Scientific Research Publishing

Advances in Pure Mathematics, 2018, 8, 782-791
http://www.scirp.org/journal/apm
ISSN Online: 2160-0384
ISSN Print: 2160-0368

Least Squares Method from the View Point of Deep Learning II: Generalization

Kazuyuki Fujii[1,2]

[1]International College of Arts and Sciences, Yokohama City University, Yokohama, Japan
[2]Department of Mathematical Sciences, Shibaura Institute of Technology, Saitama, Japan
Email: fujii@yokohama-cu.ac.jp

How to cite this paper: Fujii, K. (2018) Least Squares Method from the View Point of Deep Learning II: Generalization. *Advances in Pure Mathematics*, **8**, 782-791.
https://doi.org/10.4236/apm.2018.89048

Received: August 30, 2018
Accepted: September 22, 2018
Published: September 25, 2018

Abstract

The least squares method is one of the most fundamental methods in Statistics to estimate correlations among various data. On the other hand, Deep Learning is the heart of Artificial Intelligence and it is a learning method based on the least squares method, in which a parameter called learning rate plays an important role. It is in general very hard to determine its value. In this paper we generalize the preceding paper [K. Fujii: Least squares method from the view point of Deep Learning: Advances in Pure Mathematics, **8**, 485-493, 2018] and give an admissible value of the learning rate, which is easily obtained.

Keywords

Least Squares Method, Statistics, Deep Learning, Learning Rate, Gerschgorin's Theorem

1. Introduction

This paper is a sequel to the preceding paper [1].

The least squares method in Statistics plays an important role in almost all disciplines, from Natural Science to Social Science. When we want to find properties, tendencies or correlations hidden in huge and complicated data we usually employ the method. See for example [2].

On the other hand, Deep Learning is the heart of Artificial Intelligence and will become a most important field in Data Science in the near future. As to Deep Learning see for example [3]-[10].

Deep Learning may be stated as a successive learning method based on the least squares method. Therefore, to reconsider it from the view point of Deep Learning is natural and instructive. We carry out the calculation thoroughly of

the successive approximation called gradient descent sequence, in which a parameter ϵ called learning rate plays an important role.

One of main points is to determine the range of the learning rate, which is a very hard problem [8]. We showed in [1] that a difference in methods between Statistics and Deep Learning leads to different results when the learning rate changes.

We generalize the preceding results to the case of the least squares method by polynomial approximation. Our results may give a new insight to both Statistics and Data Science.

2. Least Squares Method

Let us explain the least squares method by polynomial approximation [9]. The model function $f(x)$ is a polynomial in x of degree M given by

$$f(x) = w_0 + w_1 x + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j. \tag{1}$$

For N pieces of two dimensional real data

$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

we assume that their scatter plot is given like **Figure 1**.

The coefficients of (1)

$$\boldsymbol{w} = (w_0, w_1, \cdots, w_M)^{\mathrm{T}} \tag{2}$$

must be determined by the data set (T denotes the transposition of a vector or a matrix).

For this set of data the error function is given by

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}\{y_i - f(x_i)\}^2 = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \sum_{j=0}^{M} w_j x_i^j\right)^2. \tag{3}$$

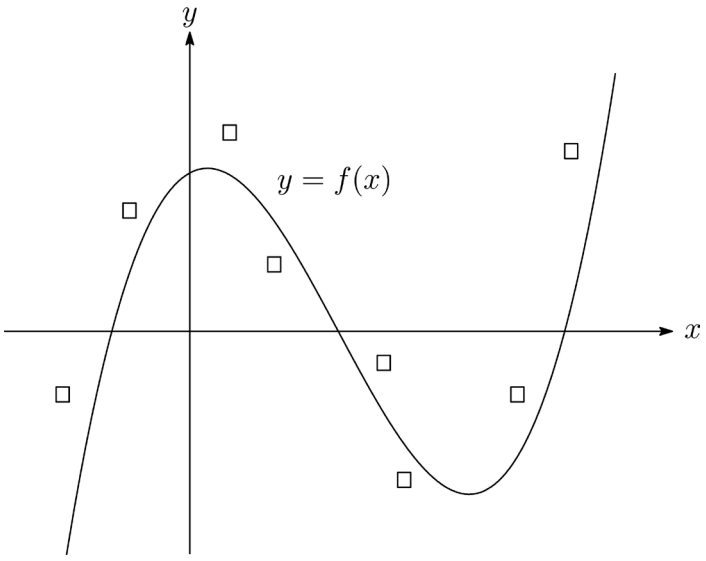

Figure 1. Scatter plot.

The aim of least squares method is to minimize the error function (3) with respect to $\boldsymbol{w}$ in (2). Usually it is obtained by solving the simultaneous differentiable equations

$$\begin{cases} \dfrac{\partial E(\boldsymbol{w})}{\partial w_0} = 0, \\ \dfrac{\partial E(\boldsymbol{w})}{\partial w_1} = 0, \\ \vdots \\ \dfrac{\partial E(\boldsymbol{w})}{\partial w_M} = 0. \end{cases}$$

However, in this paper another approach based on quadratic form is given, which is instructive.

Let us calculate the error function (3). By using the definition of inner product

$$\langle \boldsymbol{a} | \boldsymbol{b} \rangle = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{b}$$

it is not difficult to see

$$2E(\boldsymbol{w}) = \langle \boldsymbol{y} - \Phi \boldsymbol{w} | \boldsymbol{y} - \Phi \boldsymbol{w} \rangle \tag{4}$$

where

$$\boldsymbol{y} = (y_1, y_2, \cdots, y_N)^{\mathrm{T}}, \quad \boldsymbol{w} = (w_0, w_1, \cdots, w_M)^{\mathrm{T}}$$

and

$$\Phi = \begin{pmatrix} 1 & x_1^1 & x_1^2 & \cdots & x_1^M \\ 1 & x_2^1 & x_2^2 & \cdots & x_2^M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N^1 & x_N^2 & \cdots & x_N^M \end{pmatrix}.$$

Here we make an important

Assumption $N > M+1$ and $\mathrm{rank}(\Phi) = M+1$ (full rank).

Let us deform (4). From

$$\begin{aligned} \langle \boldsymbol{y} - \Phi \boldsymbol{w} | \boldsymbol{y} - \Phi \boldsymbol{w} \rangle &= (\boldsymbol{y} - \Phi \boldsymbol{w})^{\mathrm{T}} (\boldsymbol{y} - \Phi \boldsymbol{w}) \\ &= (\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{w}^{\mathrm{T}} \Phi^{\mathrm{T}})(\boldsymbol{y} - \Phi \boldsymbol{w}) \\ &= (\boldsymbol{w}^{\mathrm{T}} \Phi^{\mathrm{T}} - \boldsymbol{y}^{\mathrm{T}})(\Phi \boldsymbol{w} - \boldsymbol{y}) \\ &= \boldsymbol{w}^{\mathrm{T}} \Phi^{\mathrm{T}} \Phi \boldsymbol{w} - \boldsymbol{w}^{\mathrm{T}} \Phi^{\mathrm{T}} \boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}} \Phi \boldsymbol{w} + \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} \end{aligned}$$

we set for simplicity

$$\boldsymbol{x} = \boldsymbol{w}, \quad A = \Phi^{\mathrm{T}} \Phi, \quad \boldsymbol{b} = \Phi^{\mathrm{T}} \boldsymbol{y}, \quad c = \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y}.$$

Namely, we have a general quadratic form

$$2E(\boldsymbol{w}) = \langle \boldsymbol{y} - \Phi \boldsymbol{w} | \boldsymbol{y} - \Phi \boldsymbol{w} \rangle = \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}} \boldsymbol{b} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{x} + c. \tag{5}$$

On the other hand, the deformation of (5) is well-known.

Formula For a symmetric and invertible matrix A ($:\exists A^{-1}$) we have

$$\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}} \boldsymbol{b} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{x} + c = \left(\boldsymbol{x} - A^{-1} \boldsymbol{b}\right)^{\mathrm{T}} A \left(\boldsymbol{x} - A^{-1} \boldsymbol{b}\right) - \boldsymbol{b}^{\mathrm{T}} A^{-1} \boldsymbol{b} + c. \tag{6}$$

The proof is easy. Since $A^{\mathrm{T}} = A \Rightarrow \left(A^{-1}\right)^{\mathrm{T}} = A^{-1}$ we obtain

$$\begin{aligned}\left(\boldsymbol{x} - A^{-1}\boldsymbol{b}\right)^{\mathrm{T}} A\left(\boldsymbol{x} - A^{-1}\boldsymbol{b}\right) &= \left(\boldsymbol{x}^{\mathrm{T}} - \boldsymbol{b}^{\mathrm{T}}A^{-1}\right)A\left(\boldsymbol{x} - A^{-1}\boldsymbol{b}\right)\\ &= \boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}A^{-1}\boldsymbol{b}\end{aligned}$$

and this gives (6).

Therefore, our case becomes

$$\begin{aligned}2E(\boldsymbol{w}) = &\left\{\boldsymbol{w} - \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y}\right\}^{\mathrm{T}}\Phi^{\mathrm{T}}\Phi\left\{\boldsymbol{w} - \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y}\right\}\\ &- \boldsymbol{y}^{\mathrm{T}}\Phi\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\end{aligned} \tag{7}$$

because $\Phi^{\mathrm{T}}\Phi$ is symmetric and invertible by the assumption.

If we choose

$$\boldsymbol{w} = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y} \tag{8}$$

then the minimum is given by

$$2E(\boldsymbol{w})_{\min} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\Phi\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{y}^{\mathrm{T}}\left\{E_N - \Phi\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\right\}\boldsymbol{y} \tag{9}$$

where E_N is the N-dimensional identity matrix.

Our method is simple and clear ("smart" in our terminology).

3. Least Squares Method from Deep Learning

In this section we reconsider the least squares method in Section 2 from the view point of Deep Learning.

First we arrange the data in Section 2 like

$$\text{Input data}: \left\{\left(1, x_j, x_j^2, \cdots, x_j^M\right) \mid 1 \le j \le N\right\}$$

$$\text{Teacher signal}: \left\{y_1, y_2, \cdots, y_N\right\}$$

and consider a simple neuron model in [11] (see **Figure 2**).

Here we use the polynomial (1) instead of the sigmoid function $z = \sigma(x)$.

In this case the square error function becomes

$$L(\boldsymbol{w}) = \frac{1}{2}\langle \boldsymbol{y} - \Phi\boldsymbol{w} \mid \boldsymbol{y} - \Phi\boldsymbol{w}\rangle = \frac{1}{2}\left(\boldsymbol{w}^{\mathrm{T}}\Phi^{\mathrm{T}}\Phi\boldsymbol{w} - \boldsymbol{w}^{\mathrm{T}}\Phi^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\Phi\boldsymbol{w} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\right). \tag{10}$$

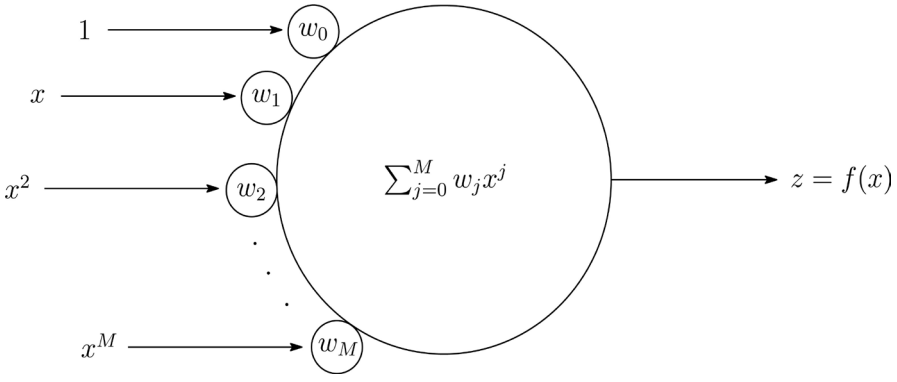

Figure 2. Simple neuron model.

We in general use $L(\boldsymbol{w})$ instead of $E(\boldsymbol{w})$ in (3).

Our aim is also to determine the parameters $\boldsymbol{w}$ in order to minimize $L(\boldsymbol{w})$. However, the procedure is different from the least squares method in Section 2. This is an important and interesting point.

The parameters $\boldsymbol{w}$ are determined successively by the gradient descent method (see for example [12]): For $t=0,1,2,\cdots$

$$\boldsymbol{w}(0)\to\boldsymbol{w}(1)\to\cdots\to\boldsymbol{w}(t)\to\boldsymbol{w}(t+1)\to\cdots$$

and

$$\boldsymbol{w}(t+1)=\boldsymbol{w}(t)-\epsilon\frac{\partial L}{\partial \boldsymbol{w}(t)} \tag{11}$$

where

$$L=L(\boldsymbol{w}(t))=\frac{1}{2}\left\{\boldsymbol{w}(t)^{\mathrm{T}}\Phi^{\mathrm{T}}\Phi\boldsymbol{w}(t)-\boldsymbol{w}(t)^{\mathrm{T}}\Phi^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{y}^{\mathrm{T}}\Phi\boldsymbol{w}(t)+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\right\} \tag{12}$$

and $\epsilon(0<\epsilon<1)$ is a small parameter called the **learning rate**.

The initial value $\boldsymbol{w}(0)$ is given appropriately. Pay attention that t is discrete time and T is the transposition.

Let us calculate (11) explicitly. Since

$$\frac{\partial L}{\partial \boldsymbol{w}(t)}=\Phi^{\mathrm{T}}\Phi\boldsymbol{w}(t)-\Phi^{\mathrm{T}}\boldsymbol{y}$$

from (12) we have

$$\boldsymbol{w}(t+1)=\boldsymbol{w}(t)-\epsilon\left\{\Phi^{\mathrm{T}}\Phi\boldsymbol{w}(t)-\Phi^{\mathrm{T}}\boldsymbol{y}\right\}=\left(E_{M+1}-\epsilon\Phi^{\mathrm{T}}\Phi\right)\boldsymbol{w}(t)+\epsilon\Phi^{\mathrm{T}}\boldsymbol{y}. \tag{13}$$

This equation is easily solved to be

$$\boldsymbol{w}(t)=\left(E_{M+1}-\epsilon\Phi^{\mathrm{T}}\Phi\right)^{t}\boldsymbol{w}(0)+\left\{E_{M+1}-\left(E_{M+1}-\epsilon\Phi^{\mathrm{T}}\Phi\right)^{t}\right\}\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y} \tag{14}$$

for $t=0,1,2,\cdots$.

The proof is left to readers.

Since this is not a final form let us continue the calculation. From (14) we have

$$\lim_{t\to\infty}\boldsymbol{w}(t)=\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\boldsymbol{y} \tag{15}$$

if

$$\lim_{t\to\infty}\left(E_{M+1}-\epsilon\Phi^{\mathrm{T}}\Phi\right)^{t}=O_{M+1} \tag{16}$$

where O_{M+1} is the N-dimensional zero matrix. (15) is just the equation (8) and it is independent of ϵ.

Let us evaluate (14) further. The matrix $\Phi^{\mathrm{T}}\Phi$ is positive definite, so all eigenvalues are positive. This can be shown as follows. Let us consider the eigenvalue equation

$$\Phi^{\mathrm{T}}\Phi\boldsymbol{v}=\lambda\boldsymbol{v}\quad(\boldsymbol{v}\neq\boldsymbol{0}).$$

Then we have

$$\lambda\langle \boldsymbol{v}|\boldsymbol{v}\rangle = \langle \lambda\boldsymbol{v}|\boldsymbol{v}\rangle = \langle \Phi^{\mathrm{T}}\Phi\boldsymbol{v}|\boldsymbol{v}\rangle = \langle \Phi\boldsymbol{v}|\Phi\boldsymbol{v}\rangle > 0 \Rightarrow \lambda > 0.$$

Therefore we can arrange all eigenvalues like

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{M+1} > 0.$$

Since $\Phi^{\mathrm{T}}\Phi$ is symmetric, it is diagonalized as

$$\Phi^{\mathrm{T}}\Phi = QDQ^{\mathrm{T}} \tag{17}$$

where Q is an element in $O(M+1)$ ($Q^{\mathrm{T}} = Q^{-1}$) and D is a diagonal matrix

$$D = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{M+1} \end{pmatrix}.$$

See for example [13].

By substituting (17) into (14) and using the equation

$$E_{M+1} - \epsilon\Phi^{\mathrm{T}}\Phi = Q\left(E_{M+1} - \epsilon D\right)Q^{\mathrm{T}} = Q\begin{pmatrix} 1-\epsilon\lambda_1 & & & \\ & 1-\epsilon\lambda_2 & & \\ & & \ddots & \\ & & & 1-\epsilon\lambda_{M+1} \end{pmatrix}Q^{\mathrm{T}}$$

we finally obtain

Theorem I A general solution to (14) is

$$\begin{aligned} \boldsymbol{w}(t) = {} & Q\begin{pmatrix} (1-\epsilon\lambda_1)^t & & & \\ & (1-\epsilon\lambda_2)^t & & \\ & & \ddots & \\ & & & (1-\epsilon\lambda_{M+1})^t \end{pmatrix}Q^{\mathrm{T}}\boldsymbol{w}(0) \\ & + Q\begin{pmatrix} \dfrac{1-(1-\epsilon\lambda_1)^t}{\lambda_1} & & & \\ & \dfrac{1-(1-\epsilon\lambda_2)^t}{\lambda_2} & & \\ & & \ddots & \\ & & & \dfrac{1-(1-\epsilon\lambda_{M+1})^t}{\lambda_{M+1}} \end{pmatrix}Q^{\mathrm{T}}\Phi^{\mathrm{T}}\boldsymbol{y}. \end{aligned} \tag{18}$$

This is our main result.

Next, let us show how to choose the learning rate $\epsilon(0 < \epsilon < 1)$, which is a very important problem in Deep Learning [7] [8].

Let us remember

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{M+1} > 0.$$

From (16) and (18) the equations

$$\lim_{t\to\infty}\left(E_{M+1} - \epsilon\Phi^{\mathrm{T}}\Phi\right)^t = O_{M+1} \Leftrightarrow \lim_{t\to\infty}\left(1-\epsilon\lambda_j\right)^t = 0 \quad \text{for } 1 \leq j \leq M+1 \tag{19}$$

determine the range of ϵ. Noting

$$|1-x|<1(\Leftrightarrow 0<x<2)\Rightarrow \lim_{n\to\infty}(1-x)^n=0$$

and

$$\epsilon\lambda_1\geq\epsilon\lambda_2\geq\cdots\geq\epsilon\lambda_{M+1}>0$$

we obtain

Theorem II The learning rate ϵ must satisfy an inequality

$$0<\epsilon\lambda_1<2\Leftrightarrow 0<\epsilon<\frac{2}{\lambda_1}. \tag{20}$$

The greater the value of ϵ, the sooner goes the gradient descent (11) so long as the convergence (19) is guaranteed. Let us note that the choice of the initial values $w(0)$ is irrelevant when the convergence condition (20) is satisfied.

Comment For example, if we choose ϵ like

$$\frac{2}{\lambda_1}<\epsilon<1$$

then we cannot recover (15), which shows a difference in methods between Statistics and Deep Learning.

4. How to Estimate the Learning Rate

How do we calculate λ_1? Since $\{\lambda_j\}$ are the eigenvalues of the matrix $\Phi^{\mathrm{T}}\Phi$, they satisfy the equation

$$F(\lambda_j)=0,\ \lambda_1\geq\lambda_2\geq\cdots\geq\lambda_{M+1}>0$$

where $F(\lambda)$ is the characteristic polynomial of $\Phi^{\mathrm{T}}\Phi$ given by

$$F(\lambda)=\left|\lambda E_{M+1}-\Phi^{\mathrm{T}}\Phi\right|. \tag{21}$$

This is abstract, so let us deform (21). For simplicity we write Φ as

$$\Phi=\begin{pmatrix}1 & x_1^1 & x_1^2 & \cdots & x_1^M\\ 1 & x_2^1 & x_2^2 & \cdots & x_2^M\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & x_N^1 & x_N^2 & \cdots & x_N^M\end{pmatrix}\equiv\left(\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(M)}\right). \tag{22}$$

Then it is easy to see

$$\Phi^{\mathrm{T}}\Phi=\left(\left\langle\boldsymbol{x}^{(i)}\middle|\boldsymbol{x}^{(j)}\right\rangle\right)_{0\leq i,j\leq M},\quad \left\langle\boldsymbol{x}^{(i)}\middle|\boldsymbol{x}^{(j)}\right\rangle=\sum_{k=1}^{N}x_k^{i+j}$$

where the notation $\langle\boldsymbol{a}|\boldsymbol{b}\rangle$ is the (real) inner product of vectors.

For clarity let us write down (21) explicitly.

$$F(\lambda)=\begin{vmatrix}\lambda-\left\langle\boldsymbol{x}^{(0)}\middle|\boldsymbol{x}^{(0)}\right\rangle & -\left\langle\boldsymbol{x}^{(0)}\middle|\boldsymbol{x}^{(1)}\right\rangle & \cdots & -\left\langle\boldsymbol{x}^{(0)}\middle|\boldsymbol{x}^{(M)}\right\rangle\\ -\left\langle\boldsymbol{x}^{(1)}\middle|\boldsymbol{x}^{(0)}\right\rangle & \lambda-\left\langle\boldsymbol{x}^{(1)}\middle|\boldsymbol{x}^{(1)}\right\rangle & \cdots & -\left\langle\boldsymbol{x}^{(1)}\middle|\boldsymbol{x}^{(M)}\right\rangle\\ \vdots & \vdots & \ddots & \vdots\\ -\left\langle\boldsymbol{x}^{(M)}\middle|\boldsymbol{x}^{(0)}\right\rangle & -\left\langle\boldsymbol{x}^{(M)}\middle|\boldsymbol{x}^{(1)}\right\rangle & \cdots & \lambda-\left\langle\boldsymbol{x}^{(M)}\middle|\boldsymbol{x}^{(M)}\right\rangle\end{vmatrix}.$$

As far as we know there is no viable method to determine the greatest root of

$F(\lambda)=0$ if M is very large[1]. Therefore, let us get satisfied by obtaining an approximate value which is both greater than λ_1 and easy to calculate.

For the purpose the Gerschgorin's theorem is very useful[2]. Let $A=(a_{ij})$ be an $n\times n$ complex (real in our case) matrix, and we set

$$R_i=\sum_{j=1,\ j\neq i}^{n}|a_{ij}| \tag{23}$$

and

$$D(a_{ii};R_i)=\{z\in\mathbf{C}\,|\,|z-a_{ii}|\leq R_i\} \tag{24}$$

for each i. This is a closed disc centered at a_{ii} with radius R_i called the Gerschgorin's disc.

Theorem (Gerschgorin [14]) For any eigenvalue λ of A we have

$$\lambda\in\bigcup_{i=1}^{n}D(a_{ii};R_i). \tag{25}$$

The proof is simple. See for example [7].

Our case is real and $n=M+1$ and

$$A=\Phi^{\mathrm{T}}\Phi=\left(\left\langle \boldsymbol{x}^{(i)}\middle|\boldsymbol{x}^{(j)}\right\rangle\right)_{0\leq i,j\leq M}.$$

Therefore, all eigenvalues $\{\lambda\}$ satisfy

$$\lambda\in\bigcup_{i=1}^{n}D(a_{ii};R_i)=\bigcup_{i=1}^{M+1}[a_{ii}-R_i,a_{ii}+R_i] \tag{26}$$

where $[A,B]$ is a closed interval and

$$a_{ii}=\left\langle \boldsymbol{x}^{(i-1)}\middle|\boldsymbol{x}^{(i-1)}\right\rangle \text{ and } R_i=\sum_{k=1,\ k\neq i}^{M+1}\left|\left\langle \boldsymbol{x}^{(i-1)}\middle|\boldsymbol{x}^{(i-1)}\right\rangle\right|.$$

If we define

$$\mathcal{F}_{M+1}=\max_{1\leq i\leq M+1}\{a_{ii}+R_i\} \tag{27}$$

then it is easy to see

$$\lambda_1\leq\mathcal{F}_{M+1}\Rightarrow\frac{2}{\mathcal{F}_{M+1}}\leq\frac{2}{\lambda_1}.$$

from (26).

Thus we arrive at an admissible value of the learning rate ϵ which is easily obtained.

Theorem III An admissible value of ϵ is

$$\epsilon=\frac{2}{\mathcal{F}_{M+1}}. \tag{28}$$

Let us show an example in the case of $M=1$ ([1]), which is very instructive for non-experts.

Example In this case it is easy to see and we set

[1] $\Phi^{\mathrm{T}}\Phi$ is not a sparse matrix.

[2] In my opinion this theorem is not so popular. Why?

$$\Phi^{\mathrm{T}}\Phi=\begin{pmatrix} N & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \end{pmatrix} \equiv \begin{pmatrix} a & x \\ x & X \end{pmatrix}$$

for simplicity. Moreover, we may assume $x>0$. Then from (21) we have

$$f(\lambda)=\left|\lambda E_2-\Phi^{\mathrm{T}}\Phi\right|=\lambda^2-(a+X)\lambda+\left(aX-x^2\right)$$

and

$$\lambda_1=\frac{a+X+\sqrt{(a+X))^2-4\left(aX-x^2\right)}}{2}$$
$$=\frac{a+X+\sqrt{(a-X)^2+4x^2}}{2}.$$

On the other hand, from (27) we have

$$\mathcal{F}_2=\max\{a+x,X+x\}$$

because $x>0$.

Then it is easy to show

$$\max\{a+x,X+x\}\geq\frac{a+X+\sqrt{(a-X)^2+4x^2}}{2}.$$

To check this inequality is left to readers. Therefore, from (28) the admissible value becomes

$$\epsilon=\frac{2}{\max\{a+x,X+x\}}.$$

We emphasize once more that $\mathcal{F}_{M+1}$ is easy to evaluate, while to calculate λ_1 is very hard if M is large.

5. Concluding Remarks

In this paper we have discussed the least squares method by polynomial approximation from the view point of Deep Learning and carried out calculation of the gradient descent thoroughly. A difference in methods between Statistics and Deep Learning delivers different results when the learning rate ϵ is changed. Theorem III is the first result to provide an admissible value of ϵ as far as we know.

Deep Learning plays an essential role in Data Science and maybe in almost all fields of Science. Therefore it is desirable for undergraduates to master it in the early stages. To master it they must study Calculus, Linear Algebra and Statistics from Mathematics. My textbook [7] is recommended.

Acknowledgements

We wishes to thank Ryu Sasaki for useful suggestions and comments.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Fujii, K. (2018) Least Squares Method from the View Point of Deep Learning. *Advances in Pure Mathematics*, **8**, 485-493.

[2] Wikipedia: Least Squares. https://en.m.wikipedia.org/wiki/Least_Squares

[3] Wikipedia: Deep Learning. https://en.m.wikipedia.org/wiki/Deep_Learning

[4] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. The MIT Press, Cambridge.

[5] Patterson, J. and Gibson, A. (2017) Deep Learning: A Practitioner's Approach, O'Reilly Media, Inc., Sebastopol.

[6] Alpaydin, J. (2014) Introduction to Machine Learning. 3rd Edition, The MIT Press, Cambridge.

[7] Fujii, K. (2018) Introduction to Mathematics for Understanding Deep Learning. Scientific Research Publishing Inc., Wuhan.

[8] Okaya, T. (2015) Deep Learning (In Japanese). Kodansha Ltd., Tokyo.

[9] Nakai, E. (2015) Introduction to Theory of Machine Learning (In Japanese). Gijutsu-Hyouronn Co., Ltd., Tokyo.

[10] Amari, S. (2016) Brain Heart Artificial Intelligence (In Japanese). Kodansha Ltd., Tokyo.

[11] Fujii, K. (2018) Mathematical Reinforcement to the Minibatch of Deep Learning. *Advances in Pure Mathematics*, **8**, 307-320. https://doi.org/10.4236/apm.2018.83016

[12] Wikipedia: Gradient Descent. https://en.m.wikipedia.org/wiki/Gradient_descent

[13] Kasahara, K. (2000) Linear Algebra (In Japanese). Saiensu Ltd., Tokyo.

[14] Gerschgorin, S. (1931) Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, **6**, 749-754.