SHARIF UNIVERSITY OF TECHNOLOGY

Deep Learning Assignment 2

Saeedreza Zouashkiani

November 21, 2022

1

1.1

Consider the Taylor expansion of the loss function at $\boldsymbol{\omega}$:

$$\boldsymbol{J}(\boldsymbol{\omega}+\delta\boldsymbol{\omega})-\boldsymbol{J}(\boldsymbol{\omega}) \approx \nabla_{\boldsymbol{\omega}}\boldsymbol{L}\delta\boldsymbol{\omega}+\frac{1}{2}\delta\boldsymbol{\omega}^T\boldsymbol{H}\delta\boldsymbol{\omega} \tag{1}$$

The first term goes to zero when the algorithm converges. So, our problem is to minimize:

$$\boldsymbol{C}=\frac{1}{2}\delta\boldsymbol{\omega}^T\boldsymbol{H}\delta\boldsymbol{\omega} \tag{2}$$

Our goal is then to set one of the weights to zero which is denoted by ω_i to minimize the increase in error given by equation (2). Then to eliminate ω_i:

$$\omega_i+\boldsymbol{e_i}^T.\delta\boldsymbol{\omega}=0 \tag{3}$$

Where $\boldsymbol{e_i}$ is a vector with a 1 corresponding to ω_i and zero elsewhere. To minimize equation (2) constrainted on equation (3), we use Lagrangian multipliers:

$$\boldsymbol{L}=\frac{1}{2}\delta\boldsymbol{\omega}^T\boldsymbol{H}\delta\boldsymbol{\omega}+\lambda(\omega_i+\boldsymbol{e_i}^T.\delta\boldsymbol{\omega}) \tag{4}$$

Differentiating with respect to $\delta\boldsymbol{\omega}$ and setting to zero:

$$\boldsymbol{H}\delta\boldsymbol{\omega}+\lambda\boldsymbol{e_i}=0 \tag{5}$$

Then if we solve for λ:

$$\lambda=\frac{\omega_i}{(\boldsymbol{H}^{-1})_{ii}} \tag{6}$$

Where $\boldsymbol{H}^{-1}$, corresponds to the i^{th} element in the inverse Hessian matrix. Then by plugging (6) into (5)

$$\delta\boldsymbol{\omega}=-\frac{\omega_i}{(\boldsymbol{H}^{-1})_{ii}}\boldsymbol{H}^{-1}.\boldsymbol{e_i} \tag{7}$$

Then plugging (7) into (2):

$$\boldsymbol{C}=\frac{1}{2}\frac{\omega_i^2}{(\boldsymbol{H}^{-1})_{ii}} \tag{8}$$

1.2

If $\boldsymbol{H} = \boldsymbol{I}$, then $\boldsymbol{H}^{-1} = \boldsymbol{I}$. Therefore:

$$\delta\boldsymbol{\omega} = -\omega_i \tag{9}$$

$$\boldsymbol{C} = \frac{1}{2}\omega_i^2 \tag{10}$$

In the previous part, we needed to compute the inverse Hessian first, and compute $\frac{1}{2}\frac{\omega_i^2}{(\boldsymbol{H}^{-1})_{ii}}$ for each element and remove the one with the least increase in the loss, but in here we compute $\frac{1}{2}\omega_i^2$ and set $\delta\boldsymbol{\omega} = -\omega_i$ for that i that minimizes the loss.

2

2.1

This is solved in part 3.1

2.2

$$\begin{aligned}
&\mathbb{E}[\frac{1}{N}||(\boldsymbol{X}(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}} - \boldsymbol{I})\boldsymbol{\epsilon}||_2^2] \\
&= \frac{1}{N}\mathbb{E}[\boldsymbol{\epsilon}^{\boldsymbol{T}}(\boldsymbol{X}(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}} - \boldsymbol{I})^{\boldsymbol{T}}(\boldsymbol{X}(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}} - \boldsymbol{I})\boldsymbol{\epsilon}] \\
&= \frac{1}{N}\mathbb{E}[\boldsymbol{\epsilon}^{\boldsymbol{T}}(\boldsymbol{B})^{\boldsymbol{T}}(\boldsymbol{B})\boldsymbol{\epsilon}]
\end{aligned} \tag{11}$$

Since $\boldsymbol{A}$ is symmetric and $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A} = \boldsymbol{A}^{\boldsymbol{2}}$, then it is idempotent. That is its eigenvalues consist of d ones and $n-d$ zeros. Therefore using 11:

$$\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B} = (\boldsymbol{A} - \boldsymbol{I})^{\boldsymbol{T}}(\boldsymbol{A} - \boldsymbol{I}) = \boldsymbol{A}^{\boldsymbol{2}} - \boldsymbol{2A} + \boldsymbol{I} = \boldsymbol{I} - \boldsymbol{A} \tag{12}$$

Since $\boldsymbol{\epsilon}^{\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{\epsilon}$ is a scalar, then its trace equals itself:

$$\begin{aligned}
&\frac{1}{N}\mathbb{E}[\boldsymbol{\epsilon}^{\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{\epsilon}] \\
&= \frac{1}{N}\mathbb{E}[\text{Trace}(\boldsymbol{\epsilon}^{\boldsymbol{T}}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{\epsilon})] \\
&= \frac{1}{N}\mathbb{E}[\text{Trace}(\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{\epsilon}^{\boldsymbol{T}}\boldsymbol{\epsilon})] \\
&= \frac{1}{N}\text{Trace}(\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}\boldsymbol{\Sigma}) \\
&= \frac{\sigma^2}{N}\text{Trace}(\boldsymbol{I} - \boldsymbol{A}) \\
&= \frac{N-d}{N}\sigma^2
\end{aligned} \tag{13}$$

Where the last part comes from the propertiy of idempotent matrices. That is Trace($\boldsymbol{A}$) = rank($\boldsymbol{X}$).

2.3

As d increases and gets closer to N, the trace of $\boldsymbol{I} - \boldsymbol{A}$ approaches zero. That is the number of independent columns of $\boldsymbol{X}$ increases, so we can have a better projection matrix than before. Naturally the training error will decrease as well.

3

3.1

We want to find:

$$
\begin{aligned}
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}} ||\boldsymbol{Y}-\boldsymbol{X\beta}||_2^2 \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\boldsymbol{Y}-\boldsymbol{X\beta})^T(\boldsymbol{Y}-\boldsymbol{X\beta}) \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} -2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}
\end{aligned} \tag{14}
$$

Define $L(\boldsymbol{\beta}) = 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$, and differentiate with respect to $\boldsymbol{\beta}$ and set to zero:

$$\frac{\partial L(\boldsymbol{\beta})}{\boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{X\beta} \tag{15}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{16}$$

3.2

Add the L2 norm of the weights to the minimization problem of the previous part:

$$
\begin{aligned}
&\underset{\boldsymbol{\beta}}{\operatorname{argmin}} ||\boldsymbol{Y}-\boldsymbol{X\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_2^2 \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} (\boldsymbol{Y}-\boldsymbol{X\beta})^T(\boldsymbol{Y}-\boldsymbol{X\beta}) + \alpha\boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} + \alpha\boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} -2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} + \alpha\boldsymbol{\beta}^T\boldsymbol{\beta}
\end{aligned} \tag{17}
$$

Define $L(\boldsymbol{\beta}) = 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} + \alpha\boldsymbol{\beta}^T\boldsymbol{\beta}$, and differentiate with respect to $\boldsymbol{\beta}$ and set to zero:

$$\frac{\partial L(\boldsymbol{\beta})}{\boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{Y} + 2\boldsymbol{X}^T\boldsymbol{X\beta} + 2\alpha\boldsymbol{\beta} \tag{18}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{19}$$

3.3

We know that both $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}$ have an expectation value of $\boldsymbol{\beta}$. Let $\boldsymbol{\beta}^* = \boldsymbol{Ay}$ and $\hat{\boldsymbol{\beta}} = \boldsymbol{By}$. Where $\boldsymbol{A} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-\mathbf{1}}\boldsymbol{X})^{-1}\boldsymbol{\Sigma}^{-\mathbf{1}}\boldsymbol{X}^{\boldsymbol{T}}$ and $\boldsymbol{B} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$.

$$
\begin{aligned}
Var(\boldsymbol{\beta}^*) &= \boldsymbol{B}Var(\boldsymbol{y})\boldsymbol{B}^T = \boldsymbol{X^T\Sigma^{-1}X}^{-1} \\
Var(\hat{\boldsymbol{\beta}}) &= \boldsymbol{A}Var(\boldsymbol{y})\boldsymbol{A}^T = \boldsymbol{(X^TX)^{-1}X^T\Sigma X(X^TX)^{-1}}
\end{aligned} \tag{20}
$$

If $\boldsymbol{XF} = \boldsymbol{\Sigma X}$, then:

$$(\boldsymbol{X^T\Sigma^{-1}X})^{-1} = \boldsymbol{F(X^TX)^{-1}} \tag{21}$$

It is obvious from 21 that $\boldsymbol{F}$ is non-singular. Using (21), we can write:

$$
\begin{aligned}
Var(\boldsymbol{\beta}^*) &= \boldsymbol{F(X^TX)^{-1}} \\
Var(\hat{\boldsymbol{\beta}}) &= \boldsymbol{(X^TX)^{-1}X^T\Sigma X(X^TX)^{-1}} = \boldsymbol{(X^TX)^{-1}X^TXF(X^TX)^{-1}} = \boldsymbol{F(X^TX)^{-1}}
\end{aligned} \tag{22}
$$

Therefore $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}$ estimate the same quantity. On the other hand, to prove when the estimators are equal, we need to show that:

$$
\begin{aligned}
\boldsymbol{\beta}^* &= \hat{\boldsymbol{\beta}} \\
(\boldsymbol{X^T\Sigma^{-1}X})^{-1} &= (\boldsymbol{X^TX})^{-1}\boldsymbol{X^T\Sigma X}(\boldsymbol{X^TX})^{-1} \\
\boldsymbol{X^TX}(\boldsymbol{X^T\Sigma^{-1}X})^{-1} &= \boldsymbol{X^T\Sigma X}(\boldsymbol{X^TX})^{-1} \\
\boldsymbol{X^T}\left(\boldsymbol{X}(\boldsymbol{X^T\Sigma^{-1}X})^{-1} - \boldsymbol{\Sigma X}(\boldsymbol{X^TX})^{-1}\right) &= 0 \\
\boldsymbol{X}(\boldsymbol{X^T\Sigma^{-1}X})^{-1} &= \boldsymbol{\Sigma X}(\boldsymbol{X^TX})^{-1} \\
\boldsymbol{X}(\boldsymbol{X^T\Sigma^{-1}X})^{-1}\boldsymbol{X^TX} &= \boldsymbol{\Sigma X} \\
\boldsymbol{XF} &= \boldsymbol{\Sigma X}
\end{aligned}
\tag{23}
$$

3.4

Extend the error function without the L-1 norm:

$$
\begin{aligned}
L(\boldsymbol{\beta},\lambda_1) &= ||\boldsymbol{Y}-\boldsymbol{X\beta}||_2^2 + \lambda_1||\boldsymbol{\beta}||_2^2 = (\boldsymbol{Y}-\boldsymbol{X\beta})^T(\boldsymbol{Y}-\boldsymbol{X\beta}) + \lambda_1\boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{Y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} + \lambda_1\boldsymbol{\beta}^T\boldsymbol{\beta}
\end{aligned}
\tag{24}
$$

Define the augmented $\hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_1}\boldsymbol{I} \end{bmatrix}$, and $\hat{\boldsymbol{Y}} = \begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{0} \end{bmatrix}$. Then equation (24) can be written as:

$$
L(\boldsymbol{\beta},\lambda_1) = ||\hat{\boldsymbol{Y}} - \hat{\boldsymbol{X}}\boldsymbol{\beta}||_2^2 \tag{25}
$$

Therefore the loss function $L(\boldsymbol{\beta},\lambda_1,\lambda_2)$ will become:

$$
L(\boldsymbol{\beta},\lambda_1,\lambda_2) = ||\hat{\boldsymbol{Y}} - \hat{\boldsymbol{X}}\boldsymbol{\beta}||_2^2 \tag{26}
$$

4

4.1

Let $J_D = (y_d - O_D)^2$ and $J_N = (y_d - O_N)^2$ denote the loss function of the Dropout and without Dropout network, respectively, and $O_D = \sum_{k=1}^n \delta_k\omega_k x_k$

$$
\begin{aligned}
\frac{\partial J_N}{\partial \omega_i} &= -2(y_d - O_N)\frac{\partial O_N}{\partial \omega_i} = -2(y_d - O_N)x_i = -2y_d x_i + 2\sum_{k=1}^n \omega_k x_k x_i \\
&= -2y_d x_i + 2\omega_i x_i^2 + 2\sum_{k=1,k\neq i}^n \omega_k x_k x_i
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
\frac{\partial J_D}{\partial \omega_i} &= -2(y_d - O_D)\frac{\partial O_D}{\partial \omega_i} = -2(y_d - O_D)\delta_i x_i \\
&= -2y_d\delta_i x_i + 2\omega_i^2\delta_i^2 x_i^2 + 2\sum_{k=1,k\neq i}^n \omega_k\delta_k\delta_i x_k x_i
\end{aligned}
\tag{28}
$$

Since equation (12) is probabilistic, we take the expectation:

$$
\begin{aligned}
\mathbb{E}[\frac{\partial J_D}{\partial \omega_i}] &= -2y_d x_i + 2\omega_i x_i^2 + 2\sigma^2\omega_i^2 x_i^2 + 2\sum_{k=1,k\neq i}^n \omega_k x_k x_i \\
&= \frac{\partial J_N}{\partial \omega_i} + 2\sigma^2\omega_i^2 x_i^2
\end{aligned}
\tag{29}
$$

4.2

The overall regularized loss function can be written as:

$$J_R = (y_d - \sum_{k=1}^{n} \delta_k \omega_k x_k)^2 + \sigma^2 \sum_{k=1}^{n} w_k^2 x_k^2 \tag{30}$$

It means that the network is regularized by the square of multiplication of weights by inputs with a regularization factor of σ^2.

5

Let the cost function be:

$$L = \frac{1}{2} \boldsymbol{\omega}^{\boldsymbol{T}} \boldsymbol{H} \boldsymbol{\omega} \tag{31}$$

Computing the first and second order derivatives:

$$\nabla_{\boldsymbol{\omega}} L = \frac{\partial L}{\partial \boldsymbol{\omega}} = \boldsymbol{H} \boldsymbol{\omega} \tag{32}$$

$$\nabla_{\boldsymbol{\omega}}^{\boldsymbol{2}} L = \frac{\partial^2 L}{\partial \boldsymbol{\omega}^{\boldsymbol{2}}} = \boldsymbol{H} \tag{33}$$

5.1

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - \epsilon \boldsymbol{H} \boldsymbol{\omega}^{(t)} = \boldsymbol{Q}(\boldsymbol{I} - \epsilon \boldsymbol{\Lambda}) \boldsymbol{Q}^{\boldsymbol{T}} \boldsymbol{\omega}^{\boldsymbol{(t)}} \tag{34}$$

5.2

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{Q}(\boldsymbol{I} - \epsilon \boldsymbol{\Lambda}) \boldsymbol{Q}^{\boldsymbol{T}} \boldsymbol{\omega}^{\boldsymbol{(0)}} \tag{35}$$

5.3

$|(\boldsymbol{I} - \epsilon \boldsymbol{\Lambda})| < \boldsymbol{1}$ must be satisfied. Therefore:

$$\epsilon < \frac{2}{\lambda_{max}} \tag{36}$$

5.4

Because of 33, the newtons method will be:

$$\boldsymbol{\omega}^{(t+1)} = (1 - \epsilon) \boldsymbol{\omega}^{(t)} \tag{37}$$

Then by choosing $\epsilon = 1$, the newton method will converge in one step.

5.5

Because in Newton's method, calculating the inverse of the Hessian matrix is expensive, gradient descent is used instead.

6

6.1

It will learn a similarity measure between 2 inputs, since the inputs are both mapped to a latent space and the distance between the 2 latent vectors is minimized. This can be used for images and text, for example.

6.2

$$\begin{aligned}
\boldsymbol{z_1} &= \boldsymbol{W}\boldsymbol{x_1} + \boldsymbol{b} \\
\boldsymbol{z_2} &= \boldsymbol{W}\boldsymbol{x_2} + \boldsymbol{b} \\
\boldsymbol{\Delta_1} &= \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{h_1}} = 2\boldsymbol{h_1} - \boldsymbol{h_2} \\
\boldsymbol{\Delta_2} &= \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{h_2}} = 2\boldsymbol{h_2} - \boldsymbol{h_1} \\
\boldsymbol{\Delta_3} &= \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{z_1}} = \boldsymbol{\Delta_1} \bigodot (\boldsymbol{1} - \boldsymbol{h_1^2}) \\
\boldsymbol{\Delta_4} &= \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{z_2}} = \boldsymbol{\Delta_2} \bigodot (\boldsymbol{1} - \boldsymbol{h_2^2}) \\
\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}} &= \boldsymbol{\Delta_3}\boldsymbol{x_1^T} + \boldsymbol{\Delta_4}\boldsymbol{x_2^T} + 2\boldsymbol{W} \\
\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{b}} &= \boldsymbol{\Delta_3} + \boldsymbol{\Delta_4}
\end{aligned} \tag{38}$$

Then the gradient descent update rule for m batch size is:

$$\begin{aligned}
\boldsymbol{W}^{(t+1)} &= \boldsymbol{W}^{(t)} - \epsilon \frac{1}{m} \sum_{i=1}^{m} \frac{\partial \boldsymbol{J}_i}{\partial \boldsymbol{W}} \\
\boldsymbol{b}^{(t+1)} &= \boldsymbol{b}^{(t)} - \epsilon \frac{1}{m} \sum_{i=1}^{m} \frac{\partial \boldsymbol{J}_i}{\partial \boldsymbol{b}}
\end{aligned} \tag{39}$$