Convergence Rate of Minimization Learning for Neural Networks

Mohamed, Marghny H.

Minamoto, Teruya

Niijima, Koichi

出版情報：DOI Technical Report. 141, 1997-10. Department of Informatics, Kyushu University
バージョン：
権利関係：

DOI-TR-141

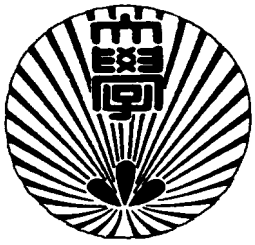
DOI Technical Report

Convergence Rate of Minimization Learning for Neural Networks

by

Marghny H. Mohamed, Teruya Minamoto and Koichi Niijima

October, 1997



Department of Informatics
Kyushu University
Fukuoka 812-81, Japan

Email: {mohamed,minamoto,niijima}@i.kyushu-u.ac.jp Phone: +81-92-583-7635

Convergence Rate of Minimization Learning for Neural Networks

Marghny H. Mohamed Teruya Minamoto Koichi Niijima

Abstract

In this paper, we present the convergence rate of the error in a neural network which was learnt by a constructive method. The constructive mechanism is used to learn the neural network by adding hidden units to this neural network. The main idea of this work is to find the eigenvalues of the transformation matrix concerning the error before and after adding hidden units in the neural network. By using the eigenvalues, we show the relation between the convergence rate in neural networks without and with thresholds in the output layer.

Keywords-Convergence rate, Constructive mechanism, Minimization learning, Hidden unit,Threshold, Error function, Matrix, Eigenvalues

1 Introduction

The size of a hidden layer in multilayer neural networks is one of the most important considerations when solving actual problems using the networks. In general, increasing the number of hidden units in a neural network may improve its approximation quality for training patterns, but not always improves the quality for new patterns. The size of neural network is desirable to be small. One way of improving a neural network is to reduce the number of hidden units preserving its quality. There are many methods to reduce the structure of the networks such as destructive, constructive, and genetic algorithms (Weymaere and Martens, 1994).

Destrucitive or pruning methods start from a fairly large network and remove unimportant connections or units (Chen, Thomas and Nixon, 1994; Hassibi and Stork, 1993; Moze and Smolensky, 1989). Constructive or growth methods start from a small network and dynamically grow the network (Fritzke, 1994; Giles, Chen, Sun, Chen, Lee and Goudreau, 1995; Heywood and Noakes, 1995; Hwang, Lay, Maechler, Martin and Schimert 1994; Nabhan and Zomaya, 1994; Niijima et al., 1997). First, these algorithms start from a minimal neural network with an input layer and an output layer. Second, add new hidden units to the network and train the corresponding weights until the neural network can map all the inputs to the corresponding outputs within an error bound. The advantage in using this method is that it can automatically find the size and the topology of the neural network without specifying them before training.

In the paper (Niijima et al., 1997), we proposed a learning algorithm which is carried out by the following two stages:

1. Determine the connection weights between the added hidden units and output units by minimizing the error function.

2. Determine the weights between the input layer and the hidden units.

This paper describes the convergence rate of the network learnt by our method above. This analysis is carried out by finding the transformation matrix concerning the error before and after adding hidden units in the neural network. The key idea is to find the eigenvalues of this matrix. We consider two types of neural network without and with thresholds in the output layer.

This paper is organized as follows. We present the analysis of the convergence rate in a neural network without and with thresholds in the output layer in Sect.2 and Sect.3 respectively. Section 4 presents the conclusion of this paper.

2 Convergence Rate in a Neural Network without Thresholds in the Output Layer

We consider a neural network which consists of an input layer with $n+1$ nodes, a hidden layer with h units, and an output layer with l units:

$$y_i = g\left(\sum_{j=1}^{h} w_{ij} f\left(\sum_{k=1}^{n+1} v_{jk} x_k\right)\right), \quad i = 1, 2, \ldots, l \ , \tag{1}$$

where x_k indicates the k-th input value, y_i the i-th output value, v_{jk} a weight connecting the k-th input node with the j-th hidden unit, and w_{ij} a weight between the j-th hidden unit and the i-th output unit. The functions $f(t)$ and $g(t)$ are given by

$$f(t) = \frac{1 - \exp(-t)}{1 + \exp(-t)} \ , \ \ g(t) = \frac{1}{1 + \exp(-t)} \ ,$$

respectively. We write (1) as

$$y = g\left(W f\left(Vx\right)\right) \ ,$$

where we set $x = {}^t(x_1, x_2, \ldots, x_n, x_{n+1})$ with $x_{n+1} = -1$, $y = {}^t(y_1, y_2, \ldots, y_l)$, $V = (v_{jk})$ and $W = (w_{ij})$. Moreover, $f(Vx)$ means ${}^t(f(V_1 x), f(V_2 x), \ldots, f(V_h x))$ and $g(Wf(Vx))$ indicates ${}^t(g(W_1 f(Vx)), g(W_2 f(Vx)), \ldots, g(W_l f(Vx)))$, where $V_j = {}^t(v_{j1}, v_{j2}, \ldots, v_{j,n+1})$ and $W_i = {}^t(w_{i1}, w_{i2}, \ldots, w_{ih})$. This network is shown in Fig.1. Let $(x^\nu, y^\nu), \nu = 1, 2, ..., m$, be training data for the network. We define an output error between the outputs of the network for the inputs x^ν and the relevant outputs y^ν by

$$J(V, W) = \sum_{\nu=1}^{m} \| g^{-1}(y^\nu) - W f(V x^\nu) \|^2 \ , \tag{2}$$

where $g^{-1}(y^\nu) = {}^t(g^{-1}(y_1^\nu), g^{-1}(y_2^\nu), \ldots, g^{-1}(y_l^\nu))$ with the inverse function $g^{-1}(s)$ of $s = g(t)$, and $\| \cdot \|$ stands for the Euclidean norm. To determine V and W, we need to minimize the error function (2).

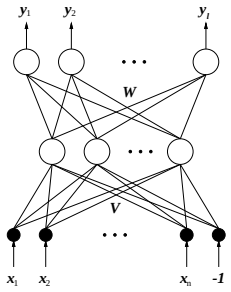


Figure 1: Neural network without thresholds in its output layer

The paper (Niijima et al., 1997) presents a technique for determining the weights V and W successively by adding one unit to the hidden layer of this network. Let $\mathbf{v}$ denote a connection weight vector between the $(h+1)$-th hidden unit and the input layer, and let $\mathbf{w}$ be a weight vector connecting the $(h+1)$-th hidden unit with the output layer. The neural network after adding the $(h+1)$-th hidden unit is shown in Fig.2.

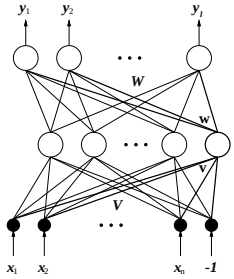


Figure 2: Neural network after adding one unit in the hidden layer in Fig.1

We denote the weight matrices $(V, \mathbf{v})$ and $(W, \mathbf{w})$ by $\tilde{V}$ and $\widetilde{W}$, respectively. Then a new error function can be written as

$$J(\tilde{V}, \widetilde{W}) = \sum_{\nu=1}^{m} \|g^{-1}(y^\nu) - \widetilde{W} f(\tilde{V} x^\nu)\|^2 \quad .$$

We describe how to determine the added weight vector $\mathbf{w}$. Since

$$\widetilde{W} f(\tilde{V} x^\nu) = W f(V x^\nu) + \mathbf{w} f(\mathbf{v} x^\nu),$$

the error function can be written as follows:

$$J(\tilde{V}, \widetilde{W}) = J(V, W) - 2\langle d, \mathbf{w} \rangle + a\|\mathbf{w}\|^2 \tag{3}$$

in which d and a denote

$$d = \sum_{\nu=1}^{m} f(\mathbf{v} x^\nu) c^\nu$$

and

$$a = \sum_{\nu=1}^{m} f^2(\mathbf{v}x^\nu),$$

where $c^\nu = g^{-1}(y^\nu) - Wf(Vx^\nu)$ and the symbol $\langle \cdot, \cdot \rangle$ indicates an inner product in R^l. When the vector $\mathbf{v}$ is fixed, the vector $\mathbf{w}$ which minimizes the error function (3) is given by

$$\mathbf{w} = \frac{d}{a}.$$

So the error after adding a hidden unit can be expressed as

$$\begin{aligned} \widetilde{c}^\nu &= g^{-1}(y^\nu) - \widetilde{W} f(\widetilde{V} x^\nu) \\ &= g^{-1}(y^\nu) - Wf(Vx^\nu) - \mathbf{w} f(\mathbf{v}x^\nu) \\ &= c^\nu - \frac{d}{a} f(\mathbf{v}x^\nu) \end{aligned}$$

Furthermore, let ${}^t\widetilde{C}_i = (c_i^1, c_i^2, \ldots, c_i^m)$, ${}^tC_i = (c_i^1, c_i^2, \ldots, c_i^m)$ and ${}^tS = (s_1, s_2, \ldots, s_m)$ with $s_\nu = f(\mathbf{v}x^\nu)$ and we have

$${}^t\widetilde{C}_i = {}^tC_i - \frac{1}{a}\, {}^tC_i S\, {}^tS. \tag{4}$$

Moreover we can rewrite (4) as

$${}^t\widetilde{C}_i = {}^tC_i \left(E_m - \frac{1}{a} S\, {}^tS \right) = {}^tC_i \Gamma_1(\mathbf{v}),$$

where E_m is the unit matrix and

$$\Gamma_1(\mathbf{v}) = E_m - S\, {}^tS/a.$$

We note that the matrix $\Gamma_1(\mathbf{v})$ is symmetric and this matrix has various remarkable characterizations. We describe two properties of them. One is

$$\langle \Gamma_1(\mathbf{v})U, U \rangle = \|U\|^2 - \frac{1}{a}\langle S, U \rangle^2 \geq \|U\|^2 - \frac{1}{a}\|S\|^2\|U\|^2 = 0 \tag{5}$$

and the other is

$$\langle \Gamma_1(\mathbf{v})U, U \rangle = \|U\|^2 - \frac{1}{a}\langle S, U \rangle^2 \leq \|U\|^2 \ . \tag{6}$$

From (5) and (6), we get

$$0 \leq \frac{\langle \Gamma_1(\mathbf{v})U, U \rangle}{\|U\|^2} \leq 1$$

which implies

$$0 \leq \lambda_j \leq 1, \qquad j = 1, \ldots, m,$$

where λ_j are the eigenvalues of the matrix $\Gamma_1(\mathbf{v})$.

Next, we calculate the eigenvalues of the matrix $\Gamma_1(\mathbf{v})$ to examine the convergence rate of the error in more detail. The matrix $\Gamma_1(\mathbf{v})$ can be written as

$$\Gamma_1(\mathbf{v}) = \begin{pmatrix} z_{11} & -z_{12} & \dots & -z_{1m} \\ -z_{21} & z_{22} & \dots & -z_{2m} \\ \dots & \dots & \dots & \dots \\ -z_{m1} & -z_{m2} & \dots & z_{mm} \end{pmatrix},$$

where $z_{ii} = 1 - s_i^2/a$, $z_{ij} = s_i s_j/a$, and $z_{ij} = z_{ji}$. The characteristic equation of this matrix is

$$\gamma_1(\lambda) = \begin{vmatrix} \lambda - z_{11} & z_{12} & \dots & z_{1m} \\ z_{21} & \lambda - z_{22} & \dots & z_{2m} \\ \dots & \dots & \dots & \dots \\ z_{m1} & z_{m2} & \dots & \lambda - z_{mm} \end{vmatrix} = 0.$$

By putting $\eta_i = (\lambda - 1)a/s_i^2$, we can reform $\gamma_1(\lambda)$ as follows:

$$\gamma_1(\lambda) = \frac{\prod_{i=1}^m {s_i}^2}{a^m} L_1(\eta) = 0,$$

where

$$L_1(\eta) = \begin{vmatrix} \eta_1 + 1 & 1 & \dots & 1 \\ 1 & \eta_2 + 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & \eta_m + 1 \end{vmatrix}.$$

The determinant $L_1(\eta)$ can be transformed by Laplace theorem as

$$L_1(\eta) = \prod_{\nu=1}^m \eta_\nu + \sum_{i=1}^m \prod_{\nu=1}^{i-1} \eta_\nu \prod_{\nu=i+1}^m \eta_\nu = \prod_{\nu=1}^m \eta_\nu + \sum_{i=1}^m \frac{\prod_{\nu=1}^m \eta_\nu}{\eta_i}.$$

Thus, we have

$$\begin{aligned} \gamma_1(\lambda) &= \frac{\prod_{i=1}^m s_i^2}{a^m} \left(\prod_{\nu=1}^m \eta_\nu + \sum_{i=1}^m \frac{\prod_{\nu=1}^m \eta_\nu}{\eta_i} \right) \\ &= \lambda(\lambda - 1)^{m-1} = 0. \end{aligned}$$

Hence the eigenvalues of this matrix are given by

$$\lambda = 1, 1, \dots, 1, 0.$$

These eigenvalues mean that the convergence rate of errors before and after adding hidden units is not depending on the connection weights $\mathbf{v}$ between the hidden layer and input layer.