Learning by a population of perceptrons

Kukjin Kang and Jong-Hoon Oh
Department of Physics, Pohang University of Science and Technology, Pohang, Kyongbuk 790-784, Korea

Chulan Kwon
Department of Physics, Myong Ji University, Yongin, Kyonggi 449-728, Korea

(Received 29 August 1996)

Learning by examples of a population of neural networks is studied in a statistical physics framework. A population of single-layer perceptrons learns from a two-layer neural network. Each member is trained independently either from the same or from different example sets. The outputs of multiple networks are combined by majority vote. We calculate the generalization curve of the group decision of the perceptrons with both discrete and continuous weights. We find an interesting nonmonotonic learning curve for the case of discrete weights, indicating that majority vote shows optimal performance when the size of the example set is finite. [S1063-651X(97)02503-8]

PACS number(s): 87.10+e, 05.50+q, 64.60.Cn

Recently, the process of combining multiple networks has been used widely within the neural network community to obtain an optimal generalization capability [1–6]. Published studies fall into two classes. One is the ''expert'' approach, where the problem is divided into manageable sizes for several subnetworks (experts) and each expert learns locally from a part of the problem domain. The outputs from the experts are combined using human expertise [4] or by independently trained gating networks [5]. The other class is the ''ensemble'' approach [1] in which we generate an ensemble of networks independently trained for the whole problem and the outputs of each network are combined with an appropriate weighting. The main difference between the two approaches is that in the first instance each neural network manages the global problem domain, while in the second, it is specialized for the local tasks.

Whereas these approaches are gaining more popularity in various applications, it is difficult to find theoretical studies that have analyzed their validity and performance. In this paper we will address some fundamental issues mainly related to the ensemble approach through the statistical mechanics formulation [7–11]. The key issues in the multiple neural network approach are how the outputs of the various subnetworks should be combined to give the best generalization performance and how to make the best use of a limited data set. Perrone and Cooper [1] proposed a method for calculating optimal weighting factors for an ensemble of neural networks. Wolpert devised a method to train a supervisor network to give the weighting factors [2]. We have studied an analytically soluble model of the majority vote of perceptrons learning dichotomy rules. This corresponds to the ''basic ensemble method,'' described by Perrone and Cooper, where the weightings of the networks are equal. It is clearly not the best choice in a real situation, but it allowed us to understand when the ensemble approach could be useful and where the weak points existed. In this paper we will focus on the relation between generalization performance and the size of training sets. We find an interesting nonmonotonic learning curve that suggests that the ensemble approach is particularly useful with a limited number of examples.

We first consider a situation of unrealizable learning by a population of perceptrons. A population of single-layer perceptrons is independently trained from examples presented by a two-layer teacher network called a committee machine. We consider cases where the training example set is either the same, or different, for each perceptron.

An individual perceptron (voter) maps the input vector $\boldsymbol{S}=\{S_i, \ldots, S_N\}$ to the output σ as

$$\sigma(\boldsymbol{W};\boldsymbol{S})=g\left(\frac{1}{\sqrt{N}}\sum_{i}^{N} W_i S_i\right), \tag{1}$$

where $\boldsymbol{W}$ is a set of the synaptic weights whose component W_i is a weight from the ith input node to the output node. We consider the transfer function $g(x)=\mathrm{sgn}(x)$.

The examples are randomly generated by a committee machine teacher with N input nodes and M hidden nodes. The network maps an input vector $\boldsymbol{S}$ to an output σ given by

$$\sigma(\boldsymbol{V};\boldsymbol{S})=g_2\left[M^{-1/2}\sum_{j}^{M} g_1\left(\frac{1}{\sqrt{N}}\boldsymbol{V}_j\cdot\boldsymbol{S}\right)\right], \tag{2}$$

where $g_1(x), g_2(x)$ are transfer functions of the hidden nodes and the output node, respectively. We also consider threshold units $g_1(x)=g_2(x)=\mathrm{sgn}(x)$. $\boldsymbol{V}_j$ is a vector representation for the synaptic weights whose component V_{ji} is a weight from the ith input node to the jth hidden node.

The energy of the system is defined as the difference between the output of the teacher network and the output of each *individual* perceptron,

$$E=\sum_{l=1}^{P} \epsilon(\boldsymbol{W};\boldsymbol{S}^l), \tag{3}$$

$$\epsilon(\boldsymbol{W};\boldsymbol{S}^l)=\Theta(-\sigma(\mathbf{V};\boldsymbol{S}^l)\sigma(\boldsymbol{W};\boldsymbol{S}^l)), \tag{4}$$

where $\Theta(x)$ is the Heaviside step function and P is the number of training examples. Each component of the input S_i^l is randomly drawn from the Gaussian distribution with variance unity.

The stochastic learning algorithm, used for each perceptron, leads, after a long time, to a Gibbs distribution of the weights as

$$P(\boldsymbol{W})=Z^{-1}e^{-\beta E(\boldsymbol{W})}, \tag{5}$$

where $\beta=1/T$ is the inverse temperature and the normalization factor Z is the partition function,

$$\int d\mu(\boldsymbol{W})e^{-\beta E(\boldsymbol{W})}. \tag{6}$$

The prior distribution of weights $d\mu(\boldsymbol{W})$ contains appropriate constraints for weights. In this paper we consider both binary weights and continuous weights with spherical constraints.

The free energy F is given by

$$-\beta F=\langle\langle \ln Z\rangle\rangle \tag{7}$$

where $\langle\langle\ \rangle\rangle=\int\Pi_l d\mu(\boldsymbol{S}^l)$ denotes the quenched average over possible example sets. We use the replica trick

$$\langle\langle \ln Z\rangle\rangle=\lim_{n\to 0}\frac{\langle\langle Z^n\rangle\rangle-1}{n}, \tag{8}$$

which has already been applied successfully to the problems of storage capacity and learning from examples [8–11].

The performance of the network is measured by the generalization error. We consider a majority vote of m perceptrons $\{\boldsymbol{W}_1,\ldots,\boldsymbol{W}_m\}$ voting for the answer

$$\sigma_g(\{\boldsymbol{W}\};\boldsymbol{S})=\mathrm{sgn}\left(\sum_a^m \sigma(\boldsymbol{W}_a;\boldsymbol{S})\right). \tag{9}$$

The average generalization error is given by

$$\epsilon_g(T,P)=\left\langle\left\langle\left\langle\int d\mu(\boldsymbol{S})\Theta(-\sigma(\boldsymbol{V};\boldsymbol{S})\sigma_g(\{\boldsymbol{W}\};\boldsymbol{S}))\right\rangle_T\right\rangle\right\rangle, \tag{10}$$

where $\langle\rangle_T$ is the thermal average over the distribution $P(\boldsymbol{W})$.

For convenience, we divide the free energy into two parts, G_0 and G_r,

$$-\beta F=N(G_0+\alpha G_r), \tag{11}$$

where $\alpha=P/N$. G_0 and G_r are of the order unity. Note that P is the number of examples per student, not the total number of examples. When a different example set is given to each student, the total number of examples is mP. In the thermodynamic limit, $N\to\infty$, the free energy can be written as a function of several order parameters. The order parameters are defined as

$$R_{aj}^{\sigma}=\frac{1}{N}\boldsymbol{W}_a^{\sigma}\cdot\boldsymbol{V}_j, \tag{12}$$

$$q_a^{\sigma\rho}=\frac{1}{N}\boldsymbol{W}_a^{\sigma}\cdot\boldsymbol{W}_a^{\rho}, \tag{13}$$

where σ,ρ are replica indices. The replica symmetric ansatz is written as

$$R_{aj}^{\sigma}=r, \tag{14}$$

$$q_a^{\sigma\rho}=\delta_{\sigma\rho}+(1-\delta_{\sigma\rho})q. \tag{15}$$

We also assume that overlaps between the student and weights of the teacher connected to different hidden nodes are equal. For $M\gg 1$, we have

$$G_r=2\int Dt\ H\left(\frac{-\sqrt{2/\pi}R}{\sqrt{q-(2/\pi)R^2}}t\right)\ln\left[1+(e^{-\beta}-1)H\left(\sqrt{\frac{q}{1-q}}t\right)\right], \tag{16}$$

$$G_0=\begin{cases}\dfrac{1}{2}q\hat{q}-R\hat{R}-\dfrac{1}{2}\hat{q}+\displaystyle\int Dt\ \ln[2\ \cosh(\sqrt{\hat{q}+\hat{R}^2}t)] & \text{(binary weight)}\\ \dfrac{q-R^2}{2(1-q)}+\dfrac{1}{2}\ln(1-q) & \text{(continuous weight)},\end{cases} \tag{17}$$

where $R\equiv\sqrt{M}r$.

The generalization error depends upon both the generalization capability of individual perceptrons and the correlation among them. To calculate the generalization performance of the group decision, we need to introduce a new order parameter. C_{ab} is defined as the overlap of the ath and bth student perceptrons:

$$C_{ab}=\frac{1}{N}\boldsymbol{W}_a\cdot\boldsymbol{W}_b=C. \tag{18}$$

When $C\leqslant O(1/m)$, the generalization error is written as

$$\epsilon_g=\frac{1}{\pi}\arccos\left[\frac{2}{\pi}\frac{\sqrt{m}R}{\sqrt{1+(2/\pi)(m-1)C}}\right], \tag{19}$$

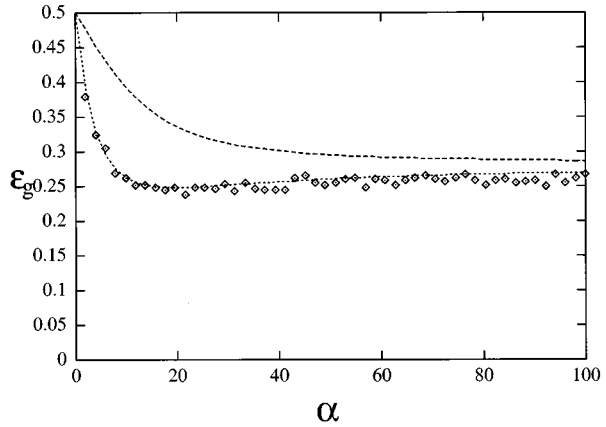

FIG. 1. The dotted line shows the generalization curves of majority vote by a population of binary perceptrons trained with the same example sets at $T=5$. For comparison, the dashed line depicts the learning curve of a single perceptron. The dots show the results of the Monte Carlo simulation, averaged over 10 independent runs where $N=M=m=51$.

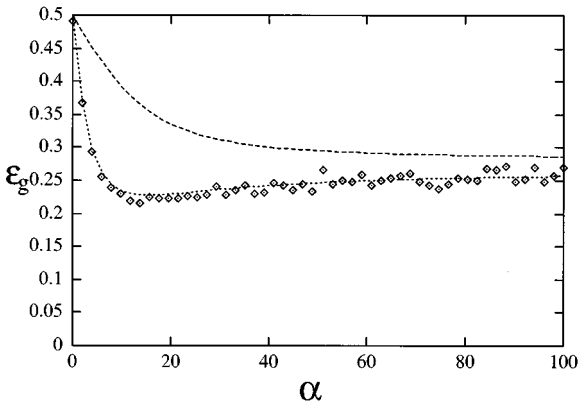

FIG. 2. The dotted line shows the generalization curves of majority vote by a population of the binary perceptrons trained with different example sets at $T=5$. For comparison, the dashed line depicts the learning curve of a single perceptron. The dots show the results of the Monte Carlo simulation, averaged over 10 independent runs where $N=M=m=51$.

and for $C>O(1/m)$,

$$\epsilon_g=\frac{1}{\pi}\arccos\left(\frac{\sqrt{2/\pi}R}{\sqrt{C}}\right). \tag{20}$$

The value of C can be determined from the saddle point equation of the free energy. We find that the overlap between different perceptrons is the same as the overlap between two replicas when the perceptrons are trained from the same example set, that is,

$$C=q. \tag{21}$$

When they are trained from independently collected training sets, we have

$$C=\begin{cases}\displaystyle\int Dx\left[\int Dy\ \tanh(\hat{R}x+\sqrt{q}y)\right]^2 & \text{for binary weights}\\ R^2 & \text{for continuous weights.}\end{cases} \tag{22}$$

By substituting the values of the order parameters obtained from the saddle-point equations into Eq. (20), we get the generalization errors for these two cases. The two resulting learning curves for the networks with binary weights are plotted in Figs. 1 and 2, together with the learning curve of a single perceptron.

Here we find an interesting nonmonotonic learning curve. The generalization error of the single perceptron decreases monotonically as a function of α. When the number of examples is small, generalization performance of the group decision by majority vote is much better than that of a single perceptron. However, generalization error reaches a minimum value at a certain value of α, and then increases again. When the number of examples approaches infinity, the generalization error converges to the same value as that of the single perceptron. This result can be interpreted as follows. When α is small, the generalization error is mainly controlled by the order parameter R, which increases with α. This order parameter C measures similarity between different students. For small α, the perceptrons have diverse configurations, and the group decision can show much better performance than a typical member of the population. As α increases further, the order parameter C increases, and the perceptrons are located in or near the shrunken version space. If they are too similar to each other, they cannot exploit the advantage of the group decision. The group decision loses the advantage over an individual student, and the generalization performance of the algorithm deteriorates. When α approaches infinity, C approaches one, and the generalization error of the majority vote approaches that of a single perceptron.

With a motivation similar to that of the ''expert'' approach, we consider the case where a different training set is given to each perceptron. Comparing Eq. (21) and Eq. (22), we find that the correlation C among the perceptrons is smaller when the perceptrons learn from different training sets. Consequently, the generalization performance is better. We need m times more examples, but this is not practical in most cases. When the total number of examples is fixed, the size of the training set available for each perceptron is $1/m$ of all of the examples, and this can lead to poor performance.

The generalization error ϵ_g of Eq. (19) is controlled by the order parameters R and C. It is small when R is large and C is small. When the number of examples is of $O(N)$, R is too small to expect much improvement from the group decision. Only when the total number of examples is of $O(\sqrt{m}N)$, the effect of C dominates and we obtain better results by dividing examples. Note that we have considered the case $m\gg 1$. It will be interesting to see how this picture changes when m is small.

It is desirable to analyze replica symmetry breaking (RSB) solutions for the study of unrealizable learning. In Fig. 3, we plot the line where the entropy of the replica symmetric (RS) solution vanishes. Below this line, the RS solution is no longer valid. Our one-step RSB calculation shows that the one-step RSB solution takes over the RS solution below this line. All the thermodynamic quantities are frozen below the transition line. The generalization error of

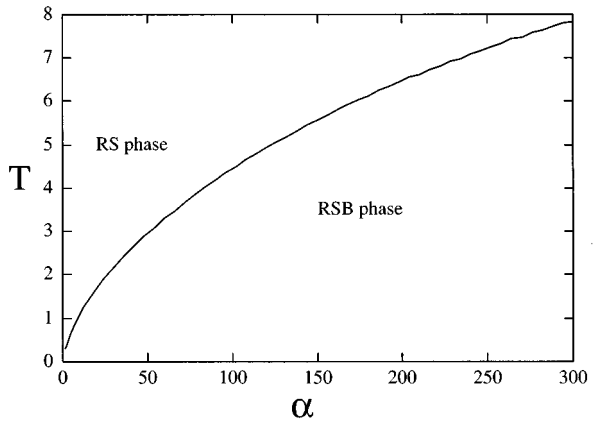

FIG. 3. The phase diagram and the zero entropy line in the α-T plane.

the RSB solution at a certain value of α is the same as that of the RS solution at the transition line. This scenario has been shown previously in studies of networks with binary weights [9,12]. However, the generalization error obtained from the RSB solution is numerically very close to that of the RS solution. At modestly high temperature, the optimal α, which gives the best generalization error, exists in the RS phase.

In the replica symmetric phase, voting with equal weighting can be a good choice for combining multiple outputs of the voters. When replica symmetry is broken, however, the overlaps among the perceptrons have different values, and majority votes with equal weighting is not the optimal strategy. The algorithm of Perrone and Cooper [1] may be a useful solution in this situation.

We also considered the case of majority voting by perceptrons with continuous weights. Although we did not observe the nonmonotonic learning curve, we again concluded that a group decision is useful only when the sizes of the examples are relatively small.

When the perceptrons learn an unrealizable task, the generalization error converges to a nonzero constant: $\epsilon_0 \equiv (1/\pi)\cos^{-1}(\sqrt{2/\pi})$. When each perceptron learns from the same example set, the asymptotic behavior is written as $\epsilon_g - \epsilon_0 \simeq T/2\alpha$, whereas a single perceptron has asymptotics $\epsilon_g - \epsilon_0 \simeq T/\alpha$. This means that we need only half the examples to achieve the same performance when we use multiple networks. When α is small, this can be a big advantage, but it does not make much difference when α is large and the generalization error is close to the limiting value ϵ_0.

We note that the group decision is less sensitive to the temperature. As the weight space of the perceptron is spherically symmetric, the group decision usually reduces the effect of noise.

When each perceptron learns from a different example set, the generalization error of the group decision decreases in the regime where $\alpha \sim 1/\sqrt{m}$ or the total number of examples is $O(\sqrt{m}N)$. Again, we understand that it is not desirable to divide the examples equally when the total number of examples is $O(N)$. Here α is $O(1/m)$ and the generalization error does not decrease in this region.

We have studied learning by a population of neural networks and calculated the generalization performance of the group decision by majority vote. For perceptrons with binary weight, generalization error reaches a minimum at a certain number of examples and increases again thereafter. Thus, the performance of an ensemble approach is reduced when the number of training examples is too large. A large training set restricts diversity of students and harms the generalization performance of the group decision. An education that is too standardized sometimes ruins the creativity of students. The study of perceptrons with continuous weights also implies that majority vote is more useful with small example sets.

This result yields a useful direction for selecting an approach to multiple neural networks. In the expert approach, the similarity between the experts is not important because each expert is trained from local examples. When the training example size is small, however, each expert learns from even smaller examples and the generalization capability of each network cannot be fully utilized. In learning of the ''mixture of experts'' [5], the gating networks need a certain minimum example size for effective partition of the input space. The expert approach may be more useful, therefore, when a sufficient number of training examples are available, whereas the ensemble approach is strong with a limited training example size. It would be useful to derive an algorithm to unify the two approaches. In addition, it would be interesting to compare the performance of various algorithms in this situation of a two-layer network teacher with single-layer perceptron students. We expect that the theoretical issues in more sophisticated algorithms such as the ''mixture of experts'' [5] and the ''stacked generalization'' [2] may be treated systematically with the same settings.

This work was partially supported by the Basic Science Special Program of POSTECH and the Korea Ministry of Education through the POSTECH Basic Science Research Institute. It was also supported by ''the non-directed fund'' from the Korea Research Foundation.

[1] M. P. Perrone and L. N. Cooper, in *Neural Networks for Speech and Image Processing*, edited by R. J. Mammone (Chapman-Hill, London, 1993).

[2] D. Wolpert, Neural Networks **5**, 241 (1992).

[3] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, Neural Comput. **6**, 1289 (1994).

[4] J. B. Hampshire II and A. Waibel, Adv. Neural Information Processing Syst. **2**, 203 (1990).

[5] R. A. Jacobs, N. I. Jordan, S. J. Nolwan, and G. E. Hinton, Neural Comput. **3**, 79 (1991).

[6] M. Kearns and H. S. Seung, *Proceedings of 6th Annual ACM Conference on Computational Learning Theory* (ACM, New York, 1993), pp. 101–110.

[7] For reviews, see, for example, M. Opper and W. Kinzel, in

Physics of Neural Networks, edited by J. L. Van Hemmen, E. Domany, and K. Schulten (Springer-Verlag, Berlin, in press).

[8] E. Gardner, Europhys. Lett. **4**, 481 (1987); J. Phys. A **21**, 257 (1988).

[9] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).

[10] K. Kang, J.-H. Oh, C. Kwon, and Y. Park, Phys. Rev. E **48**, 4805 (1993).

[11] K. Kang, J.-H. Oh, C. Kwon, and Y. Park, Phys. Rev. E **54**, 1811 (1996).

[12] W. Krauth and M. Mézard, J. Phys. (Paris) **50**, 3057 (1989).