NEUCOM 225

Backpropagation and stochastic gradient descent method

Shun-ichi Amari

Dept. of Mathematical Engineering, University of Tokyo, Bunkyo-ku, Hongo, Tokyo 113, Japan

Received 24 April 1992
Revised 16 September 1992

Abstract

Amari, Shun-ichi, Backpropagation and stochastic gradient descent method, Neurocomputing 5 (1993) 185–196.

The backpropagation learning method has opened a way to wide applications of neural network research. It is a type of the stochastic descent method known in the sixties. The present paper reviews the wide applicability of the stochastic gradient descent method to various types of models and loss functions. In particular, we apply it to the pattern recognition problem, obtaining a new learning algorithm based on the information criterion. Dynamical propei.ies of learning curves are then studied based on an old paper by the author where the stochastic descent method was proposed for general multilayer networks. The paper is concluded with a short section offering some historical remarks.

Keywords. Stochastic descent; generalized delta rule; dynamics of learning; pattern classification; multilayer perceptron.

1. Introduction

Backpropagation learning opened a way to a wide variety of applications of neural networks. It has prevailed all over the world and proved capabilities of the new information processing technology based on neural networks. It can be regarded as a generalization of the Widrow adaline learning rule to the perceptron to be applicable to multilayer architecture. However, more generally it is a version of the stochastic descent learning method for parameterized networks, an idea which existed as far back as the 1960s.

In the present short paper, a new result on Kullback information learning is presented in this framework. We will also review the old paper by Amari [2] in which the stochastic descent method (or the generalized delta rule) was proposed for multilayer networks. Learning by hidden units was demonstrated by computer simulation in the sixties (Amari [3]). However, the present paper is not mere retrospection. Instead, we should like to examine, from a current point of view, several old ideas, some of which have been rediscovered, but some of which

Correspondence to: Prof. Shun-ichi Amari, Dept. of Mathematical Engineering, University of Tokyo, Bunkyo-ku, Hongo, Tokyo 113, Japan, fax: 81 3 5689 5752, email: amari@sat.t.u-tokyo.ac.jp

still remained unexplored. They include, for example, the asymptotic behavior of accuracy and speed of learning, and the dynamic response of a learning network, when the optimal target changes with time.

The present paper is a supplement to the review paper by Amari [5], which stresses the mathematical foundations of statistical neurodynamics.

2. Stochastic descent learning rule

Let us consider an information processing system which receives a vector input signal x and emits an output signal z. The system includes feedforward type connections, but not feedback connections. It defines a mapping from the set $X = \{x\}$ of input signals to the set $Z = \{z\}$ of output signals,

$$z = F(x) . \tag{2.1}$$

When the system includes a number of modifiable parameters $\vartheta = (\vartheta_1, ..., \vartheta_n)$, the input-output function (2.1) is specified by ϑ,

$$z = F(x ; \vartheta) . \tag{2.2}$$

Here, we assume that F is differentiable with respect to ϑ.

When an input signal x is processed by a system specified by ϑ, a loss is caused because the system might not be optimally tuned. The loss is denoted by $l(x ; \vartheta)$. In some case, a desired output y accompanies x. In this case, the loss is written as $l(x, y ; \vartheta)$, denoting the loss when x with a desired or teacher signal y is processed by the network specified by ϑ.

Let us assume that the input signal x is generated subject to a fixed but unknown probability distribution $p(x)$ each time independently. The accompanying desired output y is usually a function of x

$$y = y_d(x)$$

called the desired output. It is sometimes disturbed by noise. In this case, y is generated subject to the conditional probability $p(y \mid x)$ and the expectation of y,

$$y_d(x) = E\,[y \mid x] = \int y\, p(y \mid x)\, \mathrm{d}y\,, \tag{2.3}$$

where $E\,[y \mid x]$ is the conditional expectation of y under the condition that the input x is the desired signal [24]. The stochastic y is its noisy version. In the noiseless case, y is written as

$$p(y \mid x) = \delta\big(y - y_d(x)\big)$$

by using the delta function.

In the case of learning without a teacher (i.e. self-organization), the accompanying signal y is missing, so in this case the loss $l(x, \vartheta)$ does not include y.

The risk $R(\vartheta)$ of using a network specified by ϑ is given by the expectation of the loss,

$$R(\vartheta) = E\,[l(x, y ; \vartheta)] = \int l(x, y ; \vartheta)\, p(x)\, p(y \mid x)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{2.4}$$

The network that minimizes the risk is said to be optimal and the optimal parameter is denoted by ϑ_0.

By receiving an input-output pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ at time t, $t = 1, 2, \ldots$, the stochastic descent rule modifies the current parameter ϑ_t at t to

$$\vartheta_{t+1} = \vartheta_t - \alpha_t C \frac{\partial l(\boldsymbol{x}_t, \boldsymbol{y}_t;\, \vartheta_t)}{\partial \vartheta}, \tag{2.5}$$

where α_t is a positive constant which may depend on t, C is a positive-definite matrix and $\partial/\partial\vartheta$ is the gradient operator. Since

$$E\left[\frac{\partial l(\boldsymbol{x}, \boldsymbol{y};\, \vartheta)}{\partial \vartheta}\right] = \frac{\partial R(\vartheta)}{\partial \vartheta},$$

the above rule modifies the current ϑ in the direction of decreasing $R(\vartheta)$ on average depending on the randomly generated $(\boldsymbol{x}, \boldsymbol{y})$. This is the reason why it is called the stochastic descent method [2]. In the simplest case, the α_t is put equal to a constant α and C is merely a unit matrix E.

3. Examples of learning systems

3.1 Adaline and potential function method

Let us consider the simplest case where output z is a scalar. An adaptive neuron introduced by Widrow [23] is the following linear element,

$$z = F(\boldsymbol{x}, \vartheta) = \vartheta \cdot \boldsymbol{x},$$

with the loss function

$$l(\boldsymbol{x}, y;\, \vartheta) = \tfrac{1}{2}\{y_d(\boldsymbol{x}) - z\}^2,$$

where y_d typically takes the values 1 or -1.

Let

$$e(\boldsymbol{x}) = y_d(\boldsymbol{x}) - z(\boldsymbol{x})$$

be the error signal. The learning rule (2.5) is then written as

$$\vartheta_{t+1} = \vartheta_t + \alpha_t C e(\boldsymbol{x})\boldsymbol{x}.$$

The potential function method by Aizerman et al. [1] uses a number of fixed non-linear functions $a_i(\boldsymbol{x})$ as elements to produce the output

$$z = F(\boldsymbol{x}, \vartheta) = \sum \vartheta_i a_i(\boldsymbol{x}).$$

The loss is also

$$l(\boldsymbol{x}, y;\, \vartheta) = \tfrac{1}{2}\{y_d(\boldsymbol{x}) - z\}^2.$$

The learning rule is

$$\vartheta_{t+1} = \vartheta_t + \alpha_t C e(\boldsymbol{x})\, \boldsymbol{a}(\boldsymbol{x}) ,$$

where

$$\boldsymbol{a}(\boldsymbol{x}) = \Big(a_1(\boldsymbol{x}), ..., a_n(\boldsymbol{x})\Big) .$$

It should be remarked that the above loss functions, and hence the risk functions, are quadratic in ϑ. Hence, there are no local minima except for the global minimum in $R(\vartheta)$.

3.2 Backpropagation

Consider a multilayer neural network with analog neurons. Let $x_j^{(m-1)}$ be the jth input to the neurons of the mth layer ($m = 1, 2, ..., k$), where $x_0^{(m-1)} \equiv 1.0$ is the required bias input. The ith element of the mth layer calculates the weighted sum of inputs,

$$u_i^{(m)} = \sum_j w_{ij}^{(m)} x_j^{(m-1)} ,$$

where $w_{ij}^{(m)}$ are the connection weights of the mth layer, and emits the output

$$x_i^{(m)} = f(u_i^{(m)}) ,$$

where f is a sigmoidal function. This $x_i^{(m)}$ in turn becomes the ith input of the next $(m+1)$th layer. The overall input to the network is given by

$$x_j = x_j^{(0)}$$

and the overall input is given from the kth layer by

$$z_i = x_i^{(k)} = f(u_i^{(k)}) .$$

The above equations, therefore, give the input-output relation

$$\boldsymbol{z} = \boldsymbol{F}(\boldsymbol{x}, \vartheta)$$

recurrently, where the parameter ϑ is composed of all the $w_{ij}^{(m)}$. When the squared error is used as the loss,

$$l(\boldsymbol{x}, \boldsymbol{y}; \vartheta) = \frac{1}{2} \sum (y_i - z_i)^2 ,$$

the stochastic gradient method gives the learning rule of the type,

$$\Delta w_{ij}^{(m)} = -\alpha \frac{\partial l(\boldsymbol{x}, \boldsymbol{y}; \vartheta)}{\partial \vartheta_i} = \alpha e_i^{(m)} x_j^{(m-1)} .$$

Here, $e_i^{(m)}$ is called the error signal of the ith neuron of the mth layer. The error signals are given recurrently by

$$e_i^{(k)} = f'(u_i^{(k)}) ,$$

$$e_i^{(m)} = f'(u_i^{(m)}) \sum_j w_{ji}^{(m)} e_j^{(m+1)} , \qquad m = 1, ..., k-1 .$$

It is remarkable that the error signals $e_i^{(m)}$ can be recurrently calculated from the last layer by backpropagating them. This fact was discovered by Werbos [22], Parker [18], and Rumelhart, Williams and Hinton [20], so that the method is called backpropagation.

3.3 Radial basis function method

Given a scalar function $h(\boldsymbol{x})$, consider a function given by a weighted sum of functions $h(\boldsymbol{x} - \boldsymbol{b}_i)$,

$$z = F(\boldsymbol{x}\,;\,\vartheta) = \sum_{i=1}^{n} w_i h(\boldsymbol{x} - \boldsymbol{b}_i)\,.$$

This is called the radial basis function expansion. Here, the modifiable parameter ϑ is a vector consisting of

$$\vartheta = (w_1, ..., w_n\,;\, \boldsymbol{b}_1, ..., \boldsymbol{b}_n)\,.$$

The Gaussian function is typically used as $h(\boldsymbol{x})$, and its superiority was shown by Poggio and Girosi [16]. When an appropriate loss function is used (typically the squared error), the stochastic descent rule is easily applied to this case.

3.4 Learning vector quantization

Consider the problem of dividing the signal space $X = \{\boldsymbol{x}\}$ into n regions D_i,

$$\bigcup_{i=1}^{n} \overline{D_i} = X\,, \qquad D_i \cap D_j = \emptyset \;\; (i \neq j)\,.$$

This is called the vector quantization of the signal space X. Let $\boldsymbol{w}_i$ be the representative of region D_i.

Given n representatives $\boldsymbol{w}_i$ $(i = 1, ..., n)$, we calculate the distance between input $\boldsymbol{x}$ and $\boldsymbol{w}_i$,

$$d_i(\boldsymbol{x}) = \tfrac{1}{2}\,|\boldsymbol{x} - \boldsymbol{w}_i|^2\,.$$

An input signal $\boldsymbol{x}$ is decided to belong to D_i, when $\boldsymbol{w}_i$ is the closest representative to $\boldsymbol{x}$, that is

$$d_i(\boldsymbol{x}) < d_j(\boldsymbol{x})\,, \qquad j \neq i\,.$$

In other words, regions D_i are defined by

$$D_i = \{\boldsymbol{x} \mid d_i(\boldsymbol{x}) < d_j(\boldsymbol{x}),\, j \neq i\}\,.$$

Consider a system consisting of n representatives $\vartheta = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_n)$, which quantize X into n regions. The quantization error

$$l(\boldsymbol{x}, \vartheta) = \min_i\; d_i(\boldsymbol{x})$$

can be regarded as the loss function when $\boldsymbol{x}$ is processed by a system with parameter ϑ. The risk function $R(\vartheta)$ is written as

$$R(\vartheta) = \sum_{i=1}^{n} \int_{D_i} \tfrac{1}{2} p(x)\, |x - w_i|^2 \, dx \, .$$

The quantization which minimizes $R(\vartheta)$ is said to be optimal. Kohonen [13] proposed the following learning method called the learning vector quantization method,

$$\begin{aligned} \Delta w_i &= -\alpha_t C(w_i - x) \, , & x &\in D_i \, , \\ \Delta w_j &= 0 & x &\notin D_j \, . \end{aligned}$$

No specific teacher signals y are given, so that this is a type of self-organization. This learning rule is also given by the stochastic descent method. The calculation of the gradient $R(\vartheta)$ is not immediate, because the regions D_i depend on ϑ. However, it was proved in the old paper by Amari [2]. See also Kohonen [14].

4. Information-theoretic approach to learning pattern classifier

Let us consider a pattern classification problem with k pattern classes C_i ($i = 1, ..., k$). Signals x belonging to C_i are assumed to be distributed subject to $p(x \mid C_i)$. Let p_i be the prior distribution of class C_i.

Let $g_i(x)$, $i = 1, ..., k$, be a set of discriminant functions to classify signals. When x is input, calculate $g_i(x)$, and if

$$\max_i \, g_i(x) = g_{i_0}(x) \, ,$$

decide that the pattern x belongs to C_{i_0}. The discriminant functions partition the signal space X into the regions

$$D_i = \{x \mid g_i(x) > g_j(x), \; j \neq i\} \, , \tag{4.1}$$

and a signal $x \in D_i$ is classified into C_i. It is well known that the best classification rule is given by

$$g_i(x) = \log p(C_i \mid x) \, , \tag{4.2}$$

where $p(C_i \mid x)$ is the posterior distribution of C_i under the condition that x is observed, and is given by

$$p(C_i \mid x) = \frac{p_i p(x \mid C_i)}{p(x)} \, , \tag{4.3}$$

$$p(x) = \sum_i p_i p(x \mid C_i) \, . \tag{4.4}$$

This rule minimizes the expected misclassification rate.

Now we consider a neural network whose ouput units correspond to the categories C_i. Let $z_i(x, \vartheta)$ be the output of the ith element, where ϑ summarizes all the modifiable parameters of the network. It is expected that the outputs $z_i(x, \vartheta)$ approximate the optimal discriminant functions by learning. More precisely, we try to approximate the logarithm of the conditional probability $p(C_i \mid x)$ by $z_i(x, \vartheta)$ such that the conditional probability suggested by the network is written in terms of the output $z_i(x, \vartheta.)$ as

$$z(C_i \mid \boldsymbol{x};\, \vartheta) = c(\vartheta, \boldsymbol{x}) \exp\{z_i(\boldsymbol{x}, \vartheta)\}\,. \tag{4.5}$$

Here, $c(\vartheta, \boldsymbol{x})$ is the normalizing constant to satisfy

$$\sum_{i=1}^{k} z(C_i \mid \boldsymbol{x};\, \vartheta) = 1\,.$$

It is written as

$$c(\vartheta, \boldsymbol{x}) = \exp\{-z(\boldsymbol{x}, \vartheta)\}\,, \tag{4.6}$$

where

$$z(\boldsymbol{x}, \vartheta) = \log\left[\sum_{i=1}^{k} \exp\{z_i(\boldsymbol{x}, \vartheta)\}\right]. \tag{4.7}$$

The natural divergence measure between two probability distributions $\boldsymbol{p} = \big(p(C_i)\big)$ and $\boldsymbol{z} = \big(z(C_i)\big)$ is given by the Kullback-Leibler information [15],

$$D(\boldsymbol{p} : \boldsymbol{z}) = \sum_{i=1}^{n} p(C_i) \log \frac{p(C_i)}{z(C_i)}\,. \tag{4.8}$$

A differential geometrical theory of the divergence is given by Amari [4], and Amari, Kurata and Nagaoka [8]. We search for the learning rule which minimizes the expectation of the divergence between $\{p(C_i \mid \boldsymbol{x})\}$ and $\{z(C_i \mid \boldsymbol{x}, \vartheta)\}$,

$$E\,[D(\boldsymbol{p} : \boldsymbol{z})] = \int p(\boldsymbol{x})\, D\,[p(C_i \mid \boldsymbol{x}) : z(C_i \mid \boldsymbol{x}, \vartheta)]\, \mathrm{d}\boldsymbol{x}\,. \tag{4.9}$$

When a signal $\boldsymbol{x} \in C_i$ is input, we define a loss of processing it by the network with parameter ϑ by

$$l(\boldsymbol{x}, C_i;\, \vartheta) = -z_i(\boldsymbol{x};\, \vartheta) + z(\boldsymbol{x}, \vartheta)\,. \tag{4.10}$$

Here the true class $y = C_i$ is given as the accompanying teacher signal. The stochastic descent learning rule is

$$\Delta\vartheta = -\alpha_t C\, \frac{\partial}{\partial\vartheta} \{(z(\boldsymbol{x}, \vartheta) - z_i(\boldsymbol{x};\, \vartheta)\}\,, \tag{4.11}$$

and is easily calculated from the input-output relation of the underlying network. When the network is multilayer, the error signals backpropagate as before.

In order to show that the above learning rule minimizes the expected divergence $E\,[D(\boldsymbol{p} : \boldsymbol{z})]$, we calculate the risk function

$$R(\vartheta) = E\,[l(\boldsymbol{x}, C_i;\, \vartheta)]\,.$$

Theorem.

$$R(\vartheta) = E\left[l(\boldsymbol{x}, y;\, \vartheta)\right] = E\left[D(\boldsymbol{p} : \boldsymbol{z};\, \vartheta)\right] + H(C) - H(X)\,, \tag{4.12}$$

where

$$\begin{aligned} D(\boldsymbol{p} : \boldsymbol{z};\, \vartheta) &= \sum p(C_i \,|\, \boldsymbol{x}) \log \frac{p(C_i \,|\, \boldsymbol{x})}{z(C_i \,|\, \boldsymbol{x};\, \vartheta)}\,, \\ H(C) &= -\sum p_i \log p_i\,, \\ H(X) &= -\int p(\boldsymbol{x}) \log p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\,. \end{aligned}$$

The proof is straightforward, but not simple. This implies that by the learning rule (4.11) the network approximates the conditional probability $p(C_i \,|\, \boldsymbol{x})$ in the sense that it minimizes the information $D(\boldsymbol{p} : \boldsymbol{q})$. This is a natural measure of the divergence of two probability distributions.

There are some other possible loss functions. One is to use the squared error of decision

$$l(\boldsymbol{x}, y;\, \vartheta) = \frac{1}{2} \sum_i |z(C_i \,|\, \boldsymbol{x};\, \vartheta) - \delta_{y_i}|^2\,,$$

where $\delta_{y_i} = 1$ when $y = C_i$ and 0 otherwise. This minimizes the risk function

$$R(\vartheta) = \frac{1}{2} \sum_{i=1}^{n} \int p(\boldsymbol{x})\, |z(C_i \,|\, \boldsymbol{x}, \vartheta) - p(C_i \,|\, \boldsymbol{x})|^2\, \mathrm{d}\boldsymbol{x}\,.$$

Another candidate is to minimize the difference in $p(C_i \,|\, \boldsymbol{x})$ and $z(C_i \,|\, \boldsymbol{x}, \vartheta)$ for each i by using an information-theoretic measure,

$$R(\vartheta) = \sum_{i=1}^{n} \int p(\boldsymbol{x})\, D^*[p(C_i) : z(C_i)]\, \mathrm{d}\boldsymbol{x}\,,$$

$$D^*[p : z] = p \log \frac{p}{z} + (1 - p) \log \frac{1 - p}{1 - z}\,.$$

Each of them also leads us to a stochastic descent learning rule. These rules are studied by Hampshire and Pearlmutter [12]. (See also Richard and Lippmann [17] and others). However, (4.12) is a more natural loss function for approximating the class probability distributions.

5. Dynamic properties of learning

Since the learning rule is stochastic, it is not necessarily guaranteed that ϑ_t converges to the optimal ϑ_0, even if we consider only a neighborhood of ϑ_0 in order to avoid the convergence to a local minimum. It is known from stochastic approximation (see e.g. [21]) that, when α_t is chosen to satisfy

$$\sum_{t=1}^{\infty} \alpha_t = \infty\,, \qquad \sum_{t=1}^{\infty} \alpha_t^2 < \infty\,, \tag{5.1}$$

ϑ_t converges to ϑ_0 with probability 1 under regularity conditions, provided $R(\vartheta)$ is unimodal. For example, (5.1) is satisfied by

$$\alpha_t = \frac{c}{t} \,.$$

This result is theoretically good, but practically not so useful. Because, when the optimal ϑ_0 deviates slightly, training time would be very long for adjusting the deviation because α_t might have become too small. In this sense, it is desirable to keep α to a small constant.

When α is a constant, Amari [2] analyzed the dynamic behavior of ϑ_t in the neighborhood of ϑ_0 and obtained the convergence speed to ϑ_0 and the fluctuation of ϑ_t around ϑ_0. It is surprising that these results are not yet well known in the neural network community. We show the results in the following. The proof is not difficult and is omitted (see the original Amari [2]). We need some preparation for stating the results.

Let us define two matrices A and B: A is the Hessian matrix of $R(\vartheta)$ at ϑ_0,

$$A = \frac{\partial^2}{\partial\vartheta\,\partial\vartheta}\, R(\vartheta_0) \,. \tag{5.2}$$

B is the covariance matrix, that is the second moments of $\partial l(\boldsymbol{x}, \boldsymbol{y}\,;\,\vartheta)/\partial\vartheta$ at ϑ_0

$$B = E\left[\left(\frac{\partial l}{\partial\vartheta}\right)\left(\frac{\partial l}{\partial\vartheta}\right)^{\mathrm{T}}\right], \tag{5.3}$$

where $\partial l/\partial\vartheta$ is a column vector and T denotes the transposition. Furthermore, let S be a linear operator which transforms a matrix M to

$$SM = CAM + (CAM)^{\mathrm{T}} \,, \tag{5.4}$$

where C is the matrix from Eq. (2.5). Since ϑ_t is a random variable depending on the randomly generated training sequences $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, 2, \dots$, we evaluate its expectation and covariance under the assumption that α is small and the initial value ϑ_1 is in a small neighborhood of ϑ_0.

Theorem. *The expectation of ϑ_t is given by*

$$E\left[\vartheta_t\right] = \vartheta_0 + (E - \alpha CA)^{t-1}(\vartheta_1 - \vartheta_0) \,. \tag{5.5}$$

The covariance matrix of ϑ_t is given by

$$\mathrm{Cov}\left[\vartheta_t\right] = 2\alpha\left\{\tilde{E} - (\tilde{E} - \alpha S)^{t-1}\right\} S^{-1}CBC^{\mathrm{T}} \,, \tag{5.6}$$

where $\tilde{E}$ is the identity operator.

The theorem shows that ϑ_t converges to ϑ_0 on average and fluctuates around ϑ_0 with covariance

$$2\alpha S^{-1}CBC^{\mathrm{T}} \,.$$

Let λ_0 be the minimum eigenvalue of the matrix CA. Then, the convergence speed is $(1 - \alpha\lambda_0)^t$, so that it is quicker when α is larger. However, the accuracy of convergence is given by the covariance matrix $2\alpha S^{-1}CBC^{\mathrm{T}}$, showing that ϑ_t fluctuates around ϑ_0 in the order of $\sqrt{\alpha}$. This implies that the accuracy is better when α is small. This trade-off between the accuracy and speed of convergence was pointed out in the old paper by Amari [2]. See also a recent paper [11] for a more modern treatment.

Learning is especially useful when the environment changes with time. We consider the case where the optimal ϑ_0 changes with time as

$$\vartheta_0(t) = \vartheta_0 + \boldsymbol{d} \sin \omega t \,, \tag{5.7}$$

where ω is the angular frequency of deviation. We assume that $\boldsymbol{d}$ is small, that ω is small, and that $\boldsymbol{d}$ is an eigenvector of CA corresponding to an eigenvalue λ. The general case can be obtained by superposition. We search for the condition that ϑ_t can follow this change of $\vartheta_0(t)$ well.

Theorem. *The expectation of ϑ_t is given by*

$$E\left[\vartheta_t\right] = \vartheta_0 + a\boldsymbol{d} \sin(\omega t - \varphi) \,, \tag{5.8}$$

where the damping coefficient a and the phase lag φ are obtained by

$$a = \frac{1}{\sqrt{1+z^2}} \,, \tag{5.9}$$

$$\varphi = -\tan^{-1} z \,, \tag{5.10}$$

$$z = \frac{\omega}{\alpha\lambda} \,. \tag{5.11}$$

When z is small, that is, when ω is smaller than $\alpha\lambda$, ϑ_t can follow the change of the optimal $\vartheta_0(t)$ well by learning.

There are many ideas to accelerate the learning process. The above analysis gives a method of choosing α. Heskes and Kappen [11] did a similar analysis independently, and make use of this information to choose α adaptively. Amari [2] proposed a method of increasing α when two successive $\Delta \boldsymbol{w}_t$, $\Delta \boldsymbol{w}_{t+1}$ have a positive inner product

$$\Delta \boldsymbol{w}_t \cdot \Delta \boldsymbol{w}_{t+1} > 0$$

and decreasing α when $\Delta \boldsymbol{w}_t \cdot \Delta \boldsymbol{w}_{t+1} < 0$. However, this idea appeared again and again but has not yet been fully studied.

6. Historical remarks and future perspectives

In the early sixties, many researchers enthusiastically studied learning capabilities of neural networks consisting of linear elements or nonlinear elements. The multilayer perceptron and back-coupled perceptron were also studied as models of various types of information processing (Rosenblatt [19], Block, Knight and Rosenblatt [10]). One may then ask the reason why such natural learning as backpropagation was not proposed at that time. We can point out two reasons.

The backpropagation method is a stochastic descent method, and it can easily be generalized from Widrow adaline learning. However, an adaline is a linear element so that the input-output relation of a network of adalines is also linear. Therefore, a multilayer adaline network can be reduced to a single layer network, and it is not effective to introduce hidden layers. The situation is the same in the potential function method where the output is non-linear in x but linear in the parameters ϑ. On the other hand, the risk function is a quadratic function of ϑ, so that the convergence to the global minimum is always guaranteed in these models. If non-linearity in the parameter is introduced, there is no such guarantee of global convergence. The backpropagation method suffers from the existence of local minima. Therefore, researchers were not brave enough to enter in such an unknown non-linear realm. Moreover, computers were too limited at that time for trying large-scale simulations.

A multilayer perceptron network is non-linear in the parameters ϑ. Therefore, it was not easy to introduce general learning rules. Even for a simple one-layer perceptron, some years were needed before the perceptron convergence theorem was proved [9]. Moreover, the learning rule was not written in the gradient descent form, because the output function is not differentiable. The greatest difficulty in introducing learning of hidden units in a multilayer perceptron layed in the fact that the outputs of perceptrons are binary signals taking on values of 0 and 1. Thus, the output is not differentiable with respect to the parameters.

Amari [2] began his study by rewriting the perceptron learning rule in the form of gradient descent even when binary output signals are used. A sigmoid function was then naturally introduced. It was then immediate to reach the general stochastic descent method including learning of hidden units. Amari [2] analyzed mainly the properties of linear discriminant functions and the dynamical properties of their learning, and the generalization to multilayer and more general models was explicitly presented. The error signals can be calculated by this gradient method, but it was not remarked that the error signals backpropagate in the process of practical calculations. Therefore, while the generalized delta rule was presented in this 1967 paper, the backpropagation algorithm itself was not given.

General stochastic descent learning was explained in detail in a 1968 Japanese book [3], where an example of learning by hidden units was presented by computer simulation. The network has two input units, four hidden units and one output unit with fixed weights. It was not possible to perform a larger size experiment by using a Japanese computer in the sixties.

By 1968 it was understood that there were local minima in $R(\vartheta)$ in many examples. This was the main reason why I did not go further along this line at the time. Instead, I devoted myself to more general mathematical research on neural networks such as statistical neuro-dynamics, associative memory, self-organization, etc. (see e.g. [5, 7]). Over the past 10 years the information geometrical method (Amari [4]) has been developed and is now being applied to analysis of neural manifolds [6, 8]. This will hopefully help open a new chapter of neural network research.

References

[1] M.A. Aizerman, E.M. Braverman and L.I. Rozonoer, A probabilistic problem on training automata to recognize classes and the method of potential functions, *Automation and Remote Control* 25 (1964) 1307–1323.

[2] S. Amari, Theory of adaptive pattern classifiers, *IEEE Trans. Elect. Comput.* EC-16 (1967) 299–307.

[3] A. Amari, *Geometrical Theory of Information* (in Japanese) (Kyoritsu Publishing, 1968).

[4] S. Amari, Differential-geometrical methods of statistics, *Springer Lecture Notes in Statistics* 28 (1985).

[5] S. Amari, Mathematical foundations of neurocomputing, *Proc. IEEE* 78 (1990) 1443–1463.

[6] S. Amari, Dualistic geometry of the manifold of higher-order neurons, *Neural Networks* 4 (1991) 443–451.

[7] S. Amari, N. Fujita and S. Shinomoto, Four types of learning curves, *Neural Computat.* 4 (1992) 605–618.

[8] S. Amari, K. Kurata and H. Nagaoka, Information geometry of Boltzmann machines, *IEEE Trans. Neural Networks* 3 (Mar. 1992) 260–271.

[9] H.D. Block, The perceptron, a model for brain functioning I, *Rev. Modern Physics* 34 (1962) 123–135.

[10] H.D. Block, B.W. Knight and F. Rosenblatt, Analysis of a four-layer series coupled perceptron II, *Rev. Modern Physics* 34 (1962) 135–142.

[11] T.M. Heskes and B. Kappen, Learning processes in neural networks, *Physical Rev.* A 440 (1991) 2718–2726.

[12] J.B. Hampshire and B.A. Pearlmutter, Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function, in: D. Touretzky, J. Elman, T. Sejnowski and G. Hinton, eds., *Proc. Connectionists Summer School* (Morgan Kaufmann, San Mateo, CA, 1990).

[13] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biol. Cybernet.* 43 (1982) 59–69.

[14] T. Kohonen, Self organizing maps: Optimization approaches, in: T. Kohonen, K. Makisara, O. Simula and J. Kangas, eds., *Artificial Neural Networks*, vol. 2 (North-Holland, Amsterdam, 1991) 981–990.

[15] S. Kullback, *Information Theory and Statistics* (Wiley, New York, 1959).

[16] T. Poggio and F. Girosi, Regularization algorithms for learning that are equivalent to multilayer networks, *Science* 247 (1990) 978–982.

[17] M.D. Richard and R.P. Lippmann, Neural network classifiers estimate Bayesian a posteriori probabilities, *Neural Comput.* 3 (1991) 461–483.

[18] D.B. Parker, Learning logic, Tech. Rep. TGR-47, M.I.T. Sloan School of Management, Cambridge, MA, 1982.

[19] F. Rosenblatt, *Principles of Neurodynamics* (Spartan, 1961).

[20] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart and J.L. McClelland, eds., *Parallel Distributed Processing*, vol. 1 (MIT Press, Cambridge, MA, 1986) 318–362.

[21] T. Wasan, *Stochastic Approximation* (Cambridge University Press, 1969).

[22] P. Werbos, Beyond regression: New tool for prediction and analysis in the behavioral sciences, Ph.D. thesis, Harvard Univ., 1974.

[23] B. Widrow, *A Statistical Theory of Adaptation* (Pergamon Press, Oxford, 1963).

[24] H. White, Learning in artificial networks: A statistical perspective, *Neural Comput.* 1 (1989) 425–464.

Shun-ichi Amari was a Professor at Kyushu University from 1963 to 1967, and since then he has been a Professor at the Department of Mathematical Engineering and Information Physics, University of Tokyo. He has been engaged in research in wide areas of mathematical engineering or applied mathematics, such as topological network theory, differential geometry of continuum mechanics, pattern recognition and information sciences. In particular, he has devoted himself to mathematical foundations of neural network theory, including statistical neurodynamics, dynamical theory of neural fields, associative memory, self-organization, and general learning theory. Another main subject of his research is information geometry proposed by himself, which develops and applies modern differential geometry to statistical inference, information theory, and modern control theory, proposing a new powerful method of information sciences and probability theory.

Dr. Amari is a founding governing board member of the International Neural Networks Society, a Coeditor-in-Chief of *Neural Networks*, and is on the editorial or advisory editorial boards of more than 15 international journals. He was the Conference Chair of the first International Joint Conference on Neural Networks, and he gave the Mahalanobis lecture on *Differential Geometry of Statistical Inference* at the 47th session of the International Statistical Institute.