Statistical Analysis of Learning Dynamics

Noboru Murata and Shun-ichi Amari
RIKEN Brain Science Institute

Abstract

Learning is a flexible and effective means of extracting the stochastic structure of the environment. It provides an effective method for blind separation and de-convolution in signal processing. Two different types of learning are used, namely batch learning and on-line learning. The batch learning procedure uses all the training examples repeatedly so that its performance is compared to the statistical estimation procedure. On-line learning is more dynamical, updating the current estimate by observing a new datum one by one. On-line learning is slow in general but works well in the changing environment.

The present paper gives a unified framework of statistical analysis for batch and on-line learning. The topics include the asymptotic learning curve, generalization error and training error, over-fitting and over-training, efficiency of learning, and an adaptive method of determining learning rate.

keywords
learning from examples, batch learning, on-line learning, over-training, adaptive learning rate
total pages and figures

32 pages, 24 figures

authors' addresses

Noboru Murata (responsible for any correspondence)
Laboratory for Information Synthesis
RIKEN Brain Science Institute
Hirosawa 2-1, Wako
Saitama 351-0198, Japan
Phone : +81-48-467-9663
Fax : +81-48-467-9693
E-mail: mura@brain.riken.go.jp

Shun-ichi Amari
Laboratory for Information Synthesis
RIKEN Brain Science Institute
Hirosawa 2-1, Wako
Saitama 351-0198, Japan
Phone : +81-48-467-9669
Fax : +81-48-467-9687
E-mail: amari@brain.riken.go.jp
1 Introduction

Learning provides a flexible and effective means of extracting the stochastic structure of the environment, and neural networks are utilized for solving difficult signal processing problems such as nonlinear modeling and blind adaptation, because of the universal approximation property and the learning ability in supervised and unsupervised ways. In the present paper, we study statistical properties of on-line and batch learning methods, including modifiable learning rates. Although many works on the statistical analysis of adaptive signal processing have been already done (see, for example, Widrow, 1963, Widrow and Walach, 1996, Haykin, 1994, 1996), here we try to give basic statistical understanding of neural network-based adaptive signal processing taking account of the nonlinear nature of them and various learning rules derived from different kinds of loss functions.

Recent improvement of computers has made it possible to implement large scale learning systems that have a huge number of modifiable parameters on relatively small computers easily and effectively. They have been intensively used for practical applications, for example, image processing such as face recognition and finger print discrimination, and signal processing such as voice recognition, signal prediction, and independent component analysis such as blind separation and blind de-convolution (see, for example Rumelhart et al., 1986, Ripley, 1996, Cichocki and Unbehauen, 1993). The large scale learning systems, such as multi-layer perceptrons (Minsky and Papert, 1988) and radial basis functions (Poggio and Girosi, 1990), are originally motivated to mimic brain functions. They consist of one or a few kinds of relatively simple elements with a large number of modifiable connections, and their architectures are homogeneous and not complicated. In order to tune these large scale learning systems, batch learning and on-line learning methods are usually applied.

A basic procedure of batch learning is made up of gathering examples and tuning the parameters of the learning system such that a certain loss function is minimized based on these examples. This procedure is regarded as a kind of statistical inference and a conventional statistical analysis is applicable to it (White, 1989, Amari, 1990, Amari and Murata, 1993). In a statistical framework, examples are assumed to be
generated from a certain unknown probability distribution. Each example is a randomly
generated so that the parameters estimated from such examples are influenced by the
probabilistic fluctuation. We investigate an ensemble of such estimators and discuss
their characteristics and performance on average.

Over-training is an interesting problem for learning systems with a large number of
parameters. When the number of examples is not sufficiently large compared with the
number of modifiable parameters, a learning system over-fits given specific examples
and as a result some of the parameters have quite large values. In particular, when
we use an iterative procedure to train the learning system, the generalization ability is
improved in an early period of training, and then it turns worse. This is the phenomena
called over-training. We give a geometrical picture explaining the mechanism of over-
training phenomena.

On the other hand, an on-line learning procedure observes an example and updates
the parameters based on it one by one sequentially. In this case examples once used for
training are discarded and are not used again. As a special case, after observing a set of
examples like batch learning, on-line learning can be applied to the set by resampling
examples from the sample set. This method can be analyzed by combining the results
of batch and on-line learning, but we don’t discuss it particularly here.

In on-line learning, each update of the parameters is based only upon an example.
According to a probabilistic fluctuation of each example as mentioned before, the pa-
rameters have ambiguity and this ambiguity makes on-line learning less accurate than
the batch learning case. Conversely, according to this fluctuation, on-line learning can
follow the change of the environment where the learning system is placed. However,
if we control the modification strength of each update deliberately, namely reducing
the learning rate gradually, on-line learning can achieve the same accuracy with batch
learning asymptotically. Here we investigate two specific cases, the fixed learning rate
and $1/t$-annealing case, and we show that asymptotically on-line learning can be as
effective as batch learning in a peculiar $1/t$-annealing case. Based on these analyses,
we propose an adaptive learning rate algorithm which achieves $1/t$-annealing automatic-
ically without losing the ability of following the changing environment.

Also a framework of statistical mechanics gives powerful tools to investigate macro-

2 Learning from examples

Let us consider a learning system which receives a vector \( x = (x_1, \ldots, x_m) \in \mathbb{R}^m \) as an input and emits an output vector in \( \mathbb{R}^l \)

\[
f(x; \theta) = (f_1(x; \theta), \ldots, f_l(x; \theta)),
\]

where we assume that the learning system is specified by a parameter vector \( \theta = (\theta_1, \ldots, \theta_n)^T \in \mathbb{R}^n \). An example of this kind of learning system is a multi-layer perceptron

\[
f_{j_r}(x; \theta) = \sum_{j_{r-1}} w_{j_r,j_{r-1}} \phi \left( \cdots \left( \sum_{j_1} w_{j_{r+1},j_1} \phi \left( \sum_{i} w_{j_1,i} x_i + h_{j_1} \right) + h_{j_2} \right) \cdots \right) + h_{j_r},
\]

where \( \phi \) is a nonlinear activation function, such as \( \text{tanh} \), and the vector \( \theta \) represents all the modifiable weights \( w_{ji} \) and thresholds \( h_j \).

If we completely know desired system behavior which we want to realize by the learning system, we can utilize this knowledge and may find a good parameter, however, in most practical situations, we have only a partial knowledge of the desired system. Here we consider the problem of learning from examples, in which we try to find the optimal parameter by using only a set of examples observed by a stochastic system that we want to mimic by the learning system.

In order to define the optimality of the parameter, we define a pointwise loss function

\[
d(x, y; \theta),
\]

which measures the performance of the learning system \( \theta \) for a given input \( x \) and the desired output \( y \). In the following discussion, we assume that the loss \( d \) is differentiable with respect to parameter \( \theta \). In a statistical framework, examples are assumed to be
generated from a certain probability distribution \( P(x, y) \), and we define a loss function

\[
D(\theta) = E^{XY}(d(X, Y; \theta)),
\]

where \( E^{XY} \) denotes the average over all possible inputs and desired outputs, that is the expectation under the distribution \( P(x, y) \). The loss function \( D(\theta) \) is called the generalization error, since it is evaluated by the expectation of \( d(x, y; \theta) \) where the test pair \((x, y)\) is newly generated from \( P(x, y) \). By using the loss function, the optimal parameter is defined as

\[
\theta_* = \arg\min_{\theta} D(\theta). \tag{5}
\]

However, in usual cases we don’t know the distribution \( P(x, y) \) and cannot calculate the loss function. Then as a second best policy we try to minimize the empirical loss function

\[
\hat{D}(\theta) = \frac{1}{t} \sum_{i=1}^{t} d(x_i, y_i; \theta), \tag{6}
\]

where \( S_t = \{(x_1, y_1), \ldots, (x_t, y_t)\} \) are the observed i.i.d. examples generated from the distribution \( P(x, y) \). This empirical loss function is also called the training error since it is evaluated by the training examples themselves. If the distribution is not a pathological one, according to the law of large numbers the empirical loss function converges to the true loss function as the number of examples increases. Thus if an adequate large number of observed examples are given, it is expected that we have a good estimator for the optimal parameter. We define the minimal training error estimator as

\[
\hat{\theta} = \arg\min_{\theta} \hat{D}(\theta). \tag{7}
\]

A typical example is a conventional regression problem with multi-layer perceptrons. In this problem we assume that an input \( x \) and an output \( y \) satisfies the relationship

\[
y = f(x; \theta) + n,
\]

where \( f \) is represented by a multi-layer perceptron and \( n \) is a noise vector which obeys a certain probability density \( p(n) \), and from the observed sample set \( \{(x_1, y_1), \ldots, (x_t, y_t)\} \) we try to find an appropriate parameter vector \( \theta \). The squared error

\[
d(x, y; \theta) = \frac{1}{2} |y - f(x; \theta)|^2
\]
is often used as a loss function, and the estimator which is given by this loss function is called the least mean squares estimator or the minimum squared error estimator.

If the probability density of noise \( p(n) \) is known, the probability density of the learning system is written by

\[
p(x, y; \theta) = p(y - f(x; \theta))
\]

and the negative log likelihood function

\[
d(x, y; \theta) = -\log p(x, y; \theta)
\]

is often used as a loss function. In this case the estimator is called the maximum likelihood estimator and it is easy to show

\[
D(\theta) = E^{XY} (\log p(X, Y; \theta))
\]

\[
= E^{XY} (-\log p(X, Y) + \log p(X, Y) - \log p(X, Y; \theta))
\]

\[
= H_* + D [p(x, y)||p(x, y; \theta)],
\]

(8)

where \( p(x, y) \) and \( p(x, y; \theta) \) are the probability densities of the target system and the learning system respectively,

\[
H_* = -E^{XY} (\log p(X, Y))
\]

(9)

is the entropy of the target system and

\[
D [p(x, y)||p(x, y; \theta)] = E^{XY} \left( \log \frac{p(X, Y)}{p(X, Y; \theta)} \right)
\]

(10)

is the Kullback-Leibler divergence from probability density \( p(x, y) \) to \( p(x, y; \theta) \). Obviously, \( D[p(x, y)||p(x, y; \theta)] \geq 0 \) and the equality holds when and only when \( p(x, y) = p(x, y; \theta) \). Hence, the loss function \( D(\theta) \) measures the divergence between the true distribution \( p(x, y) \) and the distribution \( p(x, y; \theta) \) of the learning system except for a constant term \( H_* \) which denotes the stochastic uncertainty of the target system itself (see Cover and Thomas, 1991).

In particular, if the distribution of noise is normal, the log likelihood function can be written as

\[
d(x, y; \theta) = \frac{1}{2\sigma^2} |y - f(x; \theta)|^2 + \log \sqrt{2\pi\sigma},
\]
and it is equivalent to the squared error, and if the variance of the noise is known in this case, the maximum likelihood estimator coincides with the least mean squares estimator.

2.1 Statistical Properties and Learning Curves

To find the estimator which minimizes the empirical loss function, there are many possibilities such as the gradient descent method, the Newton-Raphson method, the quasi-Newton (variable metric) method and the conjugate gradient method. Also the on-line learning method, the behavior of which is discussed in the next section, is adopted with resampling from a given sample set to obtain an stochastic approximation of the estimator. In this section we describe stochastic properties of an ensemble of estimators which are functions of given sample sets.

When the number of examples is infinitely large and they are unbiased, the estimators converge to one of the local minima of the true loss function to be minimized. When the number of training examples is finite, the empirical loss function is different from the true loss function. Thus, since the training examples are biased in this sense, the estimators converge to a biased solution (see Fig. 1).

The asymptotic theory of statistics proves that the minimum training error estimator is asymptotically subject to the following normal distribution under regularity conditions (Murata et al., 1994).

Lemma 1 The minimum loss estimator is asymptotically subject to

\[ \hat{\theta} \sim N \left( \theta^*, \frac{1}{t} Q_s^{-1} G_s Q_s^{-1} \right) \]  

where \( G_s \) and \( Q_s \) are symmetric matrices defined by

\[ G_s = E^{XY} [\nabla d(X, Y; \theta_s) \nabla d(X, Y; \theta_s)^T] \]  

\[ = \left( E^{XY} \left( \frac{\partial}{\partial \theta_i} d(X, Y; \theta_s) \frac{\partial}{\partial \theta_j} d(X, Y; \theta_s) \right) \right) \]  

\[ Q_s = E^{XY} [\nabla \nabla d(X, Y; \theta_s)] \]  

\[ = \left( E^{XY} \left( \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} d(X, Y; \theta_s) \right) \right) , \]

and \( T \) denotes the transpose of a column vector.
Although the regularity conditions differ according to the problems (for more detailed discussion about asymptotic distributions of estimators, see, for example, Akahira and Takeuchi, 1981), from the practical point of view, differentiability of the loss $d$ up to the third order with respect to the parameter $\theta$ and existence of matrices $G_*$ and $Q_*$ are required. In particular, if we use the negative log likelihood as the loss function and the target system $p(x, y)$ is realized by the learning system $p(x, y; \theta_*)$, the maximum likelihood estimator obeys

$$\hat{\theta} \sim N\left(\theta_*; \frac{1}{t}G_*^{-1}\right)$$

because of the equality

$$E^{XY}\left[\nabla d(X, Y; \theta_*)\nabla d(X, Y; \theta_*)^T\right] = E^{XY}\left[-\nabla\nabla d(X, Y; \theta_*)\right],$$

where $G_*^{-1}$ is the inverse of the Fisher information matrix $G_*$, and this means the maximum likelihood estimator is the asymptotically efficient estimator (Lehmann, 1983).

By expanding the loss functions, we have the following asymptotic evaluations of $D(\theta)$ and $\hat{D}(\theta)$ in a neighborhood of $\theta_*$.

**Lemma 2** When $\theta$ belongs to the $(1/\sqrt{t})$-neighborhood of $\theta_*$,

$$D(\theta) = D(\theta_*) + \frac{1}{2}(\theta - \theta_*)^T G(\theta_*)(\theta - \theta_*) + O\left(t^{-3/2}\right)$$ (14)

$$\hat{D}(\theta) = \hat{D}_* - \frac{1}{2}(\hat{\theta} - \theta_*)^T G(\theta_*)(\hat{\theta} - \theta_*)$$

$$+ \frac{1}{2}(\theta - \hat{\theta})^T G(\theta_*)(\theta - \hat{\theta}) + O_p\left(t^{-3/2}\right),$$ (15)

where

$$\hat{D}_* = \frac{1}{t} \sum_{i=1}^{t} d(x_i, y_i; \theta_*) = \hat{D}(\theta_*)$$

$$E(\hat{D}_*) = D_*, \quad \hat{D}_* = D_* + O_p(1/\sqrt{t}),$$

and $E$ represents the average with respect to the distribution of the sample $S_t$.

By putting $\theta = \hat{\theta}$ in (14) and (15), we have asymptotic evaluations of the generalization and training errors of the learning system with estimator $\hat{\theta}$. They depend on the given sample sets $S_t$’s from which the estimator is calculated. We denote by
the average with respect to the distribution of the sample \( S_t \) which determines \( \hat{\theta} \).

We then obtain the following relation concerning the generalization error and training error (Amari and Murata, 1993, Müller et al., 1996). A similar but different universal property is proved by Amari (1993) for deterministic dichotomy machines. Also based on statistical mechanical method, similar properties are shown for learning systems of different architectures (Seung et al. (1992), Opper and Haussler, 1991).

**Corollary 1**

\[
E \left[ D(\hat{\theta}) \right] = D_* + \frac{1}{2t} \text{tr} G_* Q_*^{-1} + O \left( t^{-3/2} \right), \quad (16)
\]

\[
E \left[ \hat{D}(\hat{\theta}) \right] = D_* - \frac{1}{2t} \text{tr} G_* Q_*^{-1} + O \left( t^{-3/2} \right), \quad (17)
\]

where \( t \) is the number of training examples.

For the least mean squares estimator, the above corollary is rewritten as

\[
E \left[ D(\hat{\theta}) \right] = D_* \left( 1 + \frac{n^2}{2t} \right) + O \left( t^{-3/2} \right), \quad (18)
\]

\[
E \left[ \hat{D}(\hat{\theta}) \right] = D_* \left( 1 - \frac{n^2}{2t} \right) + O \left( t^{-3/2} \right) \quad (19)
\]

(see Fogel, 1991).

Based on this observation, model selection criteria which minimize generalization errors are proposed (Murata et al., 1991, 1994, Moody, 1992), which are sorts of generalization of the AIC (Akaike, 1974). Model selection problem is also discussed by Rissanen, 1986 Barron, 1993, Vapnik, 1995, and others, based on other kinds of error bounds such as MDL, structural risk minimization.

### 2.2 Over-training

If the number of training examples is not large enough compared with the number of parameters, the difference between the optimal parameter and the estimator cannot be neglected practically. The learning system fits specific examples too much, so that it loses generalization ability. This is known as over-fitting. When we use some iterative methods to find the estimator which minimizes the empirical loss function, we sometimes observe phenomena experimentally in which the generalization error decreases in
an early period of training, reaches a minimum and then increases as training goes on, while the training error monotonically decreases. This is called over-training (Hecht-Nielsen, 1989, Hassoun, 1995).

To avoid over-training, it is considered better to stop training at an adequate time. This technique is often referred to as early stopping. Here we will consider the following simple stopping rule which has been proposed based on data splitting: Divide all the available examples into two disjoint sets. One set is used for training the learning system, and the other set is used for validating the performance of the learning system. The iterative training procedure is stopped at the point that minimizes the error on the validation set. Note that dividing the available examples into two fixed sets is a strongly simplified implementation of k-fold cross-validation (cf. Efron and Tibshirani, 1993).

Here we consider the following special case of learning problems in order to have an intuitive geometrical interpretation under a special coordinate system. We assume:

- there exists a teacher specified by \( p(x, y; \theta_s) \) and it generates training examples,
- we use the negative log likelihood as the pointwise loss
  \[
  d(x, y; \theta) = -\log p(x, y; \theta),
  \]
- we use the gradient descent method to search the estimator \( \hat{\theta} \),
- and the Fisher information matrix \( G = (G_{ij}) \) defined by
  \[
  G_{ij} = E_{XY} \left( \frac{\partial}{\partial \theta_i} \log p(x, y; \theta) \frac{\partial}{\partial \theta_j} \log p(x, y; \theta) \right)
  \]
  has a full rank and is differentiable with respect to \( \theta \).

Let us consider the gradient descent learning rule, where the parameter \( \hat{\theta}(i) \) at the \( i \)-th step is modified by

\[
\hat{\theta}(i + 1) = \hat{\theta}(i) - \varepsilon \nabla D(\hat{\theta}(i)),
\]

where \( \varepsilon \) is a small positive constant. More precisely, \( \hat{\theta}(i) \) should be denoted by \( \hat{\theta}(i; S_t) \) since it depends on \( S_t \), but we omit \( S_t \) for the sake of simplicity. The learning process is deterministic and \( \hat{\theta}(i) \) converges to \( \hat{\theta} \), provided the initial \( \theta(0) \) is included in the basin
of attraction. For large $i$, $\hat{\theta}(i)$ is in the $(1/\sqrt{t})$-neighborhood of $\hat{\theta}$, and the gradient of $\hat{D}$ is approximated from (15) as

$$\nabla \hat{D}(\hat{\theta}(i)) = G_* \left\{ \hat{\theta}(i) - \hat{\theta} \right\} + O_p \left(t^{-3/2}\right).$$

Hence, by neglecting the term of order $1/t^{3/2}$, Eq. (21) is approximated by

$$\dot{\theta}(i+1) = \dot{\theta}(i) - \varepsilon G_* \left\{ \dot{\theta}(i) - \dot{\theta} \right\}.$$

This gives the asymptotic evaluation

$$\dot{\theta}(i) = (I - \varepsilon G_*)^{i-i'} \left\{ \dot{\theta}(i') - \dot{\theta} \right\} + \dot{\theta},$$

where $I$ is the identity matrix and $i' (< i)$ is assumed to be large.

In order to make the analysis easier, we take the coordinate system such that the Fisher information matrix $G_*$ is equal to the identity matrix $I$,

$$G_* = I \tag{22}$$

at $\theta_*$. This is possible without loss of generality, and the results of the following analysis are the same whichever coordinate system we use. Under this coordinate system, we have

$$\dot{\theta}(i) = (1 - \varepsilon)^{i-i'} \left\{ \dot{\theta}(i') - \dot{\theta} \right\} + \dot{\theta},$$

showing that the trajectory $\dot{\theta}(i)$ linearly approaches $\dot{\theta}$ in the neighborhood of $\dot{\theta}$.

When the trajectory $\dot{\theta}(i)$ approaches $\dot{\theta}$ as learning goes on, $i = 1, 2, \ldots$, the generalization behavior of the learning system $\dot{\theta}(i)$ is evaluated by the sequence

$$D(i) = D \left( \dot{\theta}(i) \right) , \quad i = 1, 2, \ldots \tag{23}$$

The generalization error $D(i)$ decreases in an early period of learning but it increases later. Therefore, there exists an optimal stopping time $i$ at which $D(i)$ is minimized. Let us compose a hypersphere $S$, the center of which is at $(1/2)(\theta_* + \hat{\theta})$ and which passes through both $\theta_*$ and $\hat{\theta}$, as is shown in Fig. 2. If the trajectory $\dot{\theta}(i)$ crosses the sphere $S$, the distance between $\dot{\theta}(i)$ and $\hat{\theta}$ once decreases and then increases during learning. This is a schematic mechanism of over-training. If the trajectory $\dot{\theta}(i)$ doesn’t
cross the sphere $S$, the distance monotonically decreases until the trajectory reaches the estimator $\hat{\theta}$.

In order to find the optimal stopping time for each trajectory, we divide the available examples into two disjoint sets. Let us divide $t$ examples $S_t$ into $r t$ examples of the training set $S_{tr}$ and $r' t$ examples of the validation set $S_{val}$, where

$$r + r' = 1.$$  \hspace{1cm} (24)

Let $\hat{\theta}$ be the estimator from $r t$ training examples, and let $\tilde{\theta}$ be the estimator from the other $r' t$ validation examples, that is $\hat{\theta}$ and $\tilde{\theta}$ minimize the training loss function

$$\hat{D}(\theta) = \frac{1}{r t} \sum_{(x_i, y_i) \in S_{tr}} d(x_i, y_i; \theta)$$ \hspace{1cm} (25)

and validation loss function

$$\tilde{D}(\theta) = \frac{1}{r' t} \sum_{(x_i, y_i) \in S_{val}} d(x_i, y_i; \theta)$$ \hspace{1cm} (26)

respectively. Since the training examples and validation examples are independent, both $\hat{\theta}$ and $\tilde{\theta}$ are asymptotically subject to independent normal distributions with mean $\theta_*$ and covariance matrices $1/(r t)$ and $1/(r' t)$ respectively, because we use a coordinate system such as $G_* = I$.

Let $\hat{\theta}$ be the estimator from $r t$ training examples, and let $\tilde{\theta}$ be the estimator from the other $r' t$ validation examples, that is $\hat{\theta}$ and $\tilde{\theta}$ minimize the training loss function

An interesting question is from which direction the trajectory $\hat{\theta}(i)$ approaches $\hat{\theta}$. Even if the initial $\hat{\theta}(0)$ is uniformly distributed, we cannot say that $\hat{\theta}(i)$ approaches $\hat{\theta}$ isotropically, since the dynamics in (21) are highly nonlinear in an early stage of
learning. In other words, the distribution of $\hat{\theta}(i')$ is not isotropic but may have biased directions. Although the trajectories are not isotropically distributed around $\theta_*$, the quantity $\hat{\theta}$ is isotropically distributed around $\theta_*$ because $G_*$ is put equal to $I$ at $\theta_*$. This implies that the ensemble averages $E[D(\hat{\theta}(i))]$ and $E[\hat{D}(\hat{\theta}(i))]$ are the same as those calculated by assuming that $\hat{\theta}(i)$ approaches $\hat{\theta}$ isotropically. The generalization error of $\hat{\theta}^*$ is given by (14), so that we calculate the expectation of $|\hat{\theta}^* - \theta_*|^2$ assuming that $\hat{\theta}$ and $\hat{\theta}$ are subject to normal distributions $N(\theta_*, I/rt)$ and $N(\theta_*, I/r't)$ respectively.

**Lemma 3**

$$E(|\hat{\theta}^* - \theta_*|^2) = \frac{n}{tr} - \frac{1}{2t} \left( \frac{1}{r} - \frac{1}{r'} \right),$$

where $n$ is the number of the modifiable parameters.

It is immediate to show the following.

**Lemma 4** The average generalization error by the early stopping is asymptotically

$$E\left[D(\hat{\theta}^*)\right] \simeq D_* + \frac{2n - 1}{4rt} + \frac{1}{4r't}, \quad (27)$$

We can then calculate the optimal division rate $r_{opt}$ of examples which minimizes the generalization error in sense of the ensemble average.

**Theorem 1** The average generalization error is minimized asymptotically at

$$r_{opt} = 1 - \sqrt{\frac{2n - 1}{2(n - 1)}}, \quad (28)$$

The theorem shows the optimal division of examples into training and validation sets. When $n$ is large,

$$r_{opt} \simeq 1 - \frac{1}{\sqrt{2n}} \quad (29)$$

showing that only $(1/\sqrt{2n}) \times 100\%$ of examples are to be used for validation and remaining most examples are used for training. When $n = 100$, this shows that 93% of examples are to be used for training and only 7% are to be kept for validation. From this we obtain

**Theorem 2** The asymptotically optimal generalization error of early stopping is

$$E\left[D(\hat{\theta}^*(r_{opt}))\right] \simeq D_* + \frac{1}{4t} (\sqrt{2n - 1} + 1)^2. \quad (30)$$

12
When \( n \) is large, we have

\[
E \left[ D(\hat{\theta}^*(r_{\text{opt}})) \right] \simeq D_* + \frac{n}{2t} \left( 1 + \sqrt{\frac{2}{n}} \right).
\]

This shows that the generalization error increases slightly by early stopping compared with learning which uses all the examples for training.

We have seen that early stopping is asymptotically not effective. Now we would like to discuss from several viewpoints why early stopping is effective when the number of examples is not sufficiently large. In order to have intuitive understanding, we draw another picture (Fig.4).

Let \( \theta(0) \) be the initial weight and let \( D \) be the distance between \( \theta(0) \) and \( \theta_* \). When \( \hat{\theta} \) lies in the sphere \( S \) which passes both of \( \theta(0) \) and \( \theta_* \) and whose diameter is \( D \), early stopping is not necessary, but when \( \hat{\theta} \) lies outside \( S \) then early stopping may improve the solution.

When \( D \) is large, the probability that the estimator \( \hat{\theta} \) is inside \( S \) is almost \( 1/2 \) because \( \hat{\theta} \) is distributed normally around the optimal parameter \( \theta_* \). However, when \( D \) is rather small, the probability becomes much smaller than \( 1/2 \). In particular, when the dimension number of the space in which \( S \) exists, that is the number of modifiable parameters, are large, this effect is remarkable, since most volume of \( S \) is concentrated at the equator. This implies that early stopping improves the performance with a probability close to 1. This shows that, when \( t \) is not large, we cannot neglect the distribution of the initial \( \theta(0) \) which is relatively close to the optimal parameter \( \theta_* \) (for numerical simulations and detailed discussion, see Amari et al., 1997).

It is possible to construct a theory by taking the distance between the initial value \( \theta(0) \) and the estimator \( \hat{\theta} \) into account. However, for the theory to be valid where \( t \) is not asymptotically large, the non-linear effect for learning trajectories cannot be neglected and we need higher-order corrections to the asymptotic properties of the estimator \( \hat{\theta} \) (cf. Amari, 1985).
3 On-line learning

The parameter updating rule of on-line learning is formulated as

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta F(x_t, y_t; \theta_t),$$

(32)

where vectors $x_t$ and $y_t$ are the input and the target output at time $t$. The update rule $F$ uses only information available at the time $t$. The most common update rule is the gradient of a pointwise loss function $d(x, y; \theta)$ (Rumelhart et al., 1986). In on-line learning, the learning rate $\eta$, which determines the relative strength of the update, plays a quite important role.

In on-line training all examples are presented sequentially and the parameters are updated at each presentation and all examples are used only once. Therefore on-line learning is considered to have advantages compared to batch learning with non-linear optimization methods: the low computational cost and the low memory storage. And thanks to its stochastic properties, on-line learning is thought to be less sensitive to over-fitting and have an adaptability to a changing environment. On the other hand it is thought that the accuracy of on-line learning can not be as good as batch learning.

Usually an on-line learning rule (stochastic gradient descent method) is defined as follows.

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta C \nabla d(x_t, y_t; \hat{\theta}_t),$$

(33)

where $(x_t, y_t)$ is a pair of input and desired output at time $t$. Here we note that

- the learning rate $\eta$ can be time-dependent, i.e. $\eta = \eta_t$,
- positive definite matrix $C$ can be time-dependent, i.e. $C = C_t$ or dependent on the estimated parameter $\hat{\theta}_t$, i.e. $C = C(\hat{\theta}_t)$,
- and an estimated parameter $\hat{\theta}_t$ at time $t$ is a random variable depending on a sequence of given previous examples $S_t = \{(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})\}$.

To have a good intuition of on-line learning, we calculate the ensemble mean and variance of the estimated parameters at time $t$:

$$\theta_t = E(\hat{\theta}_t), \quad V_t = \text{Cov}(\hat{\theta}_t),$$

(34)
where \( E \) and \( \text{Cov} \) denote the mean and covariance under all the possible sequences \( S_t \)’s.

If the estimated parameter is in a small neighborhood of the optimal parameter (or local-optimal parameter), the evolution of the ensemble average is approximated in a simple form by the following lemma (Amari, 1967).

**Lemma 5** If \( \eta \) is sufficiently small, the averaged value of a smooth function \( f(\theta) \) is approximated as

\[
E \left( f(\hat{\theta}_{t+1}) \right) = E \left( f(\hat{\theta}_t) \right) - \eta E \left( \nabla f(\hat{\theta}_t)^T C \nabla D(\hat{\theta}_t) \right) + \frac{\eta^2}{2} \text{tr} E \left( CG \hat{\theta}_t^2 \nabla^2 f(\hat{\theta}_t) \right) + O(\eta^3). \tag{35}
\]

By using this lemma, we approximate the ensemble mean and variance of parameters.

**Lemma 6** If the estimated parameter is in a neighborhood of the optimal parameter (or local-optimal parameter), the evolution of the mean and variance is approximated by the following recursive equations.

\[
\theta_{t+1} = \theta_t - \eta C Q_*(\theta_t - \theta_*), \tag{36}
\]

\[
V_{t+1} = V_t - \eta (C Q_* V_t + V_t Q_* C^T) + \eta^2 C G_* C^T - \eta^2 C Q_*(\theta_t - \theta_*)(\theta_t - \theta_*)^T Q_* C^T. \tag{37}
\]

Note that lemma 6 holds in an appropriate neighborhood of the optimal parameter, where the loss function \( D \) can be well approximated by quadratic form. But if we can access the gradient of the total loss function, we can have a more precise version of lemma 6. Moreover we can show that the distribution of the estimated parameter reaches the normal distribution when \( \eta \) is sufficiently small:

\[
\lim_{t \to \infty} P_\eta(\hat{\theta}_t) \simeq N(\theta_*, \eta V_*), \tag{38}
\]

where \( P_\eta(\hat{\theta}_t) \) denotes the distribution of the estimated parameter \( \hat{\theta}_t \) with fixed learning rate \( \eta \) (Murata et al., 1994, Heskes and Kappen, 1991) and \( V_* \) is given below.

Introducing notations

\[
K_* = C Q_*, \quad V_* = Q_*^{-1} G_* Q_*^{-1}, \tag{39}
\]

15
hereafter we use simplified recursive equations:

\[
\begin{align*}
\theta_{t+1} &= \theta_t - \eta K^* (\theta_t - \theta^*), \\
V_{t+1} &= V_t - \eta(K^* V_t + V_t K^*_T) + \eta^2 K^* V^*_T \\
&\quad - \eta^2 K^*(\theta_t - \theta^*)(\theta_t - \theta^*)^T K^*_T,
\end{align*}
\]

and we refer the eigenvalues of the matrix \(K^*\) as

\[
\lambda_1 \geq \cdots \geq \lambda_n > 0.
\]

In the following subsections, we discuss two specific cases:

- \(\eta\) is fixed and \(C\) is fixed,
- and \(\eta\) is controlled as \(\eta_t = 1/(t + 1)\) and \(C\) is fixed.

### 3.1 Fixed learning rate

In the case that a learning rate does not vary during the training process, i.e. \(\eta = \text{constant}\), Eqs. (40) and (41) can be solved directly.

**Theorem 3** By using the linear operators \(\Xi_A\) and \(\Omega_A\) defined by

\[
\begin{align*}
\Xi_A B &= AB + (AB)^T, \\
\Omega_A B &= ABA^T,
\end{align*}
\]

where \(A\) and \(B\) are matrices, the ensemble mean and variance of the parameter is solved as

\[
\begin{align*}
\theta_t &= \theta^* + (I - \eta K^*)^{t-1}(\theta_0 - \theta^*), \\
V_t &= [I - (I - \eta \Xi_{K^*})^{t-1}] \eta V_\infty \\
&\quad - [(I - \eta \Xi_{K^*} + \eta^2 \Omega_{K^*})^{t-1} - (I - \eta \Xi_{K^*})^{t-1}] V_0,
\end{align*}
\]

where \(\theta_0 = \theta_{t=1}\) is the initial value of the parameter, and the following matrices are used for expressing the variance:

\[
V_\infty = \Xi_{K^*}^{-1} \Omega_{K^*} V^*_T, \quad V_0 = (\theta_0 - \theta^*)(\theta_0 - \theta^*)^T.
\]
For the detailed proof of this theorem, see Amari (1967). Similarly we can do various response analysis for the case when the target function changes in time, such as step response, frequency response and ramp response.

Taking it into account that the eigenvalues of the linear operator $\Xi_A$ are represented by
\[
\lambda_i + \lambda_j, \ i, j = 1, \ldots, n, \tag{48}
\]
where $\lambda_i$’s are the eigenvalues of matrix $A$ and the eigenvalues of the linear operators $\Omega_A$ are represented by
\[
\lambda_i \lambda_j, \ i, j = 1, \ldots, n, \tag{49}
\]
we know the followings:

- the reasonable values for $\eta$ should be bounded by the inverse of the maximal eigenvalue $1/\lambda_1$ from the factor of variance,
- the convergence speed of the mean and variance of the estimated parameter is dominated by the minimal eigenvalue $\lambda_n$,
- the first term of the variance represents the remaining fluctuation of the estimated parameter around the optimal value, so that small $\eta$ is better to obtain an accurate estimator,
- and the second term of the variance represents the dependency on the initial value, so that large $\eta$ is better to have fast convergence.

Roughly speaking, if the learning rate $\eta$ is large, the speed of approaching to the optimal parameter $\theta^*$ is fast, but large fluctuation remains around the optimal parameter even if time $t$ becomes infinitely large. On the contrary, if $\eta$ is small, the fluctuations are small but the convergence of the ensemble mean of the estimator to the optimal parameter is slow.

This effect (Amari, 1967, cf. Murata et al., 1991 and Murata et al., 1993) can be seen from the viewpoint of generalization errors. By using the above result, we can calculate the asymptotic behavior of the ensemble average of generalization error at each time step.
Corollary 2 In the case of the fixed learning rate, learning curve is approximated by

\[
E \left[ D(\hat{\theta}_t) \right] = D(\theta_*) \\
+ \frac{1}{2} \eta \text{tr } Q_* V_\infty \\
+ \frac{1}{2} \text{tr } Q_*(I - \eta \Xi_{K_*})^{t-1}(V_0 - \eta V_\infty).
\] (50)

The first term of the learning curve represents the possible minimal loss of the learning system, the second term describes the additional loss caused by the stochastic property of on-line learning, and the third term shows the decreasing speed of errors through training. In the case of fixed learning rates, it is not possible to determine an universal ‘optimal’ fixed learning rate. However, if the training time is limited, we can define an ‘optimal’ rate which achieves the minimal generalization error within the time.

3.2 1/t-annealing

An effective adaptation of the learning rate is \( \eta_t = O(1/t). \) Since we can not solve the equation of variance directly, we give only the leading order of the solution.

Theorem 4 Suppose the learning rate is \( \eta_t = 1/(t + 1), \) then the ensemble mean and variance of the estimator are

\[
\theta_t = \theta_* + J_t (\theta_0 - \theta_*), \\
V_t \begin{cases} 
> O \left( \frac{1}{t} \right) & \text{if } \lambda_n < \frac{1}{2} \\
= \frac{1}{t} (\Xi_{K_*} - I)^{-1} \Omega_{K_*} V_* & \text{otherwise}
\end{cases}
\] (51)

(52)

where

\[
J_t = \prod_{k=2}^{t} (I - \frac{1}{k} K_*).
\] (53)

Knowing that

- the order of \( J_t \) is \( O(1/t^{\lambda_n}). \)
- the eigenvalues of the operator \((\Xi_{K_*} - I)^{-1} \Omega_{K_*}\) are

\[
\lambda_i \lambda_j / (\lambda_i + \lambda_j - 1), \quad i, j = 1, \ldots, n,
\]
the fastest convergence of the ensemble variance can be realized when the eigenvalues of $K_*$ are all 1, that is $C = Q_*^{-1}$. This case is the optimal setting of $1/t$-annealing and the solutions are written as follows.

**Corollary 3** The optimal case of $1/t$-annealing rate is achieved when all the eigenvalues $\lambda_i$ of $K_*$ are 1, that is, $C = Q_*^{-1}$. Then the ensemble mean and variance of the estimator satisfy

$$
\theta_t = \theta_* + \frac{1}{t}(\theta_0 - \theta_*),
$$

$$
V_t = \frac{1}{t}V_* + O\left(\frac{1}{t^2}\right).
$$

Like Cramér-Rao’s inequality, the variance of the estimator is larger than $V_*/t$ as shown in the previous section. The ensemble average of deviation from the optimal parameter can be calculated as

$$
E\left[ (\hat{\theta}_t - \theta_*) (\hat{\theta}_t - \theta_*)^T \right] = \frac{1}{t} V_* + O\left(\frac{1}{t^2}\right),
$$

and it is equivalent to the variance of batch learning in the first order of $1/t$. This optimal setting can be regarded as a stochastic version of the Newton-Raphson method. Multiplying the Hessian matrix of the loss function $Q_*^{-1}$, the search region of the stochastic gradient is changed effectively, and using $1/t$-annealing schedule for the learning rate, the convergence speed is optimized. Then on-line learning can be as effective as batch learning asymptotically.

In this specific case, we can also calculate the generalization error.

**Corollary 4**

$$
E\left[ D(\hat{\theta}_t) \right] = D(\theta_*)
+ \frac{1}{2t} \text{tr} Q_* V_*
+ O\left(\frac{1}{t^2}\right).
$$
The second term represents fluctuation caused by a finite number of examples, which is equivalent to the batch leaning case. If we use the log likelihood as the loss function, it corresponds to the Cramér-Rao bound. This also coincides with the learning curve of batch case in the first order of $t$.

### 3.3 Illustrative Example

We here illustrate the above consideration with a simple 2-source-2-observation blind separation problem. Blind source separation is the problem of separating independent sources given a mixed signal where the mixing process is unknown (for more detailed explanation, see Jutten and Herault, 1991, Comon, 1994). The following numerical simulations were done by using speech signals which were artificially mixed on the computer.

The source signals were spoken by different males and recorded with sampling rate of 16kHz (Figure 6). The signal $s_1(t)$ is a recorded word of “good morning” and $s_2(t)$ is a Japanese word “ko-n-ba-n-wa” that means “good evening”.

The source signals were mixed linearly as

$$x(t) = As(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0.7 \\ 0.3 & 1 \end{pmatrix} \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix}.$$  

The mixed signals are shown in Figure 7.

For extracting independent components from the mixed signals, we use a recurrent neural network described as

$$y(t) = x(t) - V(t)y(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} - \begin{pmatrix} 0 & v_1(t) \\ v_2(t) & 0 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}.$$  

As $y(t) = (V(t) + I)^{-1}x(t)$, where $I$ is the identity matrix, the source signals are completely extracted when $A = I + V(t)$, that is, $v_1(t) = 0.7$ and $v_2(t) = 0.3$.

We adopt the following learning rule (see Amari et al., 1997 for derivation of the algorithm and its stability analysis),

$$V(t + 1) = V(t) - \eta \left( I + V(t) \right) \left( \text{diag} \left( \phi(y(t))y(t)^T \right) - \phi(y(t))y(t)^T \right),$$  

20
where $\text{diag}(\cdot)$ makes a diagonal matrix with the diagonal elements of its argument and $\phi(y)$ is a column vector whose components are

$$\phi_i(y_i(t)) = \tanh(250 \times y_i(t)),$$

in our numerical experiment. Note that we used the matrix notation for simplicity of description, and the diagonal elements of the matrix $V(t)$ are fixed to 0 during learning.

Figure 8 shows behavior of estimated parameters and errors in time for different learning rates $\eta = 0.02, 0.01, 0.005, 0.0025$. The errors are evaluated by $l_2$ norm

$$\text{error}(t) = \sqrt{(v_1(t) - v_\ast_1)^2 + (v_2(t) - v_\ast_2)^2},$$

where $v_\ast_1$ and $v_\ast_2$ are the target values and in this case they are 0.7 and 0.3 respectively. The figure exhibits trade-off between speed and accuracy of learning. Figure 9 shows the trajectories of estimators in the parameter space. The horizontal axis is $v_1$ and the vertical axis is $v_2$. We see that fluctuation during learning is almost proportional to the learning rate as described in Theorem 3. The separated signals with the different learning rates are shown in Figure 10.

In Figures 11, 12 and 13, we used an annealed learning rate as

$$\eta = 0.02 \times \frac{1000}{1000 + t}.$$

We see that the fluctuation is stabilized by the annealing, however, we have to be careful to choose the annealing schedule. If we use too fast annealing, such as

$$\eta = 0.02 \times \frac{250}{250 + t},$$

learning doesn’t converge properly (Figure 14 (a)). Also if we use to slow annealing

$$\eta = 0.02 \times \frac{4000}{4000 + t},$$

stability of learning is not much improved (Figure 14 (b)).

## 4 Adaptive learning rate

From the results of the previous section, we have an intuitive idea about the learning rate for on-line learning. In practical applications, the learning rate might be scheduled as follows:
• when the estimator $\hat{\theta}_t$ is far from the optimal parameter $\theta^*$, use an appropriately large $\eta$.

• and when $\hat{\theta}_t$ is close to $\theta^*$, use $1/t$-annealing with an appropriate $C$.

A simple implementation is, for example,

$$\eta_t = \frac{T}{T+\frac{1}{t}}$$

where $T$ determines a switching time softly and $\eta_0$ is a initial learning rate. However, it is difficult to give a proper switching time a priori and when the rule changes over time, an annealed learning rate cannot follow the changes fast enough since $\eta_t$ is too small. Hence we need some sophisticated method to perform the above simple strategy automatically.

4.1 Sompolinsky-Barkai-Seung algorithm

The idea of an adaptively changing $\eta_t$ was called learning of the learning rule (Sompolinsky et al., 1995, Amari, 1967). The Sompolinsky-Barkai-Seung algorithm is extended to the differentiable loss function version as follows.

**Definition 1**

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta_t Q^{-1}(\hat{\theta}_t) \nabla d(x_t; y_t; \hat{\theta}_t),$$

$$\eta_{t+1} = \eta_t + \alpha \eta_t \left[ \beta \left( d(x_t; y_t; \hat{\theta}_t) - D^* \right) - \eta_t \right],$$

where $\alpha$ and $\beta$ are constant, $Q(\hat{\theta}_t)$ is the Hessian matrix of the expected loss function $\nabla \nabla D(\hat{\theta}_t)$ and $D^*$ is the minimal loss function $D(\theta^*)$.

Intuitively speaking, the coefficient $\eta$ in Eq.(58) is controlled by the remaining error. When the error is large, $\eta$ takes a relatively large value. When the error is small, it means that the estimator is close to the optimal parameter and $\eta$ approaches to 0 automatically.

To have an intuition of the dynamical behavior of $(\hat{\theta}_t, \eta_t)$, we consider the continuous version of the equations averaged with respect to all the possible input-output pairs $(x_t, y_t)$. 

22
Let us think about a continuous version of the algorithm:

\[ \frac{d}{dt} \theta(t) = -\eta(t) Q(\theta(t))^{-1} E_{X,Y} \left[ \nabla d(X, Y; \theta(t)) \right], \]

\[ \frac{d}{dt} \eta(t) = \alpha \eta(t) \left\{ \beta E_{X,Y} \left[ d(X, Y; \theta(t)) - D_\star \right] - \eta(t) \right\}. \]

When the estimator \( \theta(t) \) is in the neighborhood of the optimal parameter,

\[ E_{X,Y} \left[ \nabla d(X, Y; \theta_\star) \right] = 0, \]

\[ E_{X,Y} \left[ \nabla d(X, Y; \theta(t)) \right] \simeq Q_\star \left\{ \theta(t) - \theta_\star \right\}, \]

\[ E_{X,Y} \left[ d(X, Y; \theta(t)) - D_\star \right] \simeq \frac{1}{2} \left( \theta(t) - \theta_\star \right)^T Q_\star \left\{ \theta(t) - \theta_\star \right\}, \]

hold, where \( Q_\star = Q(\theta_\star) = \nabla \nabla D(\theta_\star) \). Then the equations are rewritten as

\[ \frac{d}{dt} \theta(t) = -\eta(t) \left( \theta(t) - \theta_\star \right), \]

\[ \frac{d}{dt} \eta(t) = \alpha \eta(t) \left( \beta \theta(t) - \theta_\star \right)^T Q_\star \left\{ \theta(t) - \theta_\star \right\} - \eta(t) \right\}. \]

Moreover introducing the squared error variable

\[ e(t) = \frac{1}{2} \left( \theta(t) - \theta_\star \right)^T Q_\star \left\{ \theta(t) - \theta_\star \right\}, \]

we obtain the following equation system

\[ \begin{cases} \frac{d}{dt} e(t) = -2\eta(t)e(t), \\ \frac{d}{dt} \eta(t) = \alpha \beta \eta(t)e(t) - \alpha \eta(t)^2. \end{cases} \]

It is easy to check that the equation system has the solution:

\[ \begin{cases} e(t) = \frac{1}{\beta} \left( \frac{1}{2} - \frac{1}{\alpha} \right) \cdot \frac{1}{t} \quad (\alpha > 2), \\ \eta(t) = \frac{1}{2} \cdot \frac{1}{t}. \end{cases} \]

Therefore the learning rate is automatically annealed as \( 1/t \), if the estimator approaches to the optimal parameter.

It is important to note that this \( 1/t \)-convergence rate of the square error \( e(t) \) is the optimal order for any estimator \( \hat{\theta}_t \) that converges to \( \theta_\star \). Thus Eq.(62) and (63) give us an adaptive on-line learning algorithm which converges with fast rate. This holds also if the target rule is slowly fluctuating or suddenly changing.
4.2 Modified algorithm

However, from the practical point of view the above algorithm has some problems such as

- the Hessian $Q_*$ of the total loss must be calculated,
- and the loss function must be calculated and the minimal value of the loss function must be known.

We consider a slightly generalized learning rule and extend the adaptation algorithm. Let us consider the following learning rule:

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta_t F(x_t, y_t; \hat{\theta}_t), \quad (67)$$

where $F$ is a flow which determines the modification when an example $(x_t, y_t)$ is given. We assume that

- the averaged flow with respect to the possible inputs and outputs vanishes at the optimal parameter, i.e. $E_{XY}[F(X, Y; \theta^*)] = 0$,
- and all the eigenvalues of matrix $K_* = E_{XY}[\nabla F(X, Y; \theta^*)]$ are positive.

We do not necessarily assume the existence of a loss function. In the case that we have a loss function, the flow corresponds to the gradient of the loss

$$F(x, y; \theta) = C\nabla d(x, y; \theta).$$

We can construct a loss function from the flow locally, but we are not assured of constructing a loss function globally. This concerns the integrability of the vector field.

Let us consider the continuous version of the update rule:

$$\frac{d}{dt}\theta(t) = -\eta(t)K_* \{\theta(t) - \theta_*\}, \quad (68)$$

where the approximation

$$\eta_tE_{XY}[F(X, Y; \theta_t)] \simeq \eta_tK_*(\theta_t - \theta_*). \quad (69)$$
is used. Suppose that we have a vector $v$ which satisfies

$$v^T K_* = \lambda v^T$$

(70)

and defining a new value:

$$\xi_t = E^{XY} [v^T F(X, Y; \theta_t)] \simeq v^T K_* (\theta_t - \theta_*),$$

(71)

Let us consider a modification of the rule for $\eta$ as:

$$\eta_{t+1} = \eta_t + \alpha \eta_t \left( \beta |\xi_t| - \eta_t \right).$$

(72)

Using the continuous version of the algorithm, we obtain the equation system

$$\begin{cases}
\frac{d}{dt} \xi(t) = -\lambda \eta(t) \xi(t), \\
\frac{d}{dt} \eta(t) = \alpha \eta(t) \left( \beta |\xi(t)| - \eta(t) \right),
\end{cases}$$

(73)

where $\xi(t) = v^T K_* (\theta(t) - \theta_*)$. Its solution is described as

$$\begin{cases}
\xi(t) = \frac{1}{\beta \left( \frac{1}{\lambda} - \frac{1}{\alpha} \right)} \cdot \frac{1}{t}, \\
\eta(t) = \frac{1}{\lambda} \cdot \frac{1}{t}.
\end{cases}$$

(74)

Intuitively $\xi$ corresponds to a 1-dimensional pseudo distance, where the average flow $F$ is projected to a single direction $v$. The idea is to choose a clever direction such that it is sufficient to observe all dynamical aspects of the flow only along this projection. In this sense the scalar $\xi$ is the simplest obtainable value to observe learning. Noting that $\xi$ is always positive or negative depending on its initial value and $\eta$ can be positive, the equation system (73) is almost equivalent to the equation system (65). Again similar to the last section we have shown that the algorithm converges properly without using a loss function nor the Hessian matrix.

In this algorithm, an important problem is how to get a good projection direction $v$. Knowing that usually the learning speed is dominated by the minimum eigenvalue, and as a result the trajectory of the estimated parameter is almost parallel to the direction of the minimum eigenvector, at the end of learning stage the averaged learning process can be seen as a one-dimensional problem, and we can use any vector as $v$ except for
the vectors which are orthogonal to the minimum eigenvector (see Fig. 15). Therefore
the averaged flow might be a good estimator of the minimum eigenvector,
\[ v = \frac{E^{XY}(F)}{\|E^{XY}(F)\|}, \]  
(75)
where \( \| \cdot \| \) is \( l_2 \) norm, and hence we can adopt
\[ \xi = \|E^{XY}(F)\|. \]  
(76)

Based on the above consideration, we propose the following practical implementa-
tion.

**Definition 2** Substituting the instantaneous average of the flow by a leaky average, we
propose a modified algorithm:

\[ \hat{\theta}_{t+1} = \hat{\theta}_t - \eta_{t+1} F(x_{t+1}, y_{t+1}; \hat{\theta}_t), \]  
(77)
\[ \eta_{t+1} = \min(\eta_0, \eta_t) \]
\[ r_{t+1} = (1 - \delta) r_t + \delta F(x_{t+1}, y_{t+1}; \hat{\theta}_t), \quad (0 < \delta < 1) \]  
(78)
\[ \eta_{t+1} = \eta_t + \alpha \eta_t (\beta \|r_{t+1}\| - \eta_t) , \]  
(79)

where \( \delta \) controls the leakiness of the average and \( r \) is used as auxiliary variable to
calculate the leaky average of the flow \( F \).

This set of rules also shows automatic \( 1/t \)-annealing asymptotically. For practical
applications of this algorithm, refer to Murata et al., 1997.

### 4.3 Illustrative Example

We again use the blind separation problem in the previous section to illustrate how the
adaptive learning rate works in practical application.

First we apply the proposed adaptive algorithm to the same problem. The learning
rule is the same in the previous example and the parameters for learning rate adaptation
are set as
\[ \alpha = 1, \quad \beta = 0.2, \quad \delta = 0.002, \quad \eta_0 = 0.01. \]
Figure 16 and Figure 17 show behavior of estimation and the trajectory of estimated parameter. We see that the convergence is fast and also fluctuation is reduced after convergence (cf. Figures 8 and 9). In Figure 18 evolution of the learning rate $\eta$ is shown in the upper plot. In the bottom plot, $1/\eta$ is shown and we see that $1/t$-type annealing is automatically done when the estimated parameter is close to the optimal parameter. Figure 19 shows separated signals.

Also we tried to apply the adaptation to the changing environment. Source signals are repeated twice, but the mixing process is changed as

$$x(t) = \begin{cases} \begin{pmatrix} 1 & 0.7 \\ 0.3 & 1 \end{pmatrix} s(t), & 0 \leq t \leq 1 \\ \begin{pmatrix} 1 & 0.4 \\ 0.2 & 1 \end{pmatrix} s(t), & 1 \leq t \leq 2 \end{cases}$$

The result is shown from Figure 20 through Figure 24. From Figure 22, we see that the learning rate is automatically increased, when the mixing process is changed. Because of this effect, the system can follow the change and successfully separate the signals with small fluctuation.

5 Conclusion

In this paper, we studied batch learning and on-line learning with a unified statistical framework.

Since in batch learning all the training examples are repeatedly used, it almost exhausts the knowledge of the given examples. We have shown its ensemble variance achieves Cramèr-Rao-like bound asymptotically.

In the case of on-line learning with a fixed learning rate we have found an exponential decay. More interesting is the case of the annealing learning rate. We have shown that by the adaptation rule $\eta = 1/t$, it is possible to achieve the same convergence speed as in batch learning. Especially with Hessian matrix of the loss function, that is the second order information of the loss function, on-line learning can be as effective as batch learning.
Also we gave a theoretically motivated adaptive on-line algorithm extending the work of Sompolinsky et al. These encouraging results should further support the importance of on-line training. On-line training is especially important in practical situation such as the desired rule is non-stationary. Strategies for the adaptation of the learning rate has to be addressed and more intensive analysis has to be needed in the future.

References


## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The true loss function and the empirical loss function</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Geometrical picture to determine the optimal stopping point $\hat{\theta}^*$.</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>Optimal stopping point $\hat{\theta}^*$ by validation examples.</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>Geometrical picture of the early stopping effect for pre-asymptotic range.</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>Dependency of convergence speed on learning rate $\eta$.</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>Source signals.</td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>Mixed signals.</td>
<td>37</td>
</tr>
<tr>
<td>8</td>
<td>Behavior of estimation with different learning rates.</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>Trajectories of learning with different learning rates.</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>Separated signals with different learning rates.</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>Behavior of estimation with an annealed learning rate.</td>
<td>41</td>
</tr>
<tr>
<td>12</td>
<td>Trajectory of learning with an annealed learning rate.</td>
<td>41</td>
</tr>
<tr>
<td>13</td>
<td>Separated signals by an annealed learning rate.</td>
<td>42</td>
</tr>
<tr>
<td>14</td>
<td>Behavior of estimation with fast and slow annealed learning rates.</td>
<td>42</td>
</tr>
<tr>
<td>15</td>
<td>Last stage of on-line learning.</td>
<td>43</td>
</tr>
<tr>
<td>16</td>
<td>Behavior of estimation with an adaptive learning rate.</td>
<td>44</td>
</tr>
<tr>
<td>17</td>
<td>Trajectory of learning with an adaptive learning rates.</td>
<td>44</td>
</tr>
<tr>
<td>18</td>
<td>Behavior of an adaptive learning rates.</td>
<td>45</td>
</tr>
<tr>
<td>19</td>
<td>Separated signals by an adaptive learning rate.</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>Behavior of estimation with an adaptive learning rate.</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Trajectory of learning with an adaptive learning rates.</td>
<td>46</td>
</tr>
<tr>
<td>22</td>
<td>Behavior of an adaptive learning rates.</td>
<td>47</td>
</tr>
<tr>
<td>23</td>
<td>Mixed signals with two different matrices.</td>
<td>47</td>
</tr>
<tr>
<td>24</td>
<td>Separated signals by an adaptive learning rate.</td>
<td>48</td>
</tr>
</tbody>
</table>
Figure 1: The true loss function and the empirical loss function

Figure 2: Geometrical picture to determine the optimal stopping point $\hat{\theta}^*$. 
Figure 3: Optimal stopping point $\hat{\theta}^*$ by validation examples.

Figure 4: Geometrical picture of the early stopping effect for pre-asymptotic range.
Figure 5: Dependency of convergence speed on learning rate $\eta$. 
Figure 6: Source signals.

Figure 7: Mixed signals.
Figure 8: Behavior of estimation with different learning rates.
Figure 9: Trajectories of learning with different learning rates.
(a) $\eta = 0.02$
(b) $\eta = 0.01$
(c) $\eta = 0.005$
(d) $\eta = 0.0025$

Figure 10: Separated signals with different learning rates.
Figure 11: Behavior of estimation with an annealed learning rate.

Figure 12: Trajectory of learning with an annealed learning rate.
Figure 13: Separated signals by an annealed learning rate.

Figure 14: Behavior of estimation with fast and slow annealed learning rates.
Figure 15: Last stage of on-line learning.
Figure 16: Behavior of estimation with an adaptive learning rate.

Figure 17: Trajectory of learning with an adaptive learning rates.
Figure 18: Behavior of an adaptive learning rates.

Figure 19: Separated signals by an adaptive learning rate.
Figure 20: Behavior of estimation with an adaptive learning rate.

Figure 21: Trajectory of learning with an adaptive learning rates.
Figure 22: Behavior of an adaptive learning rates.

Figure 23: Mixed signals with two different matrices.
Figure 24: Separated signals by an adaptive learning rate.