# Assignment 1 non-programming questions

Hongyu Chen

October 15, 2020

## 1 Question 2

### 1.1 Question 2(c)

In matrix poly function we have two triply-nested loops. There are one addition and one multiplication for each iteration in the most-inner loop. Since we are using square matrix which size are n*n, we will has total of $2N^3$ of these iterations. So we will has :
timing(100): $2 * 100^3 = 2000000$ additions and multiplications
timing(300): $2 * 300^3 = 54000000$ additions and multiplications
timing(1000): $2 * 1000^3 = 2000000000$ additions and multiplications

## 2 Question 4

### 2.1 Question 4(e)

First we need to prove that

$[\frac{\partial J}{\partial w}]_j = [X^T(y-t)/N]_j$
It is equivalent to :

$\frac{\partial J}{\partial w_j} = [X^T(y-t)/N]_j$
Which means we need to show that:

$\sum_{i=1}^{N}(y^i - t^i)(x_j^i)/N = [X^T(y-t)/N]_j$
We know that:
$\sum_{i=1}^{N}(y^i - t^i)(x_j^i)/N = \frac{(y^0-t^0)(x_j^0)+(y^1-t^1)(x_j^1)+(y^2-t^2)(x_j^2)....+(y^N-t^N)(x_j^N)}{N}$

And we know in the right hand side,

$$y = \begin{bmatrix} y^0 \\ y^1 \\ y^2 \\ .. \\ .. \\ .. \\ y^N \end{bmatrix}, t = \begin{bmatrix} t^0 \\ t^1 \\ t^2 \\ .. \\ .. \\ .. \\ t^N \end{bmatrix} \; X = \begin{bmatrix} ((X^0)^T \\ (X^1)^T \\ (X^2)^T \\ .. \\ .. \\ .. \end{bmatrix} = \begin{bmatrix} ((x_0^0) & (x_1^0) & ... \\ (x_0^1) & (x_1^1) & ... \\ .. & & \\ .. & & \\ .. & & \end{bmatrix}$$

We know that $[X^T]_{ji} = X_{ij}$ which is in the j th column in X ($x_j^i$).

So we can say that:

$$[X^T(y-t)/N]_j = \frac{\begin{bmatrix}((x_j^0) & (x_j^1) & ...\end{bmatrix} * ( \begin{bmatrix} y^0 \\ y^1 \\ y^2 \\ .. \\ .. \\ .. \\ y^N \end{bmatrix} - \begin{bmatrix} t^0 \\ t^1 \\ t^2 \\ .. \\ .. \\ .. \\ t^N \end{bmatrix} )}{N}$$

$$= [X^T(y-t)/N]_j = \frac{\begin{bmatrix}((x_j^0) & (x_j^1) & ...\end{bmatrix} * ( \begin{bmatrix} y^0 - t^0 \\ y^1 - t^1 \\ y^2 - t^2 \\ .. \\ .. \\ .. \\ y^N - t^N \end{bmatrix} )}{N}$$

$$= \frac{(y^0-t^0)(x_j^0)+(y^1-t^1)(x_j^1)+(y^2-t^2)(x_j^2)....+(y^N-t^N)(x_j^N)}{N}$$

$$= \sum_{i=1}^{N}(y^i - t^i)(x_j^i)/N = [\frac{\partial_J}{\partial w}]_j$$

## 2.2 Question 4(f)

To prove the logistic cross entropy, first we know the definition of logistic cross entropy:

$L_{LCE}(z,t) = L_{CE}(\sigma(z),t)$

We know that:

$\sigma(z) = \frac{1}{1+e^{(-z)}}$

So we can plug this into the equation we will get:

$L_{LCE}(z,t) = L_{CE}(\sigma(z),t)$

$= -tlog(\sigma(z)) - (1-t)log(1-\sigma(z))$

$= -tlog(\frac{1}{1+e^{(-z)}}) - (1-t)log(1-\frac{1}{1+e^{(-z)}})$

By properties of log we have:

$= -t(log(1) - log(1+e^{(-z)}) - (1-t)log(\frac{1+e^{(-z)}-1}{1+e^{(-z)}})$

we know that $log(1) = 0$ so we have:

$= -tlog(1+e^{(-z)}) - (1-t)log(\frac{e^{(-z)}}{1+e^{(-z)}})$

then for $log(\frac{e^{(-z)}}{1+e^{(-z)}})$ we can multiply $e^{(-z)}$ to both denominator and numerator, we will get:

$= -tlog(1+e^{(-z)}) - (1-t)log(\frac{1}{1+e^{(z)}})$

Again by properties of log:

$= -tlog(1+e^{(-z)}) - (1-t)(log(1) - log(1+e^{(z)}))$

$= -tlog(1+e^{(-z)}) - (1-t)(-log(1+e^{(z)}))$

$= -tlog(1+e^{(-z)}) + (1-t)(log(1+e^{(z)}))$

Which is what we need.

# 3 Question 6

## 3.1 Question 6(e)

The reason of validation accuracy in part (d) is considerably higher than in part (c) is because if we take a look at the number 5 and 6, we will notice that 5 and 6 looks very similar. If we connect 5 at left bottom corner, it will look similar as 6. But for part (d) we have 4 and 7, these two are not similar at all. So it will has a higher accuracy since it will be easier to do the classification between 4 and 7 than 5 and 6. For the second observation, validation accuracy normally will be less than training accuracy since training data is something which model is very familiar but for validation data, it is new data that is completely new to the model. But since we have discuss above that 4 and 7 are not similar to each other. I would say the reason that happens is because our validation data do

not have enough complexity. It does not cover some edge cases but these edge cases appears in the training data set. So the training accuracy will be lower than the test accuracy.

### 3.2 Question 6(f)

K in the KNN (k-Nearest Neighbors) represent you need to compare K closest neighbors. So if we choose even k instead of odd k we might end up with a tie in the decision (K/2 against K/2). The classification will be really tough and risk if we use a even number. So in order to avoid this kind of tie situation, we use odd number of K instead of even.

### 3.3 Question 6(g)

KNN produces such high accuracies on the MNIST data is because that MNIST itself is commonly used for training various image processing systems. Each MNIST image has been heavily pre-processed they are designed for image processing like KNN. So it actually much easier then the real world hand writing problems. And KNN will of course get a high accuracy since again MNIST are design for it. For binary classification problems, KNN has a good accuracy is because KNN do not have training step so new training data can be added without affect the accuracy. And KNN is also a memory based algorithm, its classifier will respond very quickly as we accept new training data during real time. It respond very fast and adapts very fast.