Submitted to ICONIP'98

EFFECT OF BATCH LEARNING IN MULTILAYER NEURAL NETWORKS

Kenji Fukumizu
Email:fuku@brain.riken.go.jp

Lab. for Information Synthesis, RIKEN Brain Science Institute
Hirosawa 2-1, Wako, Saitama, 351-0198, Japan

ABSTRACT

This paper discusses batch gradient descent learning in multilayer networks with a large number of statistical training data. We emphasize on the difference between regular cases, where the prepared model has the same size as the true function, and overrealizable cases, where the model has surplus hidden units to realize the true function. First, experimental study on multilayer perceptrons and linear neural networks (LNN) shows that batch learning induces strong overtraining on both models in overrealizable cases, which means the degrade of generalization error by surplus units can be alleviated. We theoretically analyze the dynamics in LNN, and show that this overtraining is caused by shrinkage of the parameters corresponding to surplus units.

KEYWORDS: Multilayer network, Batch learning, Overtraining, Generalization error

1. INTRODUCTION – WHY THREE-LAYER NETWORKS? –

Although multilayer networks like multilayer perceptrons (MLP) have been used in many applications, their essential difference from other models has not been perfectly clarified. From the viewpoint of function approximation, Barron ([1]) shows that the MLP is a more effective approximator in a specific function space than linear models in that it avoids the curse of dimensionality. Fukumizu ([2]) discusses a statistical characteristic of multilayer structure, showing that the Fisher information matrix of a multilayer model can be singular. This causes unusual properties of multilayer networks in *overrealizable cases*, where the true function can be realized by a network with a smaller number of hidden units than the prepared model. One example shown in Fukumizu ([3]) is that the generalization error of the maximum likelihood estimator (MLE) of a three-layer linear neural network in overrealizable cases is larger than the generalization error in regular cases. This result implies, in a sense, a disadvantage of multilayer models. Then, why should we use a multilayer network?

To answer this question, as a favorable property of multilayer networks, we elucidate the existence of *overtraining*, attainment of the minimum generalization error in the middle of learning. There is a controversy on overtraining. Many practitioners assert its existence and recommend the use of a stopping criterion. Amari et al. ([4]) analyze overtraining theoretically and conclude that the effect of overtraining is very small if the parameter approaches to the MLE following the statistical asymptotic theory. However, the usual asymptotic theory cannot be applied in overrealizable cases, and the existence of overtraining has still been an open problem in such cases. The aim of this paper is to experimentally and theoretically investigate the existence of overtraining as a first step of the analysis of learning in multilayer networks.

2. STATISTICAL LEARNING

A feed-forward neural network model can be described as a parametric family of functions $\{\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})\}$ from $\mathbb{R}^L$ to $\mathbb{R}^M$, where $\boldsymbol{x}$ is an input vector and $\boldsymbol{\theta}$ is a parameter vector. A three-layer network with H hidden units is defined by

$$f_i(\boldsymbol{x};\boldsymbol{\theta}) = \textstyle\sum_{j=1}^{H} w_{ij}\ s(\sum_{k=1}^{L} u_{jk}x_k + \zeta_j) + \eta_i, \qquad (1)$$

where $\boldsymbol{\theta} = (w_{ij}, \eta_i, u_{jk}, \zeta_j)$ and the function $s(t)$ is an activation function. In the case of MLP, the sigmoidal function $s(t) = \frac{1}{1+e^{-t}}$ is used.

We use such a model for regression problems, assuming that an output of the target system is observed with a measurement noise. A sample $(\boldsymbol{x}, \boldsymbol{y})$ from the target system satisfies

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{v}, \qquad (2)$$

where $\boldsymbol{f}(\boldsymbol{x})$ is the *true function* and $\boldsymbol{v}$ is a noise subject to $N(0, \sigma^2 I_M)$, a normal distribution with 0 as its mean and $\sigma^2 I_M$ as its variance-covariance matrix. An input $\boldsymbol{x}$ is generated randomly with its probability density function $q(\boldsymbol{x})$, which is unknown to a learner. Training data $\{(\boldsymbol{x}^{(\nu)}, \boldsymbol{y}^{(\nu)}) | \nu = 1, \ldots, N\}$ are independently generated from the joint distribution of $q(\boldsymbol{x})$ and eq.(2). We assume that $\boldsymbol{f}(\boldsymbol{x})$ is perfectly realized by the prepared model; that is, there is a true parameter $\boldsymbol{\theta}_0$ such that $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_0) = \boldsymbol{f}(\boldsymbol{x})$. An overrealizable case is defined by the condition that the true function $\boldsymbol{f}(\boldsymbol{x})$ (or the *teacher*) is realized by a network with a smaller number of hidden units than H ([3]).

The objective function of training is the following *empirical error*:

$$\mathrm{E}_{emp} = \textstyle\sum_{\nu=1}^{N} \|\boldsymbol{y}^{(\nu)} - \boldsymbol{f}(\boldsymbol{x}^{(\nu)};\boldsymbol{\theta})\|^2. \qquad (3)$$

If we assume a parametric statistical model, $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{M/2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})\|^2\right)$ for the conditional probability, the parameter that minimizes E_{emp} coincides with the MLE, whose behavior for a large number of data is given by the statistical asymptotic theory.

Generally, some numerical method is needed to calculate the MLE unless the model is linear. One widely-used method is the steepest descent method, which leads a learning rule:

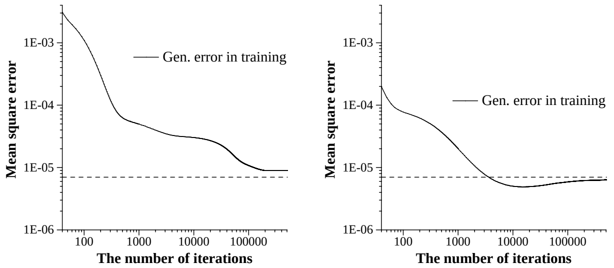
$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \beta \frac{\partial \mathrm{E}_{emp}}{\partial \boldsymbol{\theta}}, \qquad (4)$$

where β is a learning rate. In this paper, we discuss this learning rule. Since the error is given with all the fixed training data, the above learning is called *batch learning*. There are many researches on *on-line learning*, in which the parameter is always updated for a newly generated data, but we do not discuss it here.

The performance of a network is often evaluated by the generalization error:

$$\mathrm{E}_{gen} \equiv \int \|\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{f}(\boldsymbol{x})\|^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{5}$$

It is easy to know that the minimization of the empirical error roughly approximates the minimization of the generalization error. However, they are not exactly the same, and the decrease of E_{emp} during training does not ensure the decrease of E_{gen}. Then, it is extremely important to clarify the dynamical behavior of E_{gen} in learning. A curve showing E_{emp} or E_{gen} as a function of time is called a learning curve.



(a) Regular case

(b) Overrealizable case

Figure 1: Learning curves of MLP. The input, hidden, and output layer have 1, 2, and 1 units respectively. The number of training data is 100. The constant zero function is used as an overrealizable target.

3. LINEAR NEURAL NETWORKS

We must be careful in discussing experimental results on MLP especially in overrealizable cases. There is an almost flat subvariety around the global minima in overrealizable cases ([2]), and the convergence of learning is extremely slow. In addition, learning with a gradient method suffers from local minima like other nonlinear models. We cannot exclude their effects, and it often makes derived conclusions obscure.

Therefore, we introduce three-layer linear neural networks (LNN) as a model on which theoretical analysis is possible. The LNN model has the identity function as its activation, and defined by

$$\boldsymbol{f}(\boldsymbol{x}; A, B) = BA\boldsymbol{x}, \tag{6}$$

where A is a $H \times L$ matrix and B is a $M \times H$ matrix. We do not use bias terms in LNN for simplicity. We assume $H \leq L$ and $H \leq M$ throughout this paper. Although the function $\boldsymbol{f}(\boldsymbol{x}; A, B)$ is linear, the parameterization is quadratic, therefore, nonlinear. Note that the above model is not equivalent to the usual linear model $\boldsymbol{f}(\boldsymbol{x}; C) = C\boldsymbol{x}$ for a $M \times L$ matrix C, because the rank of the matrix BA is restricted to H in eq.(6), that is, the function space is the set of linear maps from $\mathbb{R}^L$ to $\mathbb{R}^M$ whose rank is no greater than H. Then, the MLE and the dynamics of learning in model eq.(6) are different from those of the usual linear model.

4. EXPERIMENTAL STUDY

In this section, we experimentally investigate the generalization error of MLP and LNN to see whether overtraining is commonly observed in both models in overrealizable cases.

The steepest descent method in MLP leads the well-known error back propagation. To avoid the problems discussed in Section 3 as much as possible, we adopt the following experimental design. For a fixed set of training data, we try 30 different vectors for initialization, and select the trial that gives the least E_{emp} at the end of the training. Figure 1 shows the average of E_{gen} over 30 different simulations changing the set of training data. It shows clear overtraining in the overrealizable case. On the other hand, the learning curve in the regular case shows no meaningful overtraining.

Next, we make simulations on LNN. It is known that the MLE of the LNN model is analytically solvable ([5]), and it is practically absurd to use the steepest descent method. However, since our interest here is not the MLE but the dynamics of learning, we study the behavior of steepest descent learning. For the training data of LNN, we use the notations: $X = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)})^T$ and $Y = (\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)})^T$. Then, the empirical error can be written by

$$\mathrm{E}_{emp} = \mathrm{Tr}[(Y - XA^TB^T)^T(Y - XA^TB^T)], \tag{7}$$

and the batch learning rule of LNN is given by

$$\begin{cases} A(t+1) & = A(t) + \beta\{B^TY^TX - B^TBAX^TX\}, \\ B(t+1) & = B(t) + \beta\{Y^TXA^T - BAX^TXA^T\}. \end{cases} \tag{8}$$

Figure 2 shows the average of learning curves for 100 simulations with various training data sets from the same probability. The two curves represent totally different behaviors. Only the overrealizable case shows eminent overtraining in the middle also in LNN.

From the above results, we can conjecture that there is an essential difference in dynamics of learning between regular and overrealizable cases, and overtraining is a universal property of the latter cases. If we use a good stopping criterion, the multilayer networks can have an advantage over conventional linear models, in that the degrade of the generalization error by surplus units is not so critical.

5. DYNAMICS OF LEARNING IN LINEAR NEURAL NETWORKS

We give a theoretical verification of the existence of overtraining, deriving an approximated solution of the continuous-time differential equation of the steepest descent method.

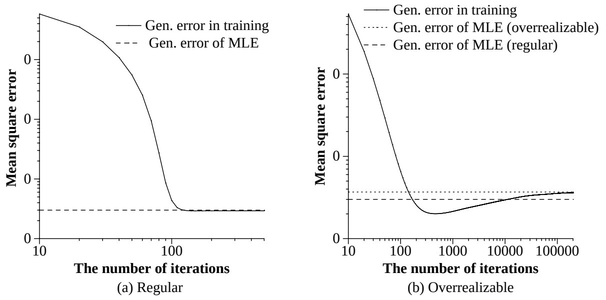
5.1. Solution of learning dynamics

In the rest of the paper, we put the following assumptions:

(a) $H \leq L = M$,
(b) $\boldsymbol{f}(\boldsymbol{x}) = B_0A_0\boldsymbol{x}$,
(c) $\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T] = \tau^2 I_L$,
(d) $A(0)A(0)^T = B(0)^TB(0)$.
(e) The rank of $A(0)$ and $B(0)$ are H.

We discuss the continuous-time differential equation instead of the discrete time update rule. Dividing the matrix Y as

$$Y = XA_0^TB_0^T + V, \tag{9}$$



(a) Regular

(b) Overrealizable

Figure 2: Learning curves of LNN. The number of input, hidden, and output units are 2, 1, and 2 respectively.

where V is the noise components, we have the differential equation of the steepest descent learning as

$$\left\{ \begin{array}{l} \dot{A} = \beta B^T(B_0A_0X^TX + V^TX - BAX^TX), \\ \dot{B} = \beta(B_0A_0X^TX + V^TX - BAX^TX)A^T. \end{array} \right. \tag{10}$$

Let $Z_O := \frac{1}{\sigma}V^TX(X^TX)^{-1/2}$, then all the elements of Z_O are independently subject to $N(0,1)$. We use the decomposition

$$X^TX = \tau^2NI_L + \tau^2\sqrt{N}Z_I, \tag{11}$$

where the off-diagonal elements of Z_I are subject to $N(0,1)$ and the diagonal elements are subject to $N(0,2)$ if N goes to infinity. Let

$$F = B_0A_0 + \frac{1}{\sqrt{N}}(B_0A_0Z_I + \frac{\sigma}{\tau}Z_O), \tag{12}$$

then we can approximate eq.(10) by

$$\left\{ \begin{array}{ll} \dot{A} & = \beta\tau^2NB^TF - \beta\tau^2NB^TBA, \\ \dot{B} & = \beta\tau^2NFA^T - \beta\tau^2NBAA^T. \end{array} \right. \tag{13}$$

The original equation eq.(10) can be considered as a perturbation of eq.(13), and this gives a good approximation if N is very large.

From the fact $\frac{d}{dt}(AA^T) = \frac{d}{dt}(B^TB)$ and the assumption (d), we have $AA^T = B^TB$. If we introduce $2L \times H$ matrix

$$R = \begin{pmatrix} A^T \\ B \end{pmatrix}, \tag{14}$$

then, R satisfies the differential equation

$$\frac{dR}{dt} = \beta\tau^2NSR - \frac{\beta\tau^2N}{2}RR^TR, \text{ where } S = \left(\begin{smallmatrix} 0 & F^T \\ F & 0 \end{smallmatrix} \right). \tag{15}$$

This is very similar to Oja's learning equation ([6]), which is known to be a solvable nonlinear differential equation ([7]). The key fact to solve eq.(15) is to derive a matrix Riccati differential equation:

$$\frac{d}{dt}(RR^T) = \beta\tau^2N\{SRR^T + RR^TS - (RR^T)^2\}. \tag{16}$$

We have the following

Theorem 1. *Assume that the rank of $R(0)$ is full. Then, the Riccati differential equation (16) has a unique solution for all $t \geq 0$, and the solution is given by*

$$\begin{aligned} R(t)R^T(t) &= e^{\beta\tau^2NSt}R(0) \\ &\times \left[I_H + \frac{1}{2}R(0)^T\{e^{\beta\tau^2NSt}S^{-1}e^{\beta\tau^2NSt} - S^{-1}\}R(0)\right]^{-1} \\ &\times R(0)^Te^{\beta\tau^2NSt}. \end{aligned} \tag{17}$$

5.2. Dynamics of generalization error

In this subsection, we show that E_{gen} in the middle of learning is smaller than E_{gen} for the MLE, the final state of E_{gen}, if the case is overrealizable.

From the assumption (c), we have

$$\mathrm{E}_{gen} = \mathrm{Tr}[(BA - B_0A_0)(BA - B_0A_0)^T]. \tag{18}$$

Since the transform of the input and output by constant orthogonal matrixes does not change E_{gen}, we can assume by the singular value decomposition that B_0A_0 is diagonal; *i.e.*

$$B_0A_0 = \begin{pmatrix} \Lambda^{(0)} & 0 \\ 0 & 0 \end{pmatrix}, \qquad \Lambda^{(0)} = \mathrm{diag}(\lambda_1^{(0)}, \ldots, \lambda_r^{(0)}), \tag{19}$$

where r is the rank of B_0A_0. Note that the true function is overrealizable if and only if $r < H$.

We employ the singular value decomposition of F;

$$F = W\Lambda U^T, \tag{20}$$

where U and W are L dimensional orthogonal matrixes, and

$$\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_L), \qquad (\lambda_1 \geq \lambda_2 \geq \cdots \lambda_L \geq 0). \tag{21}$$

We can assume that $\lambda_1 > \lambda_2 > \cdots > \lambda_L > 0$ almost surely, since F includes noise. We further assume that the singular values of B_0A_0 is sufficiently larger than the second and the third terms of eq.(12). This is satisfied in high probability if N is very large and $\sigma \leq \tau$. For simplicity, we write F as

$$F = B_0A_0 + \varepsilon Z, \qquad \varepsilon = \frac{1}{\sqrt{N}}, \tag{22}$$

where Z has the constant order, which is much larger than ε.

It is well-known that a small perturbation of a matrix causes a perturbation of the same order to the singular values. Then, the diagonal matrix Λ is decomposed as

$$\Lambda = \begin{pmatrix} \Lambda_1 & & 0 \\ & \varepsilon\tilde{\Lambda}_2 & \\ 0 & & \varepsilon\tilde{\Lambda}_3 \end{pmatrix} \tag{23}$$

where Λ_1, $\tilde{\Lambda}_2$, and $\tilde{\Lambda}_3$ are r, $H-r$, and $L-H$ dimensional matrixes of the constant order respectively, and

$$\begin{aligned} \lambda_i &= \lambda_i^{(0)} + O(\varepsilon), \qquad (1 \leq i \leq r), \\ \lambda_{r+j} &= \varepsilon\tilde{\lambda}_{r+j}, \qquad (1 \leq j \leq L-r). \end{aligned} \tag{24}$$

The purpose of this subsection is to show

$$\mathrm{E}_{gen}(t) < \mathrm{E}_{gen}(\infty) \tag{25}$$

if $r < H$ (overrealizable) and time t satisfies

$$\frac{1}{\beta\tau^2\sqrt{N}(\tilde{\lambda}_H - \tilde{\lambda}_{H+1})} \ll t \ll \frac{\log\sqrt{N}}{\beta\tau^2\sqrt{N}(\tilde{\lambda}_H - \tilde{\lambda}_{H+1})}, \tag{26}$$

or equivalently

$$\varepsilon \ll \exp\{-\beta\tau^2\sqrt{N}(\tilde{\lambda}_H - \tilde{\lambda}_{H+1})t\} \ll 1. \tag{27}$$

Since $\exp\{-\beta\tau^2\sqrt{N}(\tilde{\lambda}_H - \tilde{\lambda}_{H+1})t\} \to 0$ when $t \to \infty$, the above condition is satisfied in the middle of the learning.

The leading part of the inverted matrix in eq.(17) appears in $e^{\beta\tau^2 NSt} S^{-1} e^{\beta\tau^2 NSt}$. The solution is, then, approximately the orthogonal projection of S to a H-dimensional subspace spanned by the columns of $S^{-\frac{1}{2}} e^{\beta\tau^2 NSt} R(0)$. This converges to the eigenspace of the largest H eigenvalues of S. We will analyze the convergence elaborately to elucidate overtraining.

The diagonalization of S is given by

$$S = \Phi \begin{pmatrix} \Lambda & 0 \\ 0 & -\Lambda \end{pmatrix} \Phi^T, \text{ where } \Phi = \frac{1}{\sqrt{2}} \begin{pmatrix} U & U \\ W & -W \end{pmatrix}. \quad (28)$$

Since we assume $A(0)A(0)^T = B(0)^T B(0)$, the singular value decomposition of $R(0)$ has the form;

$$R(0) = \Theta J \Gamma G^T, \quad (29)$$

where $\Theta = \left(\begin{smallmatrix} P & 0 \\ 0 & Q \end{smallmatrix} \right)$ is an orthogonal matrix, $J = \left(\begin{smallmatrix} I_H \\ 0 \\ I_H \\ 0 \end{smallmatrix} \right)$, and $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_H)$. The subspace of the projection is

$$\begin{aligned} & S^{-\frac{1}{2}} e^{\beta\tau^2 NSt} R(0) \\ = \;& \Phi \begin{pmatrix} \Lambda^{-\frac{1}{2}} e^{\beta\tau^2 N\Lambda t} & 0 \\ 0 & \sqrt{-1}\Lambda^{-\frac{1}{2}} e^{-\beta\tau^2 N\Lambda t} \end{pmatrix} \Phi^T \Theta J \\ = \;& \Phi K \Lambda_H^{-\frac{1}{2}} \cdot e^{\beta\tau^2 N\Lambda_H t} C_H \end{aligned} \quad (30)$$

where $C = \frac{1}{\sqrt{2}}(U^T P + W^T Q) = \left(\begin{smallmatrix} C_H & * \\ C_3 & * \end{smallmatrix} \right)$, $D = \frac{1}{\sqrt{2}}(U^T P - W^T Q) = \left(\begin{smallmatrix} D_H & * \\ D_3 & * \end{smallmatrix} \right)$, $\Lambda_H = \left(\begin{smallmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{smallmatrix} \right)$, and

$$K = \begin{pmatrix} I_H \\ \Lambda_3^{-\frac{1}{2}} e^{\beta\tau^2 \sqrt{N} \tilde{\Lambda}_3 t} C_3 C_H^{-1} e^{-\beta\tau^2 N\Lambda_H t} \Lambda_H^{\frac{1}{2}} \\ \sqrt{-1}\Lambda_H^{-\frac{1}{2}} e^{-\beta\tau^2 N\Lambda_H t} D_H C_H^{-1} e^{-\beta\tau^2 N\Lambda_H t} \Lambda_H^{\frac{1}{2}} \\ \sqrt{-1}\Lambda_3^{-\frac{1}{2}} e^{-\beta\tau^2 \sqrt{N} \tilde{\Lambda}_3 t} D_3 C_H^{-1} e^{-\beta\tau^2 N\Lambda_H t} \Lambda_H^{\frac{1}{2}} \end{pmatrix}. \quad (31)$$

Using this expression, we can approximate the solution as

$$RR^T \sim 2\Phi \begin{pmatrix} \Lambda^{\frac{1}{2}} & 0 \\ 0 & \sqrt{-1}\Lambda^{\frac{1}{2}} \end{pmatrix} P_K \begin{pmatrix} \Lambda^{\frac{1}{2}} & 0 \\ 0 & \sqrt{-1}\Lambda^{\frac{1}{2}} \end{pmatrix} \Phi^T, \quad (32)$$

where $P_K = K(K^T K)^{-1} K^T$ is the orthogonal projection to the space spanned by the column vectors of K. Note that K approaches exponentially to $(I_H \; 0)^T$.

In the matrix K, the slowest order appears in the latter $H - r$ columns of the second block whose components have the order of $\exp\{-\beta\tau^2\sqrt{N}(\tilde{\lambda}_{r+j} - \tilde{\lambda}_{H+k})t\}$ (in fact, $(j,k) = (H-r, 1)$ gives the slowest). The eq.(26) asserts that the slowest order is much larger than ε. We approximate K by

$$K \sim \begin{pmatrix} I_r & 0 \\ 0 & I_{H-r} \\ 0 & K_{22} \\ \multicolumn{2}{c}{0} \end{pmatrix}. \quad (33)$$

Using this approximation, we have

$$BA \sim W\Lambda^{\frac{1}{2}} \begin{pmatrix} I_r & 0 & 0 \\ 0 & I_{H-r} - K_{22}^T K_{22} & K_{22}^T \\ 0 & K_{22} & K_{22}K_{22}^T \end{pmatrix} \Lambda^{\frac{1}{2}} U^T, \quad (34)$$

This can be considered as a *shrinkage estimator* in that the matrix norm is reduced from the MLE (we write $\hat{B}\hat{A}$) which is given by $K_{22} = 0$. Since the shrinkage occurs only in the second and third blocks which are induced by noise, $\mathrm{E}_{gen}(t)$ must be smaller than $\mathrm{E}_{gen}(\infty)$. In fact, using the fact

$$B_0 A_0 = W\Lambda^{\frac{1}{2}} \left(\left(\begin{smallmatrix} I_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix} \right) + O(\varepsilon) \right) \Lambda^{\frac{1}{2}} U^T, \quad (35)$$

the difference from the true parameter is given as follows;

$$\begin{aligned} & BA - B_0 A_0 \sim W\Lambda^{\frac{1}{2}} \\ & \left\{ \left(\begin{smallmatrix} I_r & 0 & 0 \\ 0 & I_{H-r} - K_{22}^T K_{22} & K_{22}^T \\ 0 & K_{22} & K_{22}K_{22}^T \end{smallmatrix} \right) - \left(\left(\begin{smallmatrix} I_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix} \right) + O(\varepsilon) \right) \right\} \Lambda^{\frac{1}{2}} U^T, \end{aligned} \quad (36)$$

$$\begin{aligned} & \hat{B}\hat{A} - B_0 A_0 \sim \\ & W\Lambda^{\frac{1}{2}} \left\{ \left(\begin{smallmatrix} I_r & 0 & 0 \\ 0 & I_{H-r} & 0 \\ 0 & 0 & 0 \end{smallmatrix} \right) - \left(\left(\begin{smallmatrix} I_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix} \right) + O(\varepsilon) \right) \right\} \Lambda^{\frac{1}{2}} U^T. \end{aligned} \quad (37)$$

If ε is negligible compared with K_{22}, $BA - B_0 A_0$ has the smaller matrix norm than $\hat{B}\hat{A} - B_0 A_0$. From eq.(18), this means E_{gen} of the former is smaller than that of the latter.

It is easy to see E_{gen} is decreasing if we can neglect the terms of order ε. Then, if we initialize the parameter with sufficiently large values, E_{gen} decreases at the beginning, attains smaller generalization error in the interval of eq.(26), and increases to the E_{gen} of MLE. This agrees well with the experimental results in Section 4.

6. CONCLUSION

We showed that strong overtraining is observed in batch learning of multilayer networks in overrealizable cases. From the experimental results on MLP and LNN, overtraining can be a universal property of multilayer models. We also gave a theoretical analysis of batch learning of LNN, proved the existence of overtraining in the middle of learning using the shrinkage of the estimator. Although our analysis is only on LNN, it is very suggestive to the phenomena of overtraining, which is seen in many application of multilayer networks.

References

[1] Barron, A. R. "Universal approximation bounds foe superpositions of sigmoidal function," IEEE Trans. Information Theory **39**, pp.930–945 (1993).

[2] Fukumizu, K. "A regularity condition of the information matrix of a multilayer perceptron network," Neural Networks, **9**(5), 871–879. (1996).

[3] Fukumizu, K. "Special statistical properties of neural network learning," Proc. 1997 Intern. Symp. on Nonlinear Theory and Its Applications, pp.747–750 (1997).

[4] Amari, S., Murata, N., & Müller, K. R. "Statistical theory of overtraining – is cross-validation asymptotically effective?," Advances in Neural Information Processing Systems **8**, pp.176–182. MIT Press. (1996).

[5] Baldi, P. F. & Hornik, K. "Learning in linear neural networks: a survey," IEEE Trans. Neural Networks, **6**(4), pp.837–858 (1995).

[6] Oja, E. "A simplified neuron model as a principal component analyzer," J. Math. Noiol., **15**, pp.267–273 (1989).

[7] Yan, W., Helmke, U. & Moore, J. B. "Global analysis of Oja's flow for neural networks," IEEE Trans. Neural Networks, **5**(5), pp.674–683 (1994).