CSC413 Homework 2 February 14, 2021

1. Optimization

1.1. Stochastic Gradient Descent (SGD)

1.1.1. Minimum Norm Solution

Recall Question 3.3 from Homework 1 that the unique minimum norm solution $\boldsymbol{w}^*$ by the gradient descent starting from zero initialization is $\boldsymbol{w}^* = X^T(XX^T)^{-1}\boldsymbol{t}$. I want to show that the SGD solution is equal to that.

First, let's find the gradient $\nabla_{w_i}\boldsymbol{\mathcal{L}}_i(\boldsymbol{x}_i, \boldsymbol{w}_t)$. The solution should belong to $\mathbb{R}$ since it calculates the gradient for a particular input. Thus, it is

$$\begin{aligned}\nabla_{w_i}\boldsymbol{\mathcal{L}}_i(\boldsymbol{x}_i, \boldsymbol{w}_t) &= \frac{\partial}{\partial \boldsymbol{w}_t}||\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{t}||_2^2 \\ &= 2(\boldsymbol{x}_i^T\boldsymbol{w}_t - t_i)\boldsymbol{x}_i\end{aligned}$$

Thus, the gradient $\nabla_{w_i}\boldsymbol{\mathcal{L}}_i(\boldsymbol{x}_i, \boldsymbol{w}_t)$ is a linear combination of $\boldsymbol{x}_i$, which is definitely contained in span of row of X. Since our SGD starts at zero initialization, we can get

$$\begin{aligned}\boldsymbol{w}_{t+1} &\leftarrow 0 - \eta\nabla_{w_i}\boldsymbol{\mathcal{L}}_i(\boldsymbol{x}_i, \boldsymbol{w}_t) \\ &= -2\eta(\boldsymbol{x}_i^T\boldsymbol{w}_t - t_i)\boldsymbol{x}_i\end{aligned}$$

Thus, for each step, we found that $\hat{\boldsymbol{w}}$ is always a linear combination of rows of X. In order to make the calculation easier, let $v \in \mathbb{R}$ and set $\boldsymbol{w} = X^Tv$.

Since $d > n$, we can get that XX^T is invertible. Thus, we have

$$\begin{aligned} X\hat{\boldsymbol{w}} - \boldsymbol{t} &= 0 \\ \implies XX^Tv - \boldsymbol{t} &= 0 \\ \implies XX^Tv &= \boldsymbol{t} \\ \implies v &= (XX^T)^{-1}\boldsymbol{t} && \text{since } XX^T \text{ is invertible} \\ \implies \hat{\boldsymbol{w}} &= X^T(XX^T)^{-1}\boldsymbol{t}\end{aligned}$$

Since $d > n$ gives that the invertibility holds, we can conclude that the solution is unique. We found that the solution is identical to the solution we obtained by gradient descent.

1.1.2. SGD with Momentum

We have know that the gradient $\nabla_{w_i}\boldsymbol{\mathcal{L}}_i(\boldsymbol{x}_i, \boldsymbol{w}_t)$ is a linear combination of rows of X from 1.1.1. Let consider it with momentum.

$$\delta_{t+1} = -2\eta(\boldsymbol{x}_i^T\boldsymbol{w}_t - t_i)\boldsymbol{x}_i + \alpha\delta_t$$

Since X is full rank and $d > n$, any vector $\in \mathbb{R}^d$ is a linear combination of rows of X. Thus, δ_t must be a linear combination of rows of X. Moreover, the momentum should be always a linear combination of rows of X since it only depends on the gradient and previous momentum. Similarly, since $\boldsymbol{w}$ is updated by the sum of the previous weight and the updated momentum, it is still a linear combination of rows of X. Thus, the momentum has no effects on the linearity of gradient update.

Thus, we can conclude that we can always find the minimum norm solution for the stochastic gradient descent with momentum on convergence.

1.2. Adaptive Methods

1.2.1. Minimum Norm Solution

Let's try the example first. Let $\boldsymbol{x} = \begin{pmatrix} x_1 & x_2 \end{pmatrix}$ and $w = \begin{pmatrix} a & b \end{pmatrix}$. The gradient of the loss should be

$$\begin{aligned}\nabla_a \boldsymbol{\mathcal{L}} &= \frac{\partial \boldsymbol{\mathcal{L}}}{\partial a} \\ &= 2(x_1 a + x_2 b - t)x_1\end{aligned}$$

$$\begin{aligned}\nabla_b \boldsymbol{\mathcal{L}} &= \frac{\partial \boldsymbol{\mathcal{L}}}{\partial b} \\ &= 2(x_1 a + x_2 b - t)x_2\end{aligned}$$

Thus, for $\boldsymbol{x}_1 = \begin{pmatrix}1 & 1\end{pmatrix}$ and $w_0 = \begin{pmatrix}0 & 0\end{pmatrix}$, we can get

$$\begin{aligned}G_{0,1} &= 0 + (-6)^2 = 36 \\ w_{0,1} &= 0 - \frac{6\eta}{6} = -\eta \\ G_{1,1} &= 0 + (-6)^2 = 36 \\ w_{1,1} &= 0 - \frac{6\eta}{6} = -\eta\end{aligned}$$

Thus, $w_1 = \begin{pmatrix}-\eta & -\eta\end{pmatrix}$ is a span of $\boldsymbol{x}_1$. However, for $\boldsymbol{x}_2 = \begin{pmatrix}1 & 2\end{pmatrix}$ and $w_0 = \begin{pmatrix}0 & 0\end{pmatrix}$, we can get

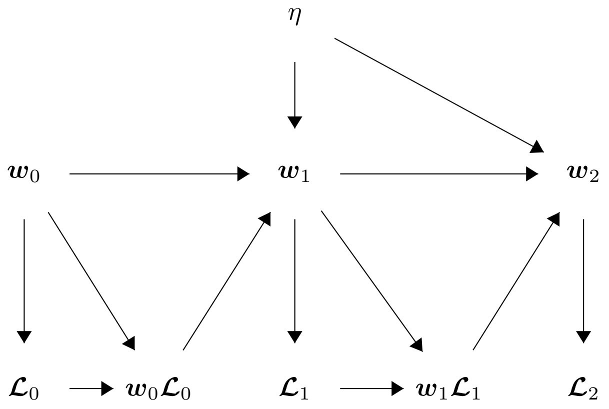
$$\begin{aligned}G_{0,1} &= 0 + (-6)^2 = 36 \\ w_{0,1} &= 0 - \frac{6\eta}{6} = -\eta \\ G_{1,1} &= 0 + (-12)^2 = 144 \\ w_{1,1} &= 0 - \frac{144\eta}{144} = -\eta\end{aligned}$$

In this case, $w_1 = \begin{pmatrix}-\eta & -\eta\end{pmatrix}$ is not a span of $\boldsymbol{x}_2$, which breaks the linearity. Thus, the adaptive method cannot always obtain the minimum norm solution.

1.2.2. (Optional)

2. Gradient-based Hyper-parameter Optimization

2.1. Computation Graph


η
$\boldsymbol{w}_0$
$\boldsymbol{w}_1$
$\boldsymbol{w}_2$
$\boldsymbol{\mathcal{L}}_0$
$\boldsymbol{w}_0\boldsymbol{\mathcal{L}}_0$
$\boldsymbol{\mathcal{L}}_1$
$\boldsymbol{w}_1\boldsymbol{\mathcal{L}}_1$
$\boldsymbol{\mathcal{L}}_2$

2.1.1.

2.1.2. Let's first figure out the memory complexity for the forward-propagation. In order to calculate the gradient, we need to store the prediction value and the actual value for n pairs of the input, which is $\mathbb{R}^n$. Since these are existing parameters, we do not need to consider them. For the updated weight and the loss, it requires d units due to its dimension. Since we do not need to store these weights and the loss values during the whole process, the memory complexity should be d. Thus, for t iterations of GD, the memory complexity for the forward-propagation is d, which is $\mathcal{O}(1)$ in terms of t.

For the back-propagation, we need to first compute $\nabla_\eta \boldsymbol{\mathcal{L}_t}$ by $\boldsymbol{w}_t$ and store it, which requires d units due to its dimension. For each iteration, we need to continue store the current gradient up to the initial weight. Thus, for t iterations of GD, the memory complexity for the standard back-propagation is dt, which is $\mathcal{O}(t)$ in terms of t.

2.1.3. (Optional)

2.2. Optimal Learning Rates

2.2.1. According to the definition of the gradient descent, we can easily know that

$$\begin{aligned}
\boldsymbol{w}_1 &= \boldsymbol{w}_0 - \eta \nabla_{\boldsymbol{w}_0} \boldsymbol{\mathcal{L}}_0 \\
&= \boldsymbol{w}_0 - \eta \frac{\partial}{\partial \boldsymbol{w}_0}(\frac{1}{n}||X\boldsymbol{w}_0 - \boldsymbol{t}||_2^2) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n} X^T(X\boldsymbol{w}_0 - \boldsymbol{t})
\end{aligned}$$

Thus, for the loss $\boldsymbol{\mathcal{L}}_1$, we can get

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_1 &= \frac{1}{n}||X\boldsymbol{w}_1 - \boldsymbol{t}||_2^2 \\
&= \frac{1}{n}||X(\boldsymbol{w}_0 - \frac{2\eta}{n}X^T(X\boldsymbol{w}_0 - \boldsymbol{t})) - \boldsymbol{t}||_2^2 \\
&= \frac{1}{n}||X\boldsymbol{w}_0 - \frac{2\eta}{n}XX^T(X\boldsymbol{w}_0 - \boldsymbol{t})) - \boldsymbol{t}||_2^2 \\
&= \frac{1}{n}||X\boldsymbol{w}_0 - \boldsymbol{t}\frac{2\eta}{n}XX^T(X\boldsymbol{w}_0 - \boldsymbol{t}))||_2^2 \\
&= \frac{1}{n}||\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a}||_2^2 \qquad \text{since } \boldsymbol{a} = X\boldsymbol{w}_0 - \boldsymbol{t}
\end{aligned}$$

2.2.2. To make the calculation easier, let's rewrite the equation of the loss $\boldsymbol{\mathcal{L}}_1$.

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_1 &= \frac{1}{n}||\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a}||_2^2 \\
&= \frac{1}{n}(\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a})^T(\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a}) \\
&= \frac{1}{n}\boldsymbol{a}^T(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^T(\boldsymbol{I} - \frac{2\eta}{n}XX^T)\boldsymbol{a} \\
&= \frac{1}{n}\boldsymbol{a}^T||\boldsymbol{I} - \frac{2\eta}{n}XX^T||_2^2\boldsymbol{a}
\end{aligned}$$

Thus, we can find the second derivative of the loss $\boldsymbol{\mathcal{L}}_1$.

$$\begin{aligned}
\frac{\partial^2 \boldsymbol{\mathcal{L}}_1}{\partial \eta^2} &= \frac{\partial}{\partial \eta}(\frac{1}{n}\boldsymbol{a}^T(\boldsymbol{I} - \frac{2\eta}{n}XX^T)\boldsymbol{a}(-\frac{2}{n}XX^T)) \\
&= \frac{\partial}{\partial \eta}(\frac{1}{n}\boldsymbol{a}^T(\boldsymbol{I} - \frac{2\eta}{n}XX^T)(-\frac{2}{n}XX^T)\boldsymbol{a}) \\
&= \frac{1}{n}\boldsymbol{a}^T(-\frac{2}{n}XX^T)(-\frac{2}{n}XX^T)\boldsymbol{a} \\
&= \frac{4}{n^3}\boldsymbol{a}^TXX^TXX^T\boldsymbol{a} \\
&= \frac{4}{n^3}||XX^T\boldsymbol{a}||_2^2 \\
&> 0
\end{aligned}$$

Thus, $\boldsymbol{\mathcal{L}}_1$ is convex with respect to the learning rate η.

2.2.3. In 2.2.2., we have computed the first derivative of $\boldsymbol{\mathcal{L}}_1$ with respect to η, which is

$$\begin{aligned}
\frac{\partial \boldsymbol{\mathcal{L}}_1}{\partial \eta} &= \frac{1}{n}\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)(-\frac{2}{n}XX^T)\boldsymbol{a} \\
&= -\frac{2}{n^2}(\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a})^T(XX^T\boldsymbol{a}) \\
&= -\frac{2}{n^2}(\boldsymbol{a}^T - \frac{2\eta}{n}\boldsymbol{a}^TXX^T)(XX^T\boldsymbol{a}) \\
&= -\frac{2}{n^2}(\boldsymbol{a}^TXX^T\boldsymbol{a} - \frac{2\eta}{n}\boldsymbol{a}^TXX^TXX^T\boldsymbol{a})
\end{aligned}$$

Set the previous equation to be 0, we can get

$$\begin{aligned}
-\frac{2}{n^2}(\boldsymbol{a}^TXX^T\boldsymbol{a} - \frac{2\eta}{n}\boldsymbol{a}^TXX^TXX^T\boldsymbol{a}) = 0 \\
\implies \boldsymbol{a}^TXX^T\boldsymbol{a} = \frac{2\eta}{n}\boldsymbol{a}^TXX^TXX^T\boldsymbol{a} \\
\implies ||X^T\boldsymbol{a}||_2^2 = \frac{2\eta}{n}||XX^T\boldsymbol{a}||_2^2 \\
\implies \eta^* = \frac{n||X^T\boldsymbol{a}||_2^2}{2||XX^T\boldsymbol{a}||_2^2}
\end{aligned}$$

Thus, the optimal learning rate η^* is $\frac{n||X^T\boldsymbol{a}||_2^2}{2||XX^T\boldsymbol{a}||_2^2}$.

2.2.4. (Optional)

2.3. Multiple Inner-loop Iterations

2.3.1. From the previous section, we know that

$$\begin{aligned}
\boldsymbol{w}_1 &= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} \\
\boldsymbol{\mathcal{L}}_1 &= \frac{1}{n}\boldsymbol{a}^T||\boldsymbol{I} - \frac{2\eta}{n}XX^T||_2^2\boldsymbol{a}
\end{aligned}$$

Let's find the expression for $\boldsymbol{w}_2$ and $\boldsymbol{\mathcal{L}}_2$.

$$\begin{aligned}
\boldsymbol{w}_2 &= \boldsymbol{w}_1 - \eta\nabla_{\boldsymbol{w}_1}\boldsymbol{\mathcal{L}}_1 \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \eta\frac{\partial}{\partial \boldsymbol{w}_1}(\frac{1}{n}||X\boldsymbol{w}_1 - \boldsymbol{t}||_2^2) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X\boldsymbol{w}_1 - \boldsymbol{t}) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X(\boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a}) - \boldsymbol{t}) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X\boldsymbol{w}_0 - \frac{2\eta}{n}XX^T\boldsymbol{a} - \boldsymbol{t}) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}X^T(\boldsymbol{a} - \frac{2\eta}{n}XX^T\boldsymbol{a}) \\
&= \boldsymbol{w}_0 - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} \\
&= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a}
\end{aligned}$$

$$\begin{aligned}\boldsymbol{\mathcal{L}}_2 &= \frac{1}{n}||X\boldsymbol{w}_2 - \boldsymbol{t}||_2^2\\ &= \frac{1}{n}||X(\boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a}) - \boldsymbol{t}||_2^2\\ &= \frac{1}{n}||X\boldsymbol{w}_0 - \frac{4\eta}{n}XX^T\boldsymbol{a} + \frac{4\eta^2}{n^2}XX^TXX^T\boldsymbol{a} - \boldsymbol{t}||_2^2\\ &= \frac{1}{n}||\boldsymbol{a} - \frac{4\eta}{n}XX^T\boldsymbol{a} + \frac{4\eta^2}{n^2}XX^TXX^T\boldsymbol{a}||_2^2\\ &= \frac{1}{n}\boldsymbol{a}^T||\boldsymbol{I} - \frac{4\eta}{n}XX^T + \frac{4\eta^2}{n^2}XX^TXX^T||_2^2\boldsymbol{a}\\ &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^2||_2^2\boldsymbol{a}\end{aligned}$$

For $\boldsymbol{w}_3$, we can get

$$\begin{aligned}\boldsymbol{w}_3 &= \boldsymbol{w}_2 - \eta\nabla_{\boldsymbol{w}_2}\boldsymbol{\mathcal{L}}_2\\ &= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \eta\frac{\partial}{\partial\boldsymbol{w}_2}(\frac{1}{n}||X\boldsymbol{w}_2 - \boldsymbol{t}||_2^2)\\ &= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X\boldsymbol{w}_2 - \boldsymbol{t})\\ &= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X(\boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a}) - \boldsymbol{t})\\ &= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{2\eta}{n}X^T(X\boldsymbol{w}_0 - \frac{4\eta}{n}XX^T\boldsymbol{a} + \frac{4\eta^2}{n^2}XX^TXX^T\boldsymbol{a} - \boldsymbol{t})\\ &= \boldsymbol{w}_0 - \frac{4\eta}{n}X^T\boldsymbol{a} + \frac{4\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{2\eta}{n}X^T\boldsymbol{a} + \frac{8\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{8\eta^3}{n^3}X^TXX^TXX^T\boldsymbol{a}\\ &= \boldsymbol{w}_0 - \frac{6\eta}{n}X^T\boldsymbol{a} + \frac{12\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{8\eta^3}{n^3}X^TXX^TXX^T\boldsymbol{a}\end{aligned}$$

Thus, the loss $\boldsymbol{\mathcal{L}}_3$ should be

$$\begin{aligned}\boldsymbol{\mathcal{L}}_3 &= \frac{1}{n}||X\boldsymbol{w}_3 - \boldsymbol{t}||_2^2\\ &= \frac{1}{n}||X(\boldsymbol{w}_0 - \frac{6\eta}{n}X^T\boldsymbol{a} + \frac{12\eta^2}{n^2}X^TXX^T\boldsymbol{a} - \frac{8\eta^3}{n^3}X^TXX^TXX^T\boldsymbol{a}) - \boldsymbol{t}||_2^2\\ &= \frac{1}{n}||\boldsymbol{a} - \frac{6\eta}{n}XX^T\boldsymbol{a} + \frac{12\eta^2}{n^2}XX^TXX^T\boldsymbol{a} - \frac{8\eta^3}{n^3}XX^TXX^TXX^T\boldsymbol{a}||_2^2\\ &= \frac{1}{n}\boldsymbol{a}^T||\boldsymbol{I} - \frac{6\eta}{n}XX^T + \frac{12\eta^2}{n^2}XX^TXX^T - \frac{8\eta^3}{n^3}XX^TXX^TXX^T||_2^2\boldsymbol{a}\\ &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^3||_2^2\boldsymbol{a}\end{aligned}$$

We can see that there is a pattern for the loss function as the iteration of the gradient descent goes up. I guess that after t iterations of GD, $\boldsymbol{\mathcal{L}}_t = \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^t||_2^2\boldsymbol{a}$. I will prove this by induction.

Proof. Let $\boldsymbol{w}_i, \boldsymbol{\mathcal{L}}_i, \nabla_{\boldsymbol{w}_i}\boldsymbol{\mathcal{L}}_i, \boldsymbol{a}, X, t$, and η be defined as above for $i \geq 0, i \in \mathbb{N}$. Define S_n be the statement that $\boldsymbol{\mathcal{L}}_n = \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n||_2^2\boldsymbol{a}$ after n iterations of gradient descent. I want to show that S_n holds by induction.
Base Case: $n = 0$.
For the case $n = 0$, we have shown that $\boldsymbol{\mathcal{L}}_0 = \frac{1}{n}||X\boldsymbol{w}_0 - \boldsymbol{t}||_2^2$. Since $\boldsymbol{a} = X\boldsymbol{w}_0 - \boldsymbol{t}$. We can

rewrite it as

$$\begin{aligned}\boldsymbol{\mathcal{L}}_0 &= \frac{1}{n}||X\boldsymbol{w}_0 - \boldsymbol{t}||_2^2 \\ &= \frac{1}{n}||\boldsymbol{a}||_2^2 \\ &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^0||_2^2\boldsymbol{a}\end{aligned}$$

Thus, S_0 holds in 2.2.1.
Induction Step: Assume S_n holds. I want to show that S_{n+1} holds.
Let's find $\boldsymbol{w}_n$ first. By the induction hypothesis, we know

$$\begin{aligned}X\boldsymbol{w}_n - \boldsymbol{t} &= \boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n \\ \implies X\boldsymbol{w}_n &= \boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n + \boldsymbol{t}\end{aligned}$$

Thus, we can find $\boldsymbol{w}_{n+1}$. By the definition of the gradient descent, we can get

$$\begin{aligned}\boldsymbol{w}_{n+1} &= \boldsymbol{w}_n - \eta\nabla_{\boldsymbol{w}_n}\boldsymbol{\mathcal{L}}_n \\ \implies X\boldsymbol{w}_{n+1} &= X\boldsymbol{w}_n - \eta X\nabla_{\boldsymbol{w}_n}\boldsymbol{\mathcal{L}}_n \\ &= \boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n - \frac{2\eta}{n}X^T\boldsymbol{a} - \eta X\frac{\partial}{\partial\boldsymbol{w}_n}(\frac{1}{n}||X\boldsymbol{w}_n - \boldsymbol{t}||_2^2) \\ &= \boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n - \frac{2\eta}{n}X^T\boldsymbol{a} - \frac{2\eta}{n}XX^T(X\boldsymbol{w}_n - \boldsymbol{t}) \\ &= \boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n + \boldsymbol{t} - \frac{2\eta}{n}XX^T\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n\end{aligned}$$

Thus, for $\boldsymbol{\mathcal{L}}_{n+1}$, we can get

$$\begin{aligned}\boldsymbol{\mathcal{L}}_{n+1} &= \frac{1}{n}||X\boldsymbol{w}_n - \boldsymbol{t}||_2^2 \\ &= \frac{1}{n}||\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n + \boldsymbol{t} - \frac{2\eta}{n}XX^T\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n||_2^2 \\ &= \frac{1}{n}||\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n - \frac{2\eta}{n}XX^T\boldsymbol{a}(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n||_2^2 \\ &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n(\boldsymbol{I} - \frac{2\eta}{n}XX^T)||_2^2\boldsymbol{a} \\ &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^{n+1}||_2^2\boldsymbol{a}\end{aligned}$$

Thus, S_{n+1} holds.
Thus, I have shown that the expression of the loss $\boldsymbol{\mathcal{L}}_n = \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^n||_2^2\boldsymbol{a}$ after n iterations of gradient descent by induction. □

2.3.2. Let's find the second derivative of the loss $\boldsymbol{\mathcal{L}}_t$. From HW1, we know that in the overparameterized case, XX^T is invertible. Let $B = \boldsymbol{I} - \frac{2\eta}{n}XX^T$, we can get that B is also invertible. Thus, there must exists an invertible matrix P consisting of the eigenvectors of the matrix B and D which is the diagonal matrix consisting of the corresponding eigenvalues of B such that $B = PDP^{-1}$.
Notice that for the derivative of a matrix to the power of some number, if the matrix is diagonalizable, the derivative of that matrix A^n should be

$$\begin{aligned}A &= PDP^{-1} \\ (A^n)^{'} &= P^{'}D^{2t}P^{-1} + 2tPD^{2t-1}D^{-1}P^{-1} - PD^{2t}P^{-1}P^{'}P^{-1}\end{aligned}$$

which is too complex. Let's rewrite the loss to the summation of the equation. Thus, it would be

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_t &= \frac{1}{n}\boldsymbol{a}^T||(\boldsymbol{I} - \frac{2\eta}{n}XX^T)^t||_2^2\boldsymbol{a} \\
&= \frac{1}{n}\boldsymbol{a}^T||(PDP^{-1})^t||_2^2\boldsymbol{a} \\
&= \frac{1}{n}\sum_{n}^{i=1}(\lambda_i)^{2t}c_i^2 \qquad \text{where } \lambda_i \text{ is the eigenvalue for the } i^{th} \text{ row} \\
&\qquad\qquad \text{and } c_i \text{ is the product of the } i^{th} \text{ row } P \text{ and } \boldsymbol{a}
\end{aligned}$$

Thus, it is much easier to compute the second derivative of the loss. Since we have proved the case for $t = 1$, let's assume $t \geq 2$, we have

$$\begin{aligned}
\frac{\partial^2\boldsymbol{\mathcal{L}}_t}{\partial\eta^2} &= \frac{\partial}{\partial\eta}(\frac{1}{n}\sum_{n}^{i=1}(\lambda_i)^{2t}c_i^2)' \\
&= \frac{\partial}{\partial\eta}(\frac{2t}{n}\sum_{n}^{i=1}(\lambda_i)^{2t-1}c_i^2(\frac{\partial\lambda_i}{\partial\eta})) \\
&= \frac{\partial}{\partial\eta}(\frac{2t}{n}\sum_{n}^{i=1}(\lambda_i)^{2t-1}c_i^2(\frac{\partial}{\partial\eta} - \frac{2\eta\lambda_i}{n})) \\
&= \frac{\partial}{\partial\eta}(-\frac{4t}{n^2}\sum_{n}^{i=1}\lambda_i(\lambda_i)^{2t-1}c_i^2) \qquad = -\frac{4t(2t-1)}{n^2}\sum_{n}^{i=1}\lambda_i(\lambda_i)^{2t-2}c_i^2(\frac{\partial\lambda_i}{\partial\eta}) \\
&= \frac{8t(2t-1)}{n^3}\sum_{n}^{i=1}\lambda_i^2(\lambda_i)^{2t-2}c_i^2 \\
&> 0
\end{aligned}$$

Thus, $\boldsymbol{\mathcal{L}}_t$ is convex with respect to the learning rate η.

3. Convolutional Neural Networks

3.1. Convolutional Filters

I would like to use translate and scale method to calculate the convolution. Thus, we can get

$$
\boldsymbol{I} * \boldsymbol{J} = 0 \times \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 1 & 1 & 1 & 1 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & & \\ & & & & & & \\ & & & & & & \end{pmatrix} + (-1) \times \begin{pmatrix} & 0 & 1 & 1 & 0 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 1 & 1 & 1 & 1 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 0 & 0 & 1 & 0 & 0 & \\ & & & & & & \\ & & & & & & \end{pmatrix} + 0 \times \begin{pmatrix} & & 0 & 1 & 1 & 0 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 1 & 1 & 1 & 1 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 0 & 0 & 1 & 0 & 0 \\ & & & & & & \\ & & & & & & \end{pmatrix}
$$

$$
+ (-1) \times \begin{pmatrix} & & & & & & \\ 0 & 1 & 1 & 0 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 1 & 1 & 1 & 1 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & & \\ & & & & & & \end{pmatrix} + 4 \times \begin{pmatrix} & & & & & & \\ & 0 & 1 & 1 & 0 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 1 & 1 & 1 & 1 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 0 & 0 & 1 & 0 & 0 & \\ & & & & & & \end{pmatrix} + (-1) \times \begin{pmatrix} & & & & & & \\ & & 0 & 1 & 1 & 0 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 1 & 1 & 1 & 1 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 0 & 0 & 1 & 0 & 0 \\ & & & & & & \end{pmatrix}
$$

$$
+ 0 \times \begin{pmatrix} & & & & & & \\ & & & & & & \\ 0 & 1 & 1 & 0 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 1 & 1 & 1 & 1 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & & \end{pmatrix} + (-1) \times \begin{pmatrix} & & & & & & \\ & & & & & & \\ & 0 & 1 & 1 & 0 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 1 & 1 & 1 & 1 & 0 & \\ & 0 & 1 & 1 & 1 & 0 & \\ & 0 & 0 & 1 & 0 & 0 & \end{pmatrix} + 0 \times \begin{pmatrix} & & & & & & \\ & & & & & & \\ & & 0 & 1 & 1 & 0 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 1 & 1 & 1 & 1 & 0 \\ & & 0 & 1 & 1 & 1 & 0 \\ & & 0 & 0 & 1 & 0 & 0 \end{pmatrix}
$$

$$
= \begin{pmatrix} & 0 & -1 & -1 & 0 & 0 & \\ & 0 & -1 & -1 & -1 & 0 & \\ & -1 & -1 & -1 & -1 & 0 & \\ & 0 & -1 & -1 & -1 & 0 & \\ & 0 & 0 & -1 & 0 & 0 & \\ & & & & & & \\ & & & & & & \end{pmatrix} + \begin{pmatrix} & & & & & & \\ 0 & -1 & -1 & 0 & 0 & & \\ 0 & -1 & -1 & -1 & 0 & & \\ -1 & -1 & -1 & -1 & 0 & & \\ 0 & -1 & -1 & -1 & 0 & & \\ 0 & 0 & -1 & 0 & 0 & & \\ & & & & & & \end{pmatrix} + \begin{pmatrix} & & & & & & \\ & 0 & 4 & 4 & 0 & 0 & \\ & 0 & 4 & 4 & 4 & 0 & \\ & 4 & 4 & 4 & 4 & 0 & \\ & 0 & 4 & 4 & 4 & 0 & \\ & 0 & 0 & 4 & 0 & 0 & \\ & & & & & & \end{pmatrix}
$$

$$
+ \begin{pmatrix} & & & & & & \\ & & 0 & -1 & -1 & 0 & 0 \\ & & 0 & -1 & -1 & -1 & 0 \\ & & -1 & -1 & -1 & -1 & 0 \\ & & 0 & -1 & -1 & -1 & 0 \\ & & 0 & 0 & -1 & 0 & 0 \\ & & & & & & \end{pmatrix} + \begin{pmatrix} & & & & & & \\ & & & & & & \\ & 0 & -1 & -1 & 0 & 0 & \\ & 0 & -1 & -1 & -1 & 0 & \\ & -1 & -1 & -1 & -1 & 0 & \\ & 0 & -1 & -1 & -1 & 0 & \\ & 0 & 0 & -1 & 0 & 0 & \end{pmatrix}
$$

$$
= \begin{pmatrix} 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 2 & -2 & 0 & 0 \\ 0 & -2 & 1 & 0 & 2 & -1 & 0 \\ 0 & 3 & 0 & 0 & 1 & -1 & 0 \\ 0 & -2 & 2 & 0 & 2 & -1 & 0 \\ 0 & 0 & -2 & 3 & -2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}
$$

Since we use zero padding around the input, we need to get rid of the numbers in the outermost layer. Thus, the actual output of the resulting matrix should be an 5×5 matrix, which is

$$
\boldsymbol{I} * \boldsymbol{J} = \begin{pmatrix} -1 & 2 & 2 & -2 & 0 \\ -2 & 1 & 0 & 2 & -1 \\ 3 & 0 & 0 & 1 & -1 \\ -2 & 2 & 0 & 2 & -1 \\ 0 & -2 & 3 & -2 & 0 \end{pmatrix}
$$

According to the lecture 4 slide, we know that this convolution kernel does the edge detection.

3.2. Size of Conv Nets
To make the calculation easier, let's write it in a table.

Layer	Input	Output	# params	# neurons	# connections
Image	-	$112 * 112 * 3$	-	-	-
Conv3-64	$112 * 112 * 3$	$112 * 112 * 64$	$3^2 * 64 * 3 + 64$	$112 * 112 * 64$	$112 * 112 * 3^2 * 64 * 3$
Max Pool	$112 * 112 * 64$	$56 * 56 * 64$	0	$56 * 56 * 64$	$56 * 56 * 2^2 * 64$
Conv3-128	$56 * 56 * 64$	$56 * 56 * 128$	$3^2 * 128 * 64 + 128$	$56 * 56 * 128$	$56 * 56 * 3^2 * 64 * 128$
Max Pool	$56 * 56 * 128$	$28 * 28 * 128$	0	$28 * 28 * 128$	$28 * 28 * 2^2 * 128$
Conv3-256	$28 * 28 * 128$	$28 * 28 * 256$	$3^2 * 128 * 256 + 256$	$28 * 28 * 256$	$28 * 28 * 3^2 * 128 * 256$
Conv3-256	$28 * 28 * 256$	$28 * 28 * 256$	$3^2 * 256 * 256 + 256$	$28 * 28 * 256$	$28 * 28 * 3^2 * 256 * 256$
Max Pool	$28 * 28 * 256$	$14 * 14 * 256$	0	$14 * 14 * 256$	$14 * 14 * 2^2 * 256$
FC-1024	$14 * 14 * 256$	1024	$14 * 14 * 256 * 1024 + 1024$	1024	$14 * 14 * 256 * 1024$
FC-100	1024	100	$100 * 1024 + 100$	100	$100 * 1024$
Softmax	100	100	-	-	-
Total	-	-	$52,444,644$	$1,957,988$	-

3.3. Receptive field
According to the definition of the receptive field, we can get that for a particular convolution layer, the size of receptive field is affected by the size of the kernel, the stride size, and the depth of the layer.

If the kernel size is large, the neuron could see more inputs from the last layer. Thus, it directly affects the size of the receptive field.

The stride size affects the distance of the different feature map, which can also affect the size of the receptive field. For example, if the stride size is large, neurons could see wider inputs from the last layer.

The depth of the layer also depends. If the layer is deeper in a Conv Net, it will composite more inputs from the previous layers. Thus, the receptive field size for the layer in the back must be equal or larger than that for the preceding layer.