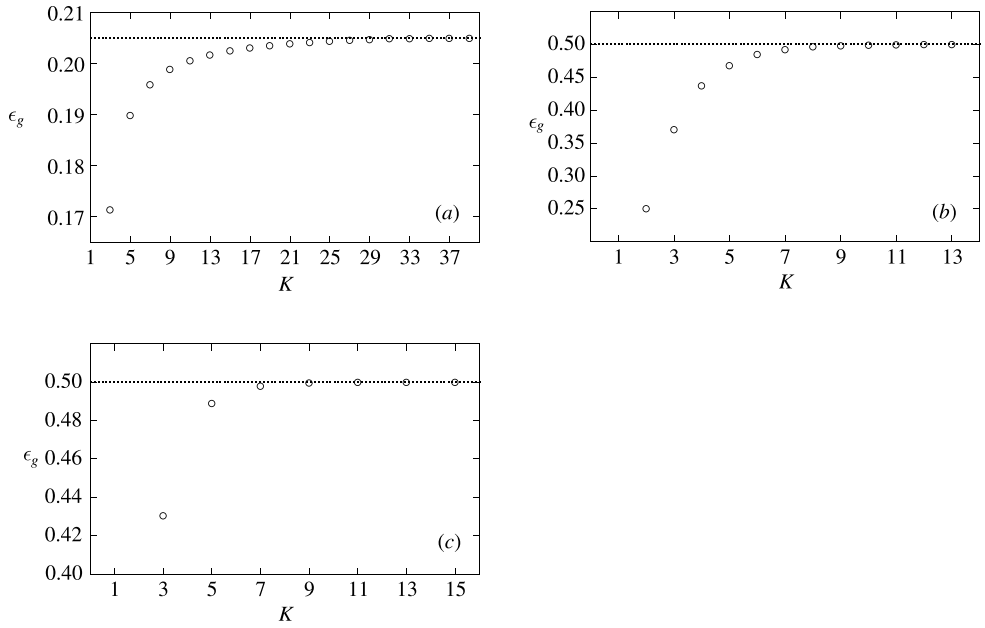

Figure 1. Minimal generalization error of a perceptron that tries to imitate different two-layered networks. The minimal *generalization error*, $\epsilon_{\rm g}$ is presented for the following realizations of the 2LN; the *committee machine* in (*a*), the *and machine* in (*b*), and the *parity machine* in (*c*). In (*a*) and (*c*) we consider only odd numbers of hidden units, K. The open circles indicate the value of $\epsilon_{\rm g}$ for the different K. The dotted lines indicate the value of $\lim_{K\to\infty}\epsilon_{\rm g}^{\rm min}(K)$ in each case.

imposes a new restriction; the number of α should be fixed. In addition, it can be easily checked that

$$\rho_l = \frac{1}{\sqrt{K}} \qquad \text{for} \quad 1 \leqslant l \leqslant K$$

satisfies equations (17) and (18) and therefore the solution is unique. Thus we have found that the optimal perceptron is the one that is equidistant from all the perceptrons composing the first layer weights of the CM:

$$\epsilon_{\rm g} = \tfrac{1}{2} - \sum_{l=1,3,\ldots,K^*} \hat{C}_l f_l \tag{19}$$

where

$$f_{2m+1} = \frac{(-1)^m}{\pi\sqrt{K}}\left[\frac{2}{\pi K}\right]^m \sum_{M=0}^{\infty} \frac{[2(M+m)-1]!!}{K^M} \sum_{\substack{\{s_i\}_{i=1}^{2m+1} \\ \sum s_i = M}} \prod_{i=1}^{2m+1} [(2s_i)!(2s_i+1)]^{-1} \tag{20}$$

and the correlations $\hat{C}_l$ are given by (16). The minimal generalization error obtained for different values of K is presented in figure 1(*a*).

Let us now consider the case of a large number of hidden units; we assume that $1 \ll K \ll N$. Expanding expression (20) at the leading term in $1/\sqrt{K}$, we obtain

$$f_{2m+1} = \frac{(-1)^m}{\pi\sqrt{K}}\left[\frac{2}{\pi K}\right]^m (2m-1)!! \,. \tag{21}$$

The correlation coefficients (16) of the CM in the large K limit become

$$\hat{C}_{2m+1} = \sqrt{\frac{2}{\pi K}} \frac{(-1)^m}{K^m} (2m-1)!!\,.$$

Inserting the last two equations into (19) and using Stirling's approximation we get that the minimal generalization error for the optimal perceptron that learns from a CM of infinite hidden units is

$$\epsilon_g = \frac{1}{2} - \sqrt{\frac{2}{\pi^3}} \sum_n \binom{2n}{n} \frac{(2\pi)^{-n}}{(2n+1)} = \arccos\left(\sqrt{\frac{\pi}{2}}\right) \cong 0.206\,.$$

This value is the same† as that of the minimum generalization error obtained in [8, 9] for a CM machine learning from another CM in the permutation symmetric phase where the CM effectively behaves as a perceptron. Since the treatment of this work is equivalent to the case of an infinite number of examples and zero temperature of [8, 9], we conclude that in the limit of a high number of examples, there is no permutation symmetry breaking. In addition, for the case of a perceptron learning from a CM of three hidden units, our result not only agrees with the limit of a high number of examples obtained in [7], but also justifies the permutation symmetry of the order parameter assumed in that work.

4.1.2. And machine and parity machine. In the case when the teacher is a parity machine of odd number of hidden units the generalization (13) error is reduced to only two terms; $\epsilon_g = \frac{1}{2} - f_K(\rho_1, \rho_2, \ldots, \rho_K)$ because there is only one non-vanishing correlation coefficient (15). Using similar arguments to those used in the previous section we obtain that the optimal perceptron that learns from a PM of $2m+1$ hidden units must satisfy: (a) $|\rho_l| = \frac{1}{\sqrt{K}}$ for $1 \leqslant l \leqslant K$ and (b) $\mathrm{sign}\left(\prod \rho_l\right) = 1$ if m is even and -1 if m is odd. That is, the generalization error ϵ_g possesses $2^{\frac{K+1}{2}}$ minima whose values are

$$\epsilon_g = \frac{1}{2} - f_K\left(\frac{1}{\sqrt{K}}\right)$$

where f_K is given by equation (21). In figure 1(b) we present the value of the generalization error ϵ_g for the optimal perceptron that learns from a PM. The limit of a large number of hidden units of the generalization error is

$$\epsilon_g = \frac{1}{2} - \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{\pi}}\right)^{K+1}.$$

In the case of the AM there is only one minimum given by $\rho_l = 1/\sqrt{K}$ for $1 \leqslant l \leqslant K$ and the asymptotic value is

$$\epsilon_g = \frac{1}{2} - \frac{A}{2^K}$$

where $A = \frac{2}{\pi} \sum_l \frac{(l-2)!!}{l!} \cong 0.76$. This result reflects the fact that a fraction of the input space for which $y_{AM} = +1$ goes to zero as $1/2^K$ while for the SLP we have that $y_P = +1$ for half of the input space.

† I would like to thank the referee for pointing this out.

5. Similarity between committee and parity machines

We now consider the case of two different 2LNs with tree-like architecture. We study the case of a PM and a CM with three hidden units. We show that they implement different sets of Boolean functions and we find the minimal generalization error.

From equations (3) and (7) we obtain that the fraction of the input space for which a PM and a CM disagree is

$$\epsilon_g = \tfrac{3}{4} - \tfrac{1}{2}(\epsilon_1 + \epsilon_2 + \epsilon_3) + (\epsilon_1\epsilon_2 + \epsilon_2\epsilon_3 + \epsilon_1\epsilon_3) - 2\epsilon_1\epsilon_2\epsilon_3 \tag{22}$$

where ϵ_l, $l = 1, 2, 3$ is the generalization error (8) of perceptrons receiving the same input. The generalization error (22) is minimized at $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1$; $\epsilon_1 = 1$, $\epsilon_2 = \epsilon_3 = 0$ and all possible permutations, yielding $\epsilon_g = \frac{1}{4}$. Note that there is a solution that satisfies permutation symmetry in the sense that the overlap of three pairs of perceptrons is the same while there are another three for this symmetry does not hold. Since the generalization error never vanishes, a PM and a CM of a 2LN with three hidden units and tree-like architecture will always implement different Boolean functions whatever their first layer weights are. It is possible to show that this result is still valid for any number of hidden units K.

6. Conclusion

We have extended a previously proposed method [5, 6] for calculating the *generalization error* of two two-layered networks. This technique consists basically of making a list of all the pairs of *internal representations* that yield different outputs for the two networks. The fraction of the input space that gives rise to such a pair is calculated. The sum of fractions for all such pairs is the generalization error.

We have applied this method for the case of a single-layer perceptron, the 'student', who tries to imitate a two-layered network, the 'teacher'. We found that the generalization error between them depends only on the overlaps of the weight vector of the perceptron with each of the weight vectors of the two-layered network and on the correlation of each hidden unit with the output unit.

We studied the generalization error as a function of the perceptron weights; in particular we focused on the perceptron that minimizes the generalization error for a given two-layered teacher network. It was found that the optimal student's weight vector belongs to the subspace spanned by the weight vectors of the teacher network. In the case when all the hidden units have the same correlation with the output unit, we found that the overlap of the weight vector of the student with each of the teacher weight vectors must be the same.

We obtained explicit expressions for the cases when the teacher is a *committee machine*, a *parity machine* and an *and machine*. We obtained that for a committee machine of any (odd) number of hidden units there exists a perceptron which is able to give a correct answer for almost 80% of the inputs.

The computational capabilities of the committee and parity machines of tree-like architecture were compared. We found that they implement a disjoint set of Boolean functions.

The results obtained in this work are also valid if we consider continuous inputs whose components x_i possess a symmetric density distribution around zero.

Acknowledgments

I would like to thank E Domany for stimulating discussions and encouragement as well as for valuable comments regarding the manuscript. Also I would like to thank Ido Kanter for many motivating discussions.

References

[1] Watkin T H L, Rau A and Biehl M 1993 *Rev. Mod. Phys.* **6** 499
[2] Seung H S, Sompolinsky H and Tishby N 1992 *Phys. Rev.* A **45** 6056
[3] Watkin T H L and Rau A 1992 *Phys. Rev.* A **45** 4102
[4] Kinouchi O and Caticha N 1993 *J. Phys. A: Math. Gen.* **25** 6161
[5] Priel A, Blatt M G, Grossman T, Domany E and Kanter I 1994 *Phys. Rev.* E **50** 577
[6] Blatt M G, Domany E and Kanter I 1995 *Int. J. Neural Syst.* **6** 225
[7] Scharnagl A 1992 Perzeptron lernt Komitee-Maschine *Diplomarbeit* Institut für Theoretische Physik, Justus-Liebig-Universität Gießen
[8] Kang K, Oh J-H, Kwon C and Park Y 1994 *Phys. Rev.* E **48** 4805
[9] Shwartze H and Hertz J 1993 *Europhys. Lett.* **21** 785
[10] Kendall M G, Stuart A and Ord J K 1987 *Kendall's Advanced Theory of Statistics* vol 1, 5th edn (London: Griffin) p 484
[11] Saad D 1994 *J. Phys. A: Math. Gen.* **27** 2719