LEARNING METHOD UTILIZING SINGULAR REGION OF MULTILAYER PERCEPTRON

Ryohei Nakano, Seiya Satoh and Takayuki Ohwaki
Department of Computer Science, Chubu University, 1200 Matsumoto-cho, Kasugai 487-8501, Japan

Keywords: Multilayer perceptron, Singular region, Learning method, Polynomial network, XOR problem.

Abstract: In a search space of multilayer perceptron having J hidden units, MLP(J), there exists a singular flat region created by the projection of the optimal solution of MLP($J-1$). Since such a singular region causes serious slowdown for learning methods, a method for avoiding the region has been aspired. However, such avoiding does not guarantee the quality of the final solution. This paper proposes a new learning method which does not avoid but makes good use of singular regions to find a solution good enough for MLP(J). The potential of the method is shown by our experiments using artificial data sets, XOR problem, and a real data set.

1 INTRODUCTION

It is known in MLP learning that an MLP(J) parameter subspace having the same input-output map as an optimal solution of MLP($J-1$) forms a singular region, and such a singular flat region causes stagnation of learning (Fukumizu and Amari, 2000). Natural gradient (Amari, 1998; Amari et al., 2000) was once proposed to avoid such stagnation of MLP learning, but even the method may get stuck in singular regions and is not guaranteed to find a good enough solution. Recently an alternative constructive method has been proposed (Minnett, 2011).

It is also known that many useful statistical models, such as MLP, Gaussian mixtures, and HMM, are singular models having singular regions where parameters are nonidentifiable. While theoretical research has been vigorously done to clarify mathematical structure and characteristics of singular models (Watanabe, 2009), experimental and algorithmic research is rather insufficient to fully support the theories.

In MLP parameter space there are many local minima forming equivalence class (Sussmann, 1992). Even if we exclude equivalence class, it is widely believed that there still remain local minima (Duda et al., 2001). When we adopt an exponential function as an activation function in MLP (Nakano and Saito, 2002), there surely exist local minima due to the expressive power of polynomials. In XOR problem, however, it was proved there is no local minima (Hamey, 1998). Thus, since we have had no clear knowledge of MLP parameter space, we usually run a learning method repeatedly changing initial weights to find a good enough solution.

This paper proposes a new learning method which does not avoid but makes good use of singular regions to find a good enough solution. The method starts with MLP having one hidden unit and then gradually increases the number of hidden units until the intended number. When it increases the number of hidden units from $J-1$ to J, it utilizes an optimum of MLP($J-1$) to form the singular region in MLP(J) parameter space. The singular region forms a line, and the learning method can descend in the MLP(J) parameter space since points along the line are saddles. Thus, we can always find a solution of MLP(J) better than the local minimum of MLP($J-1$). Our method is evaluated by the experiments for sigmoidal or polynomial-type MLPs using artificial data sets, XOR problem and a real data set.

Section 2 describes how sigular regions of MLP can be constructed. Section 3 explains the proposed method, and Section 4 shows how the method worked in our experiments.

2 SINGULAR REGION OF MULTILAYER PERCEPTRON

This section explains how an optimum of MLP($J-1$) can be used to form the singular region in MLP(J) parameter space (Fukumizu and Amari, 2000). This

Nakano R., Satoh S. and Ohwaki T..
LEARNING METHOD UTILIZING SINGULAR REGION OF MULTILAYER PERCEPTRON.
DOI: 10.5220/0003652501060111
In *Proceedings of the International Conference on Neural Computation Theory and Applications* (NCTA-2011), pages 106-111
ISBN: 978-989-8425-84-3

is universal in the sense that it does not depend on the choice of a target function or an activation function.

Consider the following MLP(J), MLP having J hidden units and one output unit. Here $\theta_J = \{w_0, w_j, w_j, j = 1,...,J\}$, and $w_j = (w_{jk})$.

$$f_J(x;\theta_J) = w_0 + \sum_{j=1}^{J} w_j z_j, \quad z_j \equiv g(w_j^T x) \tag{1}$$

Let input vector $x = (x_k)$ be K-dimensional. Given training data $\{(x^\mu, y^\mu), \mu = 1,...,N\}$, we want to find the parameter vector θ_J which minimizes the following error function.

$$E_J = \frac{1}{2}\sum_{\mu=1}^{N} (f_J^\mu - y^\mu)^2, \quad f_J^\mu \equiv f_J(x^\mu;\theta_J) \tag{2}$$

At the same time we consider the following MLP($J-1$) having $J-1$ hidden units, where $\theta_{J-1} = \{u_0, u_j, u_j, j = 2,...,J\}$.

$$f_{J-1}(x;\theta_{J-1}) = u_0 + \sum_{j=2}^{J} u_j v_j, \quad v_j \equiv g(u_j^T x) \tag{3}$$

For this MLP($J-1$) let $\widehat{\theta}_{J-1}$ denote a critical point which satisfies the following

$$\frac{\partial E_{J-1}(\theta)}{\partial \theta} = 0.$$

The necessary conditions for the critical point of MLP ($J-1$) are shown below. Here $j = 2,...,J$, $f_{J-1}^\mu \equiv f_{J-1}(x^\mu;\theta_{J-1})$, and $v_j^\mu \equiv g(u_j^T x^\mu)$.

$$\begin{aligned}
\frac{\partial E_{J-1}}{\partial u_0} &= \sum_\mu (f_{J-1}^\mu - y^\mu) = 0 \\
\frac{\partial E_{J-1}}{\partial u_j} &= \sum_\mu (f_{J-1}^\mu - y^\mu)\, v_j^\mu = 0 \\
\frac{\partial E_{J-1}}{\partial u_j} &= u_j \sum_\mu (f_{J-1}^\mu - y^\mu)\, g'(u_j^T x^\mu)\, x^\mu = 0
\end{aligned}$$

Now we consider the following three reducible projections α, β, γ, and let $\widehat{\Theta}_J^\alpha$, $\widehat{\Theta}_J^\beta$, and $\widehat{\Theta}_J^\gamma$ denote the regions obtained by applying these three projections to an optimum of MLP($J-1$) $\widehat{\theta}_{J-1} = \{\widehat{u}_0, \widehat{u}_j, \widehat{u}_j, j = 2,...,J\}$.

$$\widehat{\theta}_{J-1} \xrightarrow{\alpha} \widehat{\Theta}_J^\alpha, \quad \widehat{\theta}_{J-1} \xrightarrow{\beta} \widehat{\Theta}_J^\beta, \quad \widehat{\theta}_{J-1} \xrightarrow{\gamma} \widehat{\Theta}_J^\gamma$$

$$\begin{aligned}
\widehat{\Theta}_J^\alpha &\equiv \{\theta_J |\ w_0 = \widehat{u}_0,\ w_1 = 0, \\
&\qquad w_j = \widehat{u}_j, w_j = \widehat{u}_j, j = 2,\cdots,J\} \qquad (4) \\
\widehat{\Theta}_J^\beta &\equiv \{\theta_J |\ w_0 + w_1 g(w_{10}) = \widehat{u}_0, \\
&\qquad w_1 = [w_{10}, 0, \cdots, 0]^T, \\
&\qquad w_j = \widehat{u}_j, w_j = \widehat{u}_j, j = 2,\cdots,J\} \qquad (5) \\
\widehat{\Theta}_J^\gamma &\equiv \{\theta_J |\ w_0 = \widehat{u}_0,\ w_1 + w_2 = \widehat{u}_2, \\
&\qquad w_1 = w_2 = \widehat{u}_2, \\
&\qquad w_j = \widehat{u}_j, w_j = \widehat{u}_j, j = 3,\cdots,J\} \qquad (6)
\end{aligned}$$

(a) region $\widehat{\Theta}_J^\alpha$ is K-dimensional since free w_1 is a K-dimensional vector.

(b) region $\widehat{\Theta}_J^\beta$ is two-dimensional since three free weights must satisfy the following

$$w_0 + w_1\, g(w_{10}) = \widehat{u}_0.$$

(c) region $\widehat{\Theta}_\gamma$ is a line since we have only to satisfy

$$w_1 + w_2 = \widehat{u}_2.$$

Here we review a critical point where the gradient $\partial E/\partial\theta$ of an error function $E(\theta)$ gets zero. In the context of minimization, a critical point is classified into a local minimum and a saddle. A critical point θ_0 is classified as a local minimum when any point θ in its neighborhood satisfies $E(\theta_0) \le E(\theta)$, otherwise is classified as a saddle.

Now we classify a local minimum into a wok-bottom and a gutter. A wok-bottom θ_0 is a strict local minimum where any point θ in its neighborhood satisfies $E(\theta_0) < E(\theta)$, and a gutter θ_0 is a point having a continuous subspace connected to it where any point θ in the subspace satisfies $E(\theta) = E(\theta_0)$ or $E(\theta) \approx E(\theta_0)$.

The necessary conditions for the critical point of MLP (J) are shown below. Here $j = 2,...,J$.

$$\begin{aligned}
\frac{\partial E_J}{\partial w_0} &= \sum_\mu (f_J^\mu - y^\mu) = 0 \\
\frac{\partial E_J}{\partial w_1} &= \sum_\mu (f_J^\mu - y^\mu)\, z_1^\mu = 0 \\
\frac{\partial E_J}{\partial w_j} &= \sum_\mu (f_J^\mu - y^\mu)\, z_j^\mu = 0, \\
\frac{\partial E_J}{\partial w_1} &= w_1 \sum_\mu (f_J^\mu - y^\mu)\, g'(w_1^T x^\mu)\, x^\mu = 0 \\
\frac{\partial E_J}{\partial w_j} &= w_j \sum_\mu (f_J^\mu - y^\mu)\, g'(w_j^T x^\mu)\, x^\mu = 0
\end{aligned}$$

Then we check if regions $\widehat{\Theta}_J^\alpha$, $\widehat{\Theta}_J^\beta$, and $\widehat{\Theta}_J^\gamma$ satisfy these necessary conditions.

Note that in these regions we have $f_J^\mu = f_{J-1}^\mu$ and $z_j^\mu = v_j^\mu$, $j = 2,...,J$. Thus, we see that the first, third, and fifth equations hold, and the second and fourth equations are needed to check.

(a) In region $\widehat{\Theta}_J^\alpha$, since weight vector w_1 is free, the output of the first hidden unit z_1^μ is free, which means it is not guaranteed that the second and fourth equations hold. Thus, $\widehat{\Theta}_J^\alpha$ is not a singular region.

(b) In region $\widehat{\Theta}_J^\beta$, since $z_1^\mu (= g(w_{10}))$ is independent on μ, the second equation can be reduced to the first one, and holds. However, the fourth equation does

not hold in general unless $w_1 = 0$. Thus, the following area in $\widehat{\Theta}_J^{\beta}$ forms a singular region where w_{10} is free.

$$w_0 = \widehat{u}_0, \quad w_1 = 0, \quad w_{10} = free$$
$$w_j = \widehat{u}_j,\ w_j = \widehat{u}_j,\ \ j = 2,...,J$$

(c) In region $\widehat{\Theta}_J^{\gamma}$, since $z_1^{\mu} = v_2^{\mu}$, the second and fourth equations hold. Namely, $\widehat{\Theta}_J^{\gamma}$ is a singular region. Here we have one degree of freedom since we only have the following restriction

$$w_1 + w_2 = \widehat{u}_2 \tag{7}$$

This paper focuses on singular region $\widehat{\Theta}_J^{\gamma}$. It is rather convenient to search the region since it has only one degree of freedom and most points in the region are saddles (Fukumizu and Amari, 2000), which means we surely find a solution of MLP(J) better than that of MLP($J-1$).

3 SSF(SINGULARITY STAIRS FOLLOWING) METHOD

This section proposes a new learning method which makes good use of singular region $\widehat{\Theta}_J^{\gamma}$ of MLP. The method begins with MLP(J=1) and gradually increases the number of hidden units one by one until the intended largest number. The method is called Singularity Stairs Following (SSF) since it searches the space ascending singularity stairs one by one.

The procedure of SSF method is described below. Here J_{max} denotes the intended largest number of hidden units, and $w_0^{(J)}$, $w_j^{(J)}$, and $w_j^{(J)}$ are weights in MLP(J).

Singularity Stairs Following (SSF).
(Step 1). Find the optimal MLP(J =1) by repeating the learning changing initial weights. Let the best weights be $\widehat{w}_0^{(1)}$, $\widehat{w}_1^{(1)}$, and $w_1^{(1)}$. $J \leftarrow 1$.
(Step 2). While $J < J_{max}$, repeat the following to get MLP(J+1) from MLP(J).
(Step 2-1). If there are more than one hidden units in MLP(J), repeat the following for each hidden unit $m(= 1,...,J)$ to split.

Initialize weights of MLP(J+1) as follows:
$w_j^{(J+1)} \leftarrow \widehat{w}_j^{(J)},\ j \in \{0,1,...,J\}\backslash\{m\}$
$w_j^{(J+1)} \leftarrow \widehat{w}_j^{(J)},\ j = 1,...,J$
$w_{J+1}^{(J+1)} \leftarrow \widehat{w}_m^{(J)}$.
Initialize $w_m^{(J+1)}$ and $w_{J+1}^{(J+1)}$ many times while satisfying the restriction $w_m^{(J+1)} + w_{J+1}^{(J+1)} = \widehat{w}_m^{(J)}$ in the form of interpolation or extrapolation.

Perform MLP(J+1) learning for each initialization and get the best among MLPs(J+1) obtained for the hidden unit m to split.
(Step 2-2). Among the best MLPs(J+1) obtained above, select the true best and let it be $\widehat{w}_0^{(J+1)}, \widehat{w}_j^{(J+1)}, \widehat{w}_j^{(J+1)}$, j=1,...,J+1. $J \leftarrow J$+1.

Now we see our SSF method has the following characteristics.
(1) The optimal MLPs(J) are obtained one after another for $J = 1,...,J_{max}$. They can be used for model selection.
(2) It is guaranteed that the training performance of MLP(J+1) is better than that of MLP(J) since SSF descends in MLP(J+1) search space from the points corresponding to MLP(J) solution.

4 EXPERIMENTS

We evaluate the proposed method SSF for sigmoidal or polynomial-type MLPs using artificial data sets, XOR problem, and a real data set. Activation functions for sigmoidal and polynomial-type MLPs are $g(h) = 1/(1+e^{-h})$ and $g(h) = exp(h)$ respectively in eq. (1). Then the output of polynomial-type MLP is written as below.

$$f_J = \sum_{j=0}^{J} w_j z_j, \quad z_j = \prod_{k=1}^{K} (x_k)^{w_{jk}}$$

In performing SSF, since we have to move around in singular flat regions, we employ very weak regularization of weight decay where penalty coefficient $\rho = 0.001$. As a learning method we use a quasi-Newton method called BPQ (Saito and Nakano, 1997) since any first-order method is too slow to converge. The learning stops when any gradient element is less than 10^{-8} or the iteration exceeds 10,000 sweeps. As for the weight initialization for MLP(J=1), w_{jk} are initialized following normal Gaussian distribution, and initial w_j are set to zero without $w_0 = \overline{y}$.

4.1 Experiment of Sigmoidal MLP using Artificial Data

Our artificial data set for sigmoidal MLP was generated using the following MLP. Values of input x_k were randomly selected from the range $[-1,+1]$, and values of output y were generated by adding a small Gaussian noise $\mathcal{N}(0, 0.01^2)$ to MLP outputs. The sample size was 200.

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ -4 \end{pmatrix}, \quad (w_1, w_2) = \begin{pmatrix} -3 & 0 \\ 3 & 0 \\ 1 & 0 \\ 0 & 1/2 \\ 0 & -3 \end{pmatrix}$$

The number of hidden units was changed within 2: J_{max} = 2. We repeated MLP(J=1) learning 100 times, and obtained two kinds of solutions which are equivalent. The eigen values of Hessian matrix at the solution are all positive and the ratio of maximal to minimal eigenvalues $\lambda_{max}/\lambda_{min}$ is 10^3, which means the solution is a wok-bottom.

SSF was applied to get MLP(J=2) from the MLP(J=1). The result is shown in Fig. 1, where the horizontal axis is $w_1^{(2)}$ and the vertical axis is MSE (mean squared error). We stably got two kinds of solutions, and the better whose MSE $\approx 10^{-4}$ is a wok-bottom since $\lambda_{max}/\lambda_{min} \approx 10^4$.

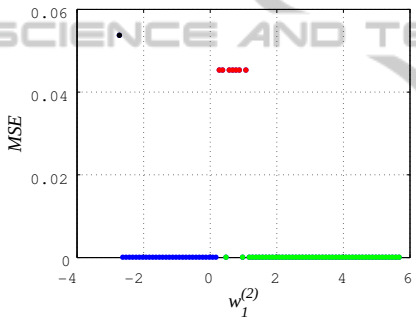

Figure 1: Result of SSF step from MLP(J=1) to MLP(J=2) for sigmoidal artificial data set.

As an existing method, we ran BPQ 100 times for MLP (J=2) and got four kinds of solutions. The same best solution as SSF was obtained 65 times and MSE of the other three are 0.0275, 0.0489, and 0.0499.

4.2 Experiment of Polynomial MLP using Artificial Data

Here we consider the following polynomial.

$$y = 2 + 4\, x_1^{-1}\, x_2^3 - 3\, x_3\, x_4^{1/2} - 2\, x_5^{-1/3}\, x_6\, x_7^2 \qquad (8)$$

Values of input x_k were randomly selected from the range $(0,1)$, values of output y were generated by adding a small Gaussian noise $\mathcal{N}(0, 0.1^2)$. The sample size was 200. Considering eq. (8), we set as $J_{max} = 3$.

We repeated MLP(J=1) learning 100 times and got two kinds of solutions whose MSE were 0.687 and 14.904. The former obtained 55 times is a wok-bottom since $\lambda_{max}/\lambda_{min} \approx 10^3$, and the latter is a gutter since $\lambda_{max}/\lambda_{min} \approx 10^{11}$. The better one was used for SSF.

SSF was applied to get MLP(J=2) from the MLP(J=1). Figure 2 shows the result. We stably obtained two kinds of solutions, and the better one is a wok-bottom since $\lambda_{max}/\lambda_{min} \approx 10^4$, and the other is a gutter since $\lambda_{max}/\lambda_{min} \approx 10^9$. The better solution was used for the next step of SSF.

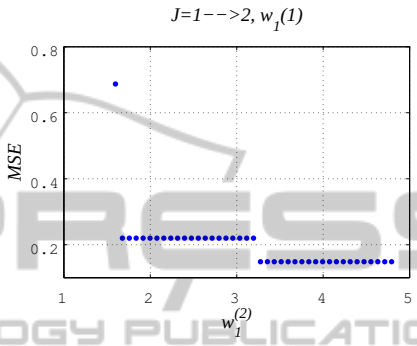

Figure 2: Result of SSF step from MLP(J=1) to MLP(J=2) for polynomial-type artificial data set.

When we apply SSF to get MLP(J=3), the hidden unit to split is either $\widehat{w}_1^{(2)}$ or $\widehat{w}_2^{(2)}$. Both reached much the same result as eq. (8), and Fig. 3 shows the result for splitting $\widehat{w}_1^{(2)}$. Again we stably got two kinds of wok-bottom solutions since $\lambda_{max}/\lambda_{min} \approx 10^5$ and 10^6.

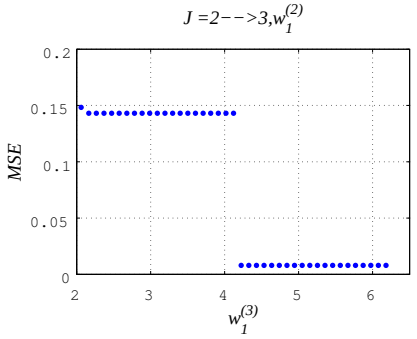

Figure 3: Result of SSF step from MLP(J=2) to MLP (J=3) for polynomial-type artificial data set.

As an existing method, we ran BPQ 100 times for MLP (J=3) and got various solutions. The same best solution as SSF was obtained 42 times.

4.3 Experiment of Sigmoidal MLP using XOR Problem

When we solve XOR problem using MLP(J=2), we have five degrees of freedom since there are nine weights for four sample points. Thus, in the learning of XOR problem, a learning method is easily trapped in the singular regions. We examined how SSF solves XOR problem.

We repeated MLP(J=1) learning 100 times using initial weights selected randomly from the range $(-5,+5)$. Even if considering equivalence class, we got various solutions which are wok-bottoms since $\lambda_{max}/\lambda_{min} \approx 10^3 \sim 10^4$.

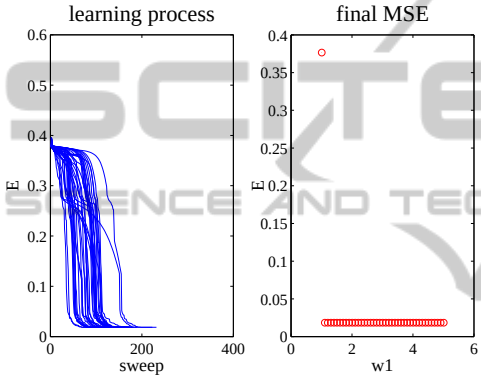

Figure 4: Result of SSF step from MLP(J=1) to MLP (J=2) for XOR problem.

Using one of the MLPs(J=1), we employed SSF to get MLP(J=2). Figure 4 shows how the learning went on and the final MSEs. Most learnings stopped within 200 sweeps, and every learning reached the true optimum, which is a wok-bottom since $\lambda_{max}/\lambda_{min} \approx 10^4$.

As an existing method, we ran BPQ 100 times for MLP (J=2). Figure 5 shows how each learning went on and 70 runs reached the true optimum while the others were trapped in the singular regions.

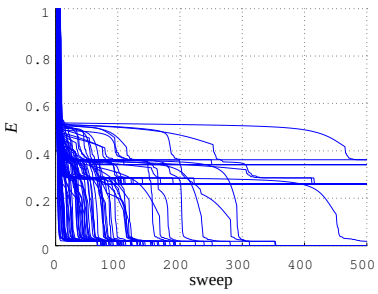

Figure 5: Learning process of an existing method for XOR problem.

4.4 Experiment of Polynomial MLP using Real Data

SSF was applied to ball bearings data (Journal of Statistics Education) ($N = 210$). The objective is to estimate fatigue (L10) using load (P), the number of balls (Z), and diameter (D). Before learning, variables were normalized as $x_k/max(x_k)$ and $(y-\bar{y})/std(y)$. We set as $J_{max} = 4$.

In MLP(J=1) learning, we obtained three kinds of solutions. For the next step of SSF we used the best solution whose MSE is 0.2757.

SSF was applied to get MLP(J=2) from the MLP(J=1) and the result is shown in Fig. 6. Most final MSEs were located at 0.223 and 0.27. The best MSE is 0.2229 and the solution is a gutter since $|\lambda_{max}/\lambda_{min}| \approx 10^{14}$.

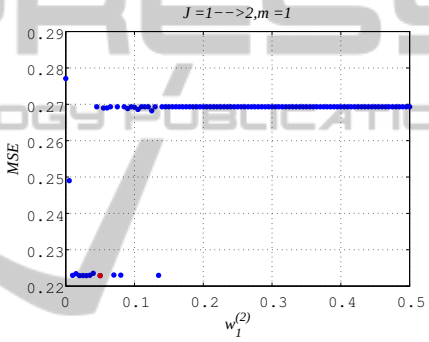

Figure 6: Result of SSF step from MLP(J=1) to MLP (J=2) for ball bearings data.

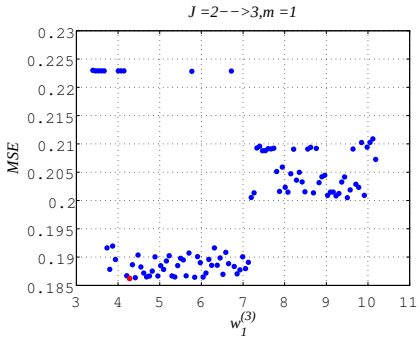

Figure 7: Result of SSF step from MLP(J=2) to MLP (J=3) for ball bearings data.

Then, SSF was applied to get MLP(J=3) from the MLP(J=2) and the result is shown in Fig. 7. Final MSEs were scattered in the form of three clusters. The best MSE is 0.1862 and the solution is a gutter since $|\lambda_{max}/\lambda_{min}| \approx 10^{15}$.

Finally, SSF was applied to get MLP(J=4) from the MLP(J=3) and the result is shown in Fig. 8. Fi-

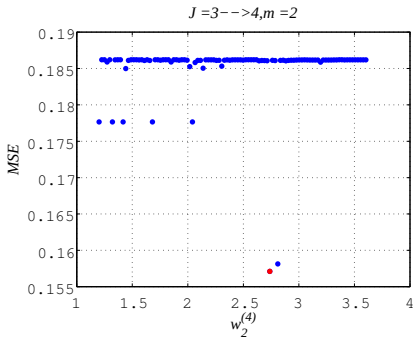

Figure 8: Result of SSF step from MLP(J=3) to MLP (J=4) for ball bearings data.

nal MSEs were scattered in the form of three clusters. The best MSE is 0.1571 and the solution is a gutter since $|\lambda_{max}/\lambda_{min}| \approx 10^{17}$.

As an existing method, we ran BPQ 100 times for MLP (J=4). Figure 9 shows the histogram of final MSEs. The figure shows the MSE of the final SSF solution is almost equivalent to that of the existing method.

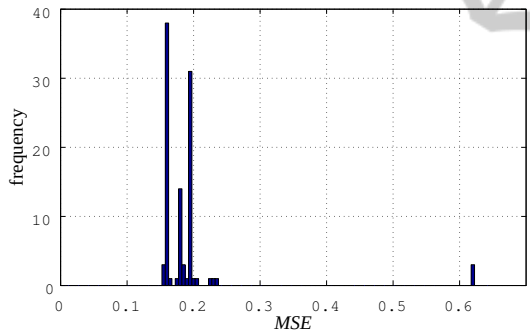

Figure 9: Solutions obtained by an existing method for ball bearings data.

5 CONCLUSIONS

This paper proposed a new MLP learning called SSF, which makes good use of singular regions. The method begins with MLP(J=1) and gradually increases the number of hidden units one by one. Our various experiments showed that SSF found solutions good enough for MLP(J). In the future we plan to improve our method.

ACKNOWLEDGEMENTS

This work was supported by Grants-in-Aid for Scientific Research (C) 22500212 and Chubu University Grant 22IS27A.

REFERENCES

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

Amari, S., Park, H. and Fukumizu, K. (2000). Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001). *Pattern classification, 2nd edition*. John Wiley & Sons, Inc.

Fukumizu, K. and Amari,S.(2000). Local minima and plateaus in heirarchical structure of multilayer perceptrons. *Neural Networks*, 13(3):317–327.

Hamey, L. G. C. (1998). XOR has no local minima: a case study in neural network error surface. *Neural Networks*, 11(4):669–681.

Minnett, R. C. J., Smith, A. T., Lennon Jr. W. C. and Hecht-Nielsen, R. (2011). Neural network tomography: network replication from output surface geometry. *Neural Networks*, 24(5):484–492.

Nakano, R. and Saito, K. (2002). Discovering polynomials to fit multivariate data having numeric and nominal variables. *LNAI* 2281:482–493.

Watanabe, S. (2009). *Algebraic geometry and statistical learning theory*. Cambridge Univ. Press.

Saito, K. and Nakano, R. (1997). Partial BFGS update and efficient step-length calculation for three-layer neural networks. *Neural Computation*, 9(1):239–257.

Sussman, H. J. (1992). Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 5(4):589–593.