Dynamics of Batch Learning in Multilayer Neural Networks

Kenji Fukumizu
The Institute of Physical and Chemical Research (RIKEN)
Wako, Saitama 351-0198, JAPAN
E-mail: fuku@brain.riken.go.jp

Abstract

We discuss the dynamics of batch learning of multilayer neural networks in the asymptotic limit, where the number of trining data is much larger than the number of parameters, emphasizing on the parameterization redundancy in overrealizable cases. In addition to showing experimental results on overtraining in multilayer perceptrons and three-layer linear neural networks, we theoretically prove the existence of overtraining in overrealizable cases of the latter model.

1 Introduction

This paper discusses the dynamics of batch gradient learning in multilayer networks. One interesting aspect in learning is *overtraining* ([1]). Although a network is trained to minimize the *empirical error* defined with finite training data, it does not ensure the decrease of the *generalization error*, the error between the trained network and the true one. Overtraining is the attainment of the minimum generalization error before the convergence of the parameter.

There is a controversy about the existence of overtraining. Many practitioners assert its existence and advocate the use of an early stopping criterion. Amari et al ([1]) theoretically show that its effect is much smaller than what is believed by practitioners, under the condition that the parameter approaches to the optimum one following the statistical asymptotic theory. However, in the case of multilayer models, the usual asymptotic theory is not applicable in *overrealizable cases*, where the true function is realized by a smaller-sized network ([2],[3]). The dynamics of learning is still an open problem in such cases.

In this paper, we investigate experimentally and theoretically the dynamics of the steepest descent learning in multilayer perceptrons and three-layer linear neural networks. Especially, we focus on the existence of overtraining in overrealizable cases, as a first step to understand learning in multilayer models.

2 Learning in three-layer networks

A three-layer network with L input, H hidden, and M output units is given by

$$f^i(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^{H} w_{ij}\, s\big(\sum_{k=1}^{L} u_{jk} x_k + \zeta_j\big) + \eta_i, \quad (1 \le i \le M), \tag{1}$$

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

In this paper, we discuss regression problems, assuming an output of the target system is observed with a noise. A sample $(\boldsymbol{x}, \boldsymbol{y})$ from the target satisfies

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

where $\boldsymbol{f}(\boldsymbol{x})$ is the *true function*, and $\boldsymbol{v}$ is a random vector whose distribution is $N(0, \sigma^2 I_M)$, a normal distribution with 0 as its mean and $\sigma^2 I_M$ as its variance-covariance matrix. An input vector $\boldsymbol{x}$ is generated randomly with its probability density function $q(\boldsymbol{x})$, which is unknown to a learner. Training data $\{(\boldsymbol{x}^{(\nu)}, \boldsymbol{y}^{(\nu)})\}_{\nu=1}^N$ are independent samples 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 $\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta}_0) = \boldsymbol{f}(\boldsymbol{x})$. If the true function $\boldsymbol{f}(\boldsymbol{x})$ is realized by a network with a smaller number of hidden units than the prepared model, we call it *overrealizable*. Otherwise, we call it *regular*. We focus on the difference of learning behaviors between these two cases.

We use the following *empirical error* as an objective function of training;

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

It is well known that the parameter that minimizes E_{emp} is equal to the maximum likelihood estimator (MLE), whose behavior for a large number of training data is given by the statistical asymptotic theory.

Generally, the MLE cannot be obtained analytically for three-layer networks, and some numerical optimization method is needed. One widely-used method is the steepest descent method, which leads the following learning rule:

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

where β is a learning rate. In this paper, we discuss only *batch learning*, in which the gradient is calculated using all training data. There are many studies on on-line learning, in which the parameter is updated each time for a newly generated data.

The performance of a network is evaluated using the generalization error:

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

The steepest descent learning tries to decrease E_{emp}, while the ideal goal is to minimize E_{gen}. The decrease of E_{emp} does not ensure the decrease of E_{gen}. Then, it is important to investigate the behavior of E_{gen} during learning.

3 Learning curves – Experimental Study –

We must be very careful in discussing experimental results on MLP, especially in overrealizable cases. Since there exist almost flat submanifolds around the global minima ([3]), the convergence of learning is extremely slow. Another

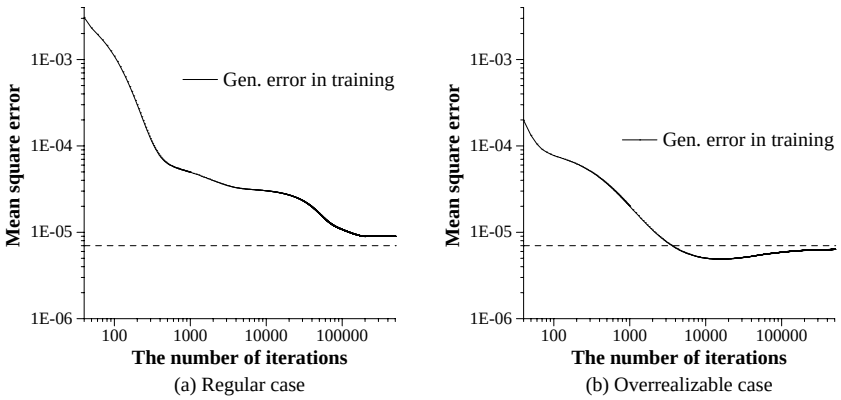


Figure 1: Average learning curves of MLP. The model in use has 1 input, 2 hidden, and 1 output unit. $N = 100$ and $\sigma = 0.01$. The dotted level is the theoretical prediction of E_{gen} of the MLE in regular cases. We use $\boldsymbol{f}(\boldsymbol{x}) = 0$ as the overrealizable target.

problem is local minima, which is common to all nonlinear models. We cannot exclude their effects, and this 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 is defined by

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

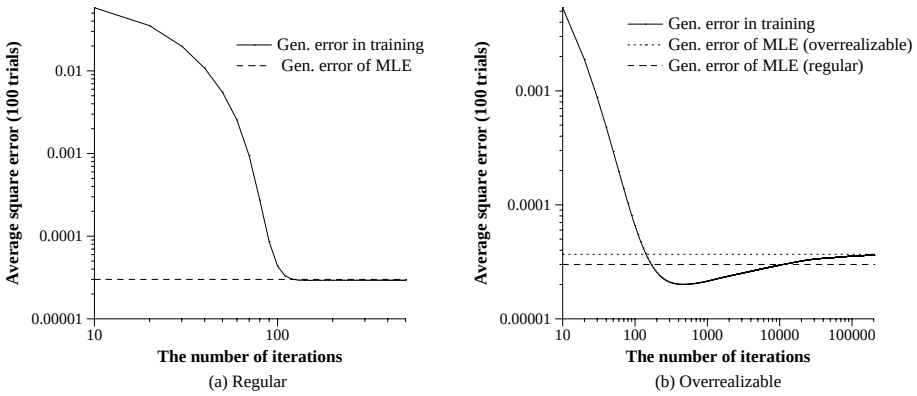
where A is a $H \times L$ matrix and B is a $M \times H$ matrix. We assume $H \leq L$ and $H \leq M$ throughout this paper. Although the function $\boldsymbol{f}(\boldsymbol{x}; A, B)$ is linear, the parameterization is quadratic, therefore, nonlinear. Note that the above model is not the same as the usual linear model $\boldsymbol{f}(\boldsymbol{x}; C) = C\boldsymbol{x}$ because of the rank restriction; the function space is the set of linear maps whose rank is no greater than H. Then, the MLE and the dynamics of learning in LNN model are different from those of the usual linear model.

We experimentally investigate the generalization error of MLP and LNN. The batch learning rule of MLP leads the well-known error back-propagation. To avoid the problems discussed above, for a fixed set of training samples, we try 30 different initializations and select the best one to give the least E_{emp} at the end, after 500000 updates. Figure 1 shows the average of generalization errors over 30 different sets of training data. It shows clear overtraining in the overrealizable case, while the regular case shows no overtraining.

Under the notations of $X = (\boldsymbol{x}^{(1)} \ldots \boldsymbol{x}^{(N)})^T$ and $Y = (\boldsymbol{y}^{(1)} \ldots \boldsymbol{y}^{(N)})^T$, the batch learning rule is given by

$$\begin{cases} A(t+1) &= A(t) + \beta B^T Y^T X - \beta B^T B A X^T X, \\ B(t+1) &= B(t) + \beta Y^T X A^T - \beta B A X^T X A^T. \end{cases} \tag{7}$$

Since the MLE of LNN is known to be solvable ([4]), the above rule is not practically used. However, our interest here is not on the MLE, but on the



(a) Regular (b) Overrealizable

Figure 2: Average learning curves of LNN. The model has 2 input, 1 hidden, and 2 output units. $N = 100$. The constant zero function is used for the overrealizable target.

dynamical behavior. Figure 2 shows the average of learning curves. The appearance of two curves are totally different. Only the overrealizable case shows sharp overtraining in the middle of learning.

From these 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, this overtraining can be an advantage over conventional linear models in that the degrade of generalization error by redundant parameters is not so critical as linear models. In the next section, we give a theory to this experimental result by obtaining the solution of a differential equation.

4 Solvable dynamics of learning in linear neural networks

We derive the solution of the continuous-time dynamics of LNN, and show the existence of overtraining in overrealizable case.

4.1 Solution of learning dynamics

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

(a) $H \leq L = M$, (b) $\boldsymbol{f}(\boldsymbol{x}) = B_0 A_0 \boldsymbol{x}$, and the rank of $B_0 A_0$ is r,
(c) $\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T] = I_L$, (d) $A(0)A(0)^T = B(0)^T B(0)$, and their rank is H.

Note that A^T and B are $L \times H$ matrixes. The assumption (d) is not restrictive, since the parameterization of eq.(6) has H^2 dimensional redundancy about the multiplication of a $H \times H$ nonsingular matrix from the left of A and from the right of B, and (d) removes the part of it.

We discuss the continuous-time differential equation instead of the discrete time update rule. If we divide the output matrix as $Y = XA_0^T B_0^T + V$, the differential equation of the steepest descent learning is given by

$$\begin{cases} \dot{A} &= \beta(B^T B_0^T A_0 X^T X + B^T V^T X - B^T B A X^T X), \\ \dot{B} &= \beta(B_0 A_0 X^t X A^T + V^T X A^T - B A X^T X A^T). \end{cases} \tag{8}$$

The behavior of eq.(8) does not coincide with that of the discrete time update rule. However, if we decide the learning coefficient β in eq.(7) sufficiently small, the solution of eq.(7) is approximately equal to that of eq.(8).

Let $Z_O := V^T X (X^T X)^{-1/2}$, and decompose $X^T X$ as $X^T X = NI_L + \sqrt{N} Z_I$. The components of Z_I and Z_O are of constant order when N is very large. Then, taking the leading terms of each order, we approximate eq.(8) as

$$\begin{cases} \dot{A} &= \beta N B^T F - \beta N B^T B A, \\ \dot{B} &= \beta N F A^T - \beta N B A A^T, \end{cases} \tag{9}$$

where $F = B_0 A_0 + \frac{1}{\sqrt{N}}(B_0 A_0 Z_I + \sigma Z_O)$. We have $AA^T = B^T B$ because of the fact $\frac{d}{dt}(AA^T) = \frac{d}{dt}(B^T B)$ and the assumption (d). If we introduce

$$R = \begin{pmatrix} A^T \\ B \end{pmatrix}, \tag{10}$$

then, R satisfies the differential equation

$$\frac{dR}{dt} = \beta N S R - \beta \frac{N}{2} R R^T R, \qquad \text{where} \quad S = \begin{pmatrix} O & F^T \\ F & O \end{pmatrix}. \tag{11}$$

This has the nonlinearity of the third order, and is very similar to Oja's learning equation ([5]), which is known to be solvable. The key to solve eq.(11) is

$$\frac{d}{dt}(RR^T) = \beta N S R R^T + \beta N R R^T S - \beta N (RR^T)^2. \tag{12}$$

This is a matrix Riccati equation, and we have the following;

Theorem 1. *Assume that the rank of $R(0)$ is full. Then, the Riccati differential equation (12) has a unique solution for all $t \geq 0$, and the solution is*

$$R(t)R^T(t) = e^{\beta N S t} R(0) \Big[I_H + \frac{1}{2} R(0)^T \{ S^{-1} e^{2\beta N S t} - S^{-1} \} R(0) \Big]^{-1} R(0)^T e^{\beta N S t}. \tag{13}$$

4.2 Dynamics of the generalization error

We can mathematically show the existence of overtraining in overrealizable cases of LNN. From the assumption (c), we have $\mathrm{E}_{gen} = \mathrm{Tr}[(BA - B_0 A_0)(BA - B_0 A_0)^T]$, which is the matrix norm of $BA - B_0 A_0$.

We assume that the positive singular values of $B_0 A_0$ are of constant order which is much larger than $\varepsilon = \frac{1}{\sqrt{N}}$. From the definition of F, the singular values of F have the perturbation of $O(\varepsilon)$ from those of $B_0 A_0$. We write $\varepsilon \tilde{\lambda}_{r+1} > \cdots > \varepsilon \tilde{\lambda}_L$ for the $L - r$ smallest singular values of F, where $\tilde{\lambda}_j$'s are of constant order. We give the following theorem without a proof.

Theorem 2 (Existence of overtraining). *Let $r < H$ (overrealizable). Under the assumptions (a)-(d) and further technical conditions, the inequality*

$$E_{gen}(t) < E_{gen}(\infty) \tag{14}$$

holds for a time t that satisfies $\varepsilon \ll \exp\{-\beta\sqrt{N}(\tilde{\lambda}_H - \tilde{\lambda}_{H+1})t\} \ll 1$.

The key of the proof is that BA is approximately the orthogonal projection of the eigen matrix of F to a H dimensional subspace, which converges to the eigenspace of the largest H singular values of F. In this convergence, the slow dynamics of $O(e^{-a\sqrt{N}t})$ induced by the eigen values of order $O(\varepsilon)$ dominates in the above time interval, and this causes the shrinkage of the redundant parameters. For more detailed explanations, see Fukumizu ([6]).

It is easy to see that E_{gen} is decreasing for a small $t > 0$ if $R(0)$ is larger than the order of $N^{-\frac{1}{2}}$. The generalization error, therefore, once decreases, and after some point, converges to the MLE as an increasing function. This means the overtraining in overrealizable cases.

5 Conclusion

We investigated the dynamical behavior of batch learning of multilayer networks in asymptotic limit. We showed experimentally that overtraining can be observed in overrealizable cases of MLP and LNN. We analyzed it theoretically, and proved the existence of overtraining in LNN. This overtraining is one of the properties caused by the redundancy of the multilayer structure ([3]). Although this paper mainly discusses only LNN, the result is suggestive to general properties of multilayer models.

References

[1] S. Amari, N. Murata, and K.R. Müller. Statistical theory of overtraining – is cross-validation asymptotically effective? In: D. S. Touretzky et al. (eds.) Advances in Neural Information Processing Systems 8, pp. 176–182. MIT Press, 1996.

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

[3] K. Fukumizu. Special statistical properties of neural network learning. In Proceedings of NOLTA'97, pp.747–750, 1997.

[4] P.F. Baldi and K. Hornik. Learning in linear neural networks: a survey. IEEE Transactions on neural networks, 6(4):837–858, 1995.

[5] W. Yan, U. Helmke, and J.B. Moore. Global analysis of Oja's flow for neural networks. IEEE Transactions on Neural Networks, 5(5):674–683, 1994.

[6] K. Fukumizu. Effect of batch learning in multilayer neural networks. In Proceedings of ICONIP'98, 1998 (to appear).