CSC311
Assignment 01 (Non-programming)
Ye Huang

Q4(f).

Let's assume that $\frac{\partial \mathcal{J}}{\partial \omega_j} = \sum_{i=1}^{N}(y^{(i)} - t^{(i)})x_j^{(i)}/N$

Consider,

$[X^T(y-t)/N]_j = \frac{1}{N}[X^T(y-t)]_j$ (Since $\frac{1}{N}$ is scalar)

Let $w = y - t$ where w is a vector and $w_i = y^{(i)} - t^{(i)}$
$= \frac{1}{N}[X^T w]_j$

Since y and t are both mx1 vectors, w is mx1, then X^T is nxm(since X is mxn).

Therefore $[X^T w]_j = [X^T]_j w$, since at the j^{th} row of $X^T w$ is just the j^{th} row of X^T multiplying(matrix wise) w, which should give us a scalar.

Then,
$\frac{1}{N}[X^T w]_j = \frac{1}{N}[X^T]_j w$

$= \frac{1}{N}(\sum_{i=1}^{M}[X^T]_{ji} w_i)$ (definition of matrix multiplication)

$= \frac{1}{N}(\sum_{i=1}^{M} X_{ij} w_i)$ (definition of matrix transpose)

$= \frac{1}{N}(\sum_{i=1}^{M} X_j^{(i)} w_i)$ ($X_i j = x_j^i$ given)

$= \frac{1}{N}(\sum_{i=1}^{M} X_j^{(i)}(y^{(i)} - w^{(i)}))$ (definition of w)

$= \frac{\partial \mathcal{J}}{\partial w_j}$

And notice, we have used only equivalence signs(=) for our proof, and all the definitions have the if-and-only-if property (the meaning of definition).

So, assuming without-loss-of-generality, we can also state that given the consequence, we can use this exact logic to show the assumption holds.

Therefore, we have proven the claim as required.

Q4(e).

Consider,

$\mathcal{L}_{LCE}(z,t) = \mathcal{L}_{CE}(\sigma(z), t)$ (definition of logistic-cross-entropy)

$= -t\,log\,\sigma(z) - (1-t)log(1-\sigma(z))$ (definition of $\mathcal{L}_{CE}$)

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

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

$= -t[-log(1+e^{-z})] - (1-t)[log(e^{-z}) - log(1+e^{-z})]$ (log1 = 0)

$= t\,log(1+e^{-z}) - (1-t)[-z - log(1+e^{-z})]$ ($log_b b^a = a$)

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

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

$= t\,log(1+e^{-z}) - (1-t)[z - log(1+e^z) - log(e^z)]$ (log rule)

$= t\,log(1 + e^{-z}) - (1 - t)[z - log(1 + e^{z}) - z]$ ($log_b b^a = a$)

$= t\,log(1 + e^{-z}) - (1 - t)[-log(1 + e^{z})]$

$= t\,log(1 + e^{-z}) + (1 - t)[log(1 + e^{z})]$

Q6(e).

Consider the image of "5" and "6", they are more similar in hand-writing in comparison to "4" and "7" (where the only difference between "5" and "6" is whether there would be a gap after the final left curl).

The reason why the validation accuracy of part (d) is better than part (c) is because we are using a larger value of K, so our algorithm becomes more resilient to noise and outliers.

For the best value of K, by having a larger K to try to distinguish "5" and "6" will only confuses the algorithm, as the hand-writings are so similar, yet the values are somewhat arbitrary and misleading.

Q6(f).

Given that we are solving a binary-classification problem, we only use odd value for K because if we use an even number for K, and half the neighbours are class 0 and the other half are class 1(binary data), then we won't be able to determine the class of the current data point.

Therefore, an odd number for K is necessary.

Q6(g).

The reason why KNN performs well on MNIST data, is because MNIST data is highly-preprocessed (grey scaling,thresholding...), and the data set is fairly simple (with the target being an integer 0 to 9).

When reducing the target down to only 2 (binary classification), we have also eliminated all the noise data in the data set (ex. in part (d), hand-writing of "4" is similar to "9", but we made a reduction data so only "4" and "7" exists) which helps greatly with the increasing the accuracy.

End of Assignment