1.1.1

We first consider a simple case with only one data, i.e., let $x_1 = [2,1]$, $w_0 = [0,0]$ and $t = 3$. By the formula of mini-batch SGD, we have

$$w_1 = w_0 - \eta \nabla_{w_0} L(x_1, w_0)$$
$$= w_0 - 2\eta x_1 (w_0^T x_1 - t_0)$$
$$= (2\eta t_0) x_1 \quad (\text{since } w_0 = [0,0])$$
$$= \lambda x_1 \quad \text{for some } \lambda \in \mathbb{R}$$

$$w_2 = w_1 - \eta \nabla_{w_1} L(x_1, w_1)$$
$$= w_1 - 2\eta x_1 (w_1^T x_1 - t_1) = w_1 - 2\eta x_1 (\lambda x_1^T x_1 - t_1)$$
$$= \lambda x_1 - \underbrace{(2\eta \lambda \|x_1\|_2^2 - 2\eta t_1)}_{\text{constant}} x_1 = \lambda' x_1 \quad \text{for some } \lambda' \in \mathbb{R}$$

Similarly, we have $w_3 = \lambda'' x_1$ for some $\lambda'' \in \mathbb{R}$. As we can see $\hat{w}$ is in the span of X. It can be generalized since for any x_j, $\nabla_{w_t} L(x_j, w_t)$ $= \nabla_{w_t} (w_t^T x_j - t_j)^2 = 2x_j \overbrace{(w_t^T x_j - t_j)}^{\text{constant}} = \lambda x_j$ for some $\lambda \in \mathbb{R}$. Since $w_0 = 0$, we have w_t lies in span(X). Therefore, we have shown that $\hat{w}$ can be written as $\mathcal{B}^T a$ for $a \in \mathbb{R}^b$ in the stationary cond. for mini-batch SGD. Also, as $\mathcal{B} \in \mathbb{R}^{b \times d}$ and $X \in \mathbb{R}^{n \times d}$ with full rank, and $b << n < d$, we have $\mathcal{B}$ in the row span of X. Therefore, we can write $\mathcal{B} = bX$ for $b \in \mathbb{R}^{b \times n}$, and so $\hat{w} = \mathcal{B}^T a = X^T b^T a$

$$= X^T (b^T a)$$
$$= X^T c$$

for some $c \in \mathbb{R}^n$. As a result, since w^* & $\hat{w}$ can be written in the same form as $X^T c$ for some $c \in \mathbb{R}^n$, we must have $w^* = \hat{w} = X^T (XX^T)^{-1} t$. To show $\hat{w}$ is indeed the minimum norm, assume there exists some other solution w, but we have: $(\hat{w}-w)^T\hat{w} = (\hat{w}-w)^T X^T c$

$$= [X(\hat{w}-w)]^T c$$

$$= [t-t]^T c = 0$$

which means that $\|w\|_2^2 = \|w-\hat{w}+\hat{w}\|_2^2 = \|w-\hat{w}\|_2^2 + \|\hat{w}\|_2^2 \geq \|\hat{w}\|_2^2$, and so we show $\hat{w}$ is indeed the minium norm.

1.2.1

Start with $x_1 = [2,1]$, $w_0 = [0,0]$ and $t = [2]$, we have:

$$\nabla_{w_0} L(x_1, w_0) = 2(w_0^T x_1 - t)x_1$$

$$= 2(-2)[2,1] = [-8,-4]$$

Therefore, $\nabla_{w_{0,0}} L(w_{0,0}) = -8$ & $\nabla_{w_{1,0}} L(w_{1,0}) = -4$, and

$$v_{0,0} = \beta v_{0,-1} + (1-\beta)(-8)^2$$

$$= 64(1-\beta)$$

Simarliy,

$$v_{1,0} = (1-\beta)(-4)^2 = 16(1-\beta)$$

As a result,

$$w_{0,1} = w_{0,0} - \frac{\eta}{\sqrt{v_{0,0}}+\varepsilon}(-8)$$

$$= \frac{8\eta}{8\sqrt{1-\beta}+\varepsilon}$$

$$w_{1,1} = w_{1,0} - \frac{\eta}{\sqrt{v_{0,1}}+\varepsilon}(-4)$$

$$= \frac{4\eta}{4\sqrt{1-\beta}+\varepsilon}$$

So we have $w_1 = [\frac{8\eta}{8\sqrt{1-\beta}+\varepsilon}, \frac{4\eta}{4\sqrt{1-\beta}+\varepsilon}]$. Now, we compute $w_{0,2}$:

$$\because \nabla_{w_1} L(x_1, w_1) = 2(w_1^T x_1 - t)x_1$$

$$= 2\left[\frac{16\eta}{8\sqrt{1-\beta}+\varepsilon} + \frac{4\eta}{4\sqrt{1-\beta}+\varepsilon} - 2\right][2,1]$$

$$= \left[\frac{16\eta}{8\sqrt{1-\beta}+\varepsilon} + \frac{4\eta}{4\sqrt{1-\beta}+\varepsilon} - 2\right][4,2]$$

We denote $\nabla_{w_{0,1}} L(w_{0,1}) = a_1$ & $\nabla_{w_{1,1}} L(w_{1,1}) = a_2$, we have

$$v_{0,1} = \beta v_{0,0} + (1-\beta)a_1^2 = (1-\beta)(64\beta + a_1^2)$$

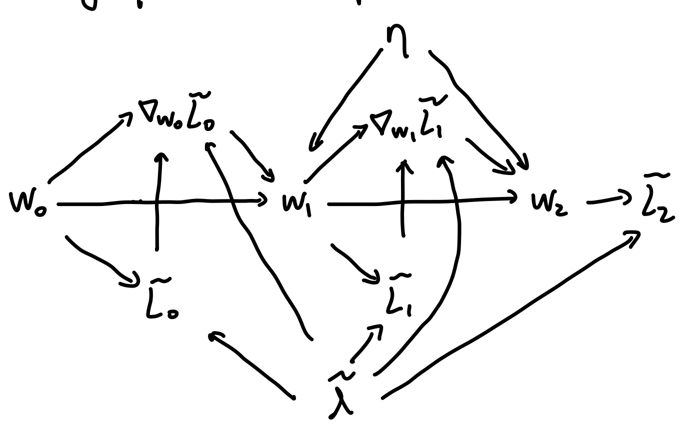
$$w_{0,2} = w_{0,1} - \frac{\eta}{\sqrt{v_{0,1}}+\varepsilon} a_1$$

$$= \frac{8\eta}{8\sqrt{1-\beta}+\varepsilon} - \frac{\eta a_1}{\sqrt{(64\beta+a_1^2)(1-\beta)}+\varepsilon}$$

However, this component is not a linear function of the first component of x_1, meaning that w stops to be in the span of X since span is the linear combination of rows in X, here it's x_1. Therefore, it violates the condition required in 1.1.1, leading to the consequence that w is not always the minimum norm. This still conv. to some other pt though.

2.1.1

The computation graph looks like follows:



2.1.2

Memory complexity for the forward-propagation is $O(1)$ since we only need to save the w_i at a time for $i=1,\dots,t$. For the backward propagation to compute $\nabla_\eta \tilde{L}_t$ is $O(t)$ since from the computation graph we know η is involved in every w_i for $i=1,\dots,t$, and so storing all w_i requires $O(t)$

2.2.1

$$\because w_1 = w_0 - \eta \nabla_{w_0} L_0 = w_0 - \frac{2\eta}{n} X^T(Xw_0 - t)$$

$$= w_0 - \frac{2\eta}{n} X^T a \quad \text{for } a = Xw_0 - t$$

$$\therefore L_1 = \frac{1}{n}\|Xw_1 - t\|_2^2$$

$$= \frac{1}{n}\|X(w_0 - \frac{2\eta}{n}X^T a) - t\|_2^2 = \frac{1}{n}\| -\frac{2\eta}{n}XX^T a + Xw_0 - t\|_2^2$$

$$= \frac{1}{n}\| -\frac{2\eta}{n}XX^T a + a\|_2^2$$

$$= \frac{1}{n}\|(I - \frac{2\eta}{n}XX^T)a\|_2^2 \quad (I \text{ is an } n\times n \text{ identity matrix})$$

$$= \frac{1}{n}[a^T(I - \frac{2\eta}{n}XX^T)^T(I - \frac{2\eta}{n}XX^T)a]$$

$$= \frac{1}{n} a^T(I - \frac{2\eta}{n}XX^T)^2 a$$

2.2.3

$$\nabla_\eta L_1 = \frac{2}{n}a^T(I - \frac{2\eta}{n}XX^T)(-\frac{2}{n}XX^T)a$$ (by chain rule, and since L_1 & η are both numbers, the derivative should be number, too)

Set $\nabla_\eta L_1 = 0$, we have $a^T(-\frac{2}{n}XX^T + \frac{4\eta}{n^2}(XX^T)^2)a = 0$

$$\Rightarrow \frac{2}{n}\eta = \frac{a^TXX^Ta}{a^T(XX^T)^2a}$$

$$\Rightarrow \eta = \frac{n}{2}\frac{a^TXX^Ta}{a^T(XX^T)^2a}$$

But $a^T X X^T a = \|X^T a\|_2^2$, and $a^T (XX^T)^2 a = a^T X X^T X X^T a$

$$= \|XX^T a\|_2^2$$

So after simplification, $\eta = \frac{n}{2} \frac{\|X^T a\|_2^2}{\|XX^T a\|_2^2}$

2.3.1

For $\tilde{L}$: $w_1 = w_0 - \eta \left[\frac{2}{n} X^T (Xw_0 - t) + 2\tilde{\lambda} w_0 \right]$

For L: $w_1 = (1-\lambda) w_0 - \frac{2\eta}{n} X^T (Xw_0 - t)$

2.3.2

Since for $\tilde{L}$, we have: $w_1 = w_0 - \frac{2\eta}{n} X^T (Xw_0 - t) - 2\tilde{\lambda}\eta w_0$, and for L, we have $w_1 = w_0 - \frac{2\eta}{n} X^T (Xw_0 - t) - \lambda w_0$, we compare two equations and we can have

$$2\tilde{\lambda}\eta w_0 = \lambda w_0$$

$$\Rightarrow \tilde{\lambda} = \frac{\lambda}{2\eta}$$

3.1

After fliping the filter, we have $\text{flip}(J) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$, and after computation, we have

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

This filter detects the horizontal edge.

3.2

total # of neurons:

① CNN: Image: 32x32
Conv 1: 32x32
pool 1: 16x16
Conv 2: 16x16
Pool 2: 8x8
Conv 3: 8x8
total: $\boxed{2688}$

② FCNN: Image: 1024
FC1: 1024
pool 1: 256
FC 2: 256
Pool 2: 64
FC 3: 64
total: $\boxed{2688}$

Total # of weights:

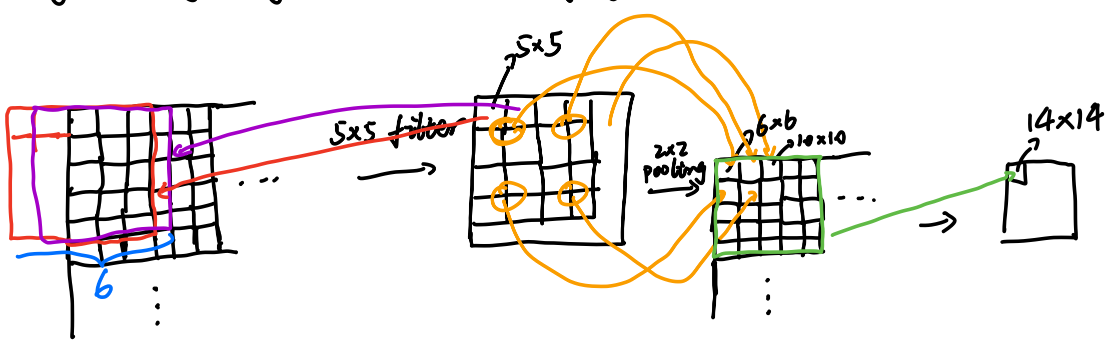
① CNN: Image to Conv1: 3x3
pool1 to conv2: 3x3
pool 2 to conv 2: 3x3
total: $\boxed{27}$

② FCNN: Im to FC1: 1024x1024
pool1 to FC2: 256x256
pool2 to FC: 64x64
total: $\boxed{1118208}$

Disadvantage of having more trainable parameters: too high memory cost

3.3.

A rough analysis guide is following graph:



The receptive field of a neuron after the second conv. layer is 14x14.

Two other things that can affect the size of the receptive field:

① max-pooling layer size ② the stride when applying the convolutional layer