ECSE 552 Deep Learning
Assignment 1

Question 1

Figure 1 shows the training & validation mean squared errors as a function of iterations. The full derivations for the MLP backpropagation formulas can be found in the assignment's Jupyter notebook.

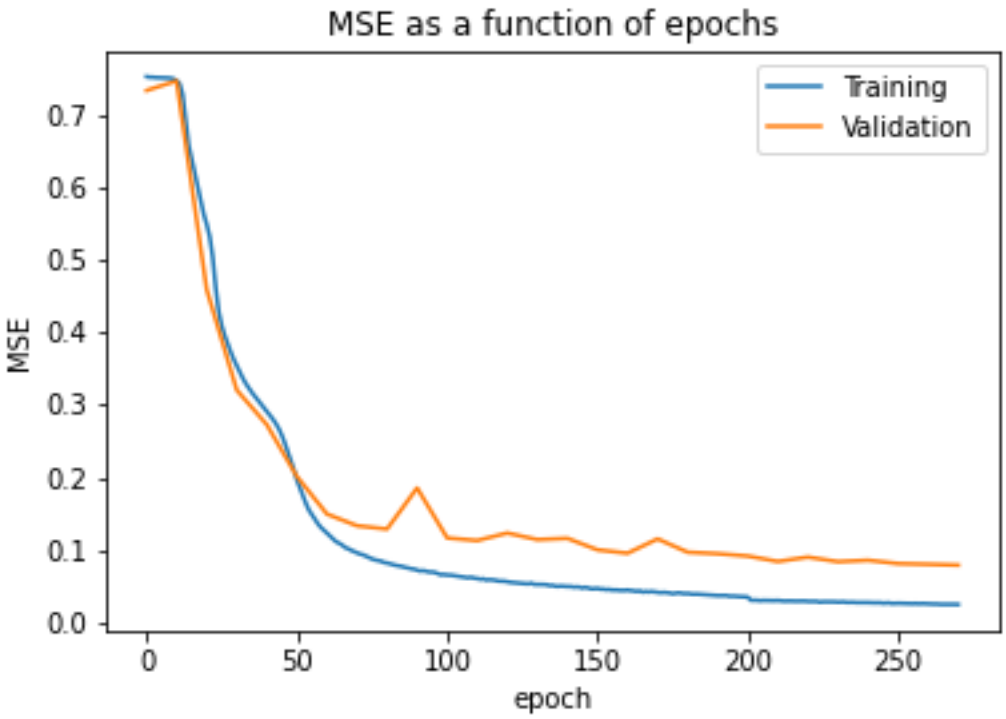

Figure 1: Error as a function of epochs

Question 2

Figure 2 again displays the accuracies as a function of epoch for both datasets.

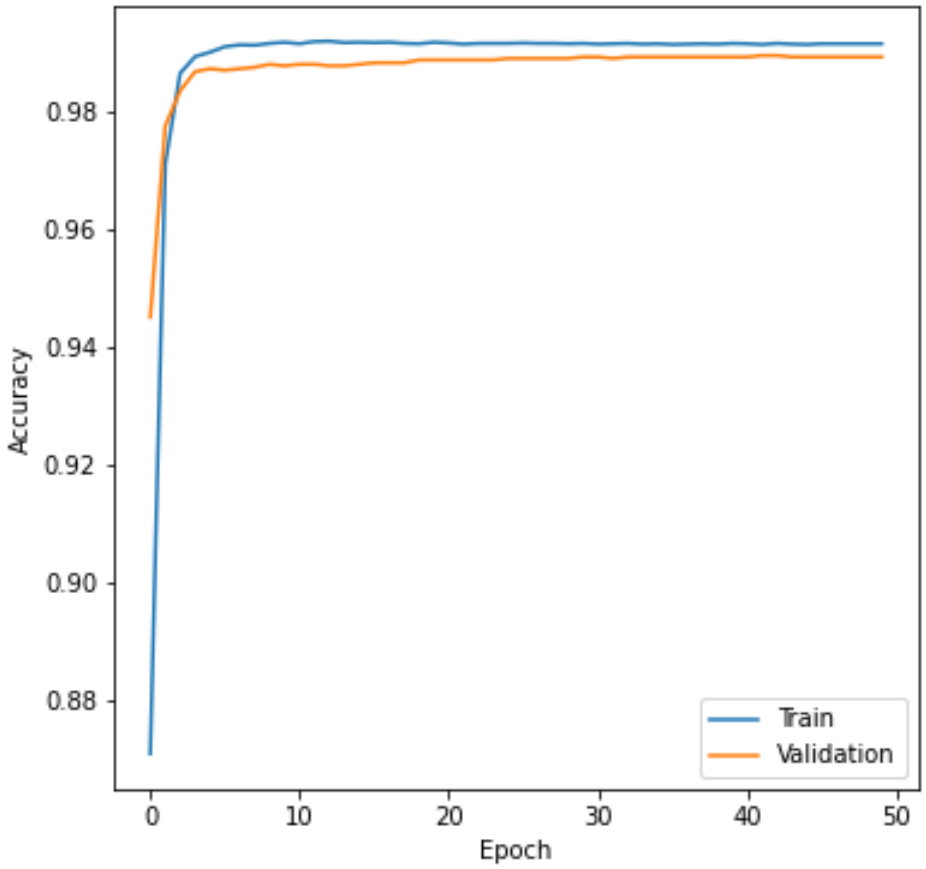

Figure 2: Accuracy as a function of epochs

Figure 3 plots the validation set and the neuron activation threshold for each of the 30 hidden units.

All derivations and explanations are included as inline text in the Jupyter notebook.

Question 3

Here, we are going to evaluate the conditions under which the sigmoid function $\sigma()$ saturates when used as activation function (AF) of the output unit and a negative log-likelihood cost function is used.

Let the output of a NN be defined as $\sigma(z)$, where $z = \mathbf{w^T h} + b$ with z and b being scalars and $\mathbf{w}$ and $\mathbf{h}$ being vectors. The likelihood of this model is:

$$P(y|z) = \sigma((2y-1)z)$$

We can verify that $P(y=1|z) = \sigma(z)$ and $P(y=0|z) = \sigma(-z) = 1 - \sigma(z)$ by plugging in the values for y.

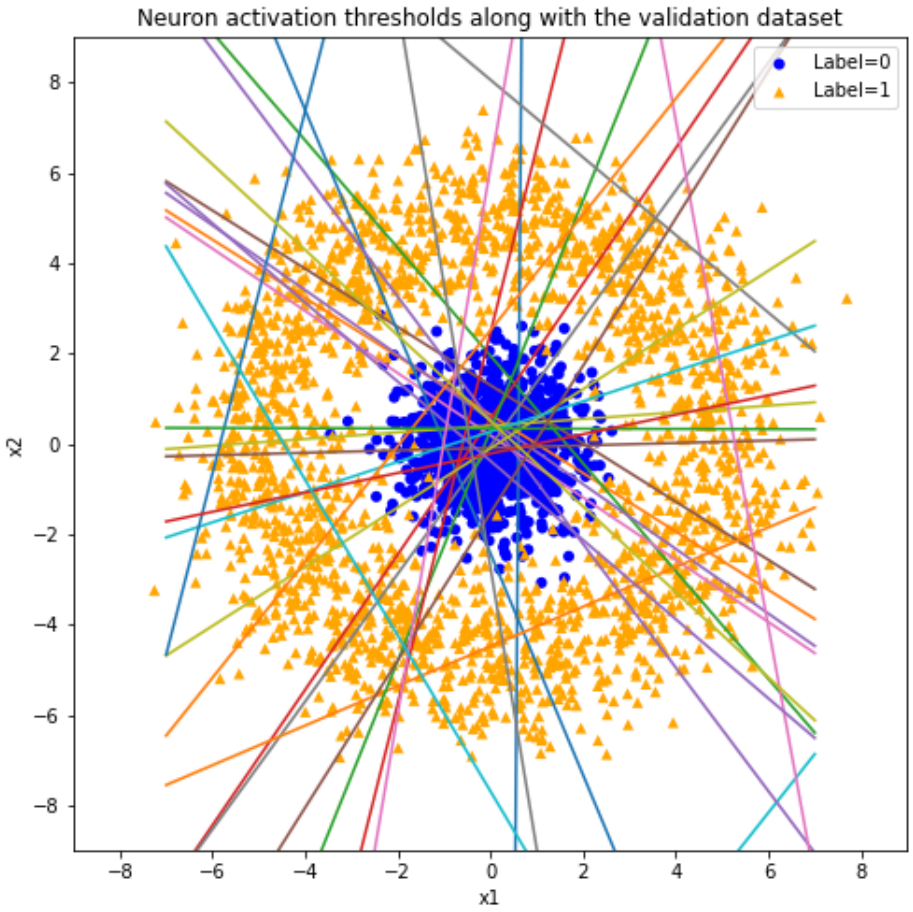

Figure 3: Validation set & Hidden layer neuron activation thresholds

Given this likelihood, we can define the cost function as negative log-likelihood:

$$J(z) = -\log(\sigma((2y-1)z))$$

We will now calculate the derivative of J with respect to z and evaluate its behavior in the following four conditions:

Step-1 The cost function $J(z) = -\log(\sigma((2y-1)z))$ is a negative log-likelihood, where $\sigma(x)$ is the sigmoid function, it is given by the formula:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

Step-2 We need to find the derivative of J with respect to z, we'll start by finding the derivative of the sigmoid function:

$\frac{d}{dx}\sigma(x) = \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right]$

$\frac{d}{dx}\sigma(x) = \frac{d}{dx}\left(1+e^{-x}\right)^{-1}$

$\frac{d}{dx}\sigma(x) = -\left(1+e^{-x}\right)^{-2}\left(-e^{-x}\right)$

$\frac{d}{dx}\sigma(x) = \frac{e^{-x}}{(1+e^{-x})^2}$
$\frac{d}{dx}\sigma(x) = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}$
$\frac{d}{dx}\sigma(x) = \frac{1}{1+e^{-x}} \cdot \frac{\left(1+e^{-x}\right)-1}{1+e^{-x}}$
$\frac{d}{dx}\sigma(x) = \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right)$
$\frac{d}{dx}\sigma(x) = \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right)$
$\frac{d}{dx}\sigma(x) = \sigma(x) \cdot (1 - \sigma(x))$

$$\frac{d\sigma(x)}{dx} = \sigma(x) * (1 - \sigma(x)) \tag{2}$$

Step-3 Here, we take the derivative of J with respect to z:

$$\frac{\partial J}{\partial z} = -\frac{\partial}{\partial z}(\log(\sigma((2y-1)z))) \tag{3}$$

Step-4 Using the chain rule, we get:

$$\frac{\partial J}{\partial z} = -\frac{1}{\sigma((2y-1)z)} * \frac{\partial \sigma((2y-1)z)}{\partial((2y-1)z)} * \frac{\partial((2y-1)z)}{\partial z} \tag{4}$$

Step-5 Substitute the derivative of the sigmoid function with respect to its input derived in (2):

$$\frac{\partial J}{\partial z} = -\frac{1}{\sigma((2y-1)z)} * (\sigma((2y-1)z) * (1 - \sigma((2y-1)z))) * \frac{\partial((2y-1)z)}{\partial z} \tag{5}$$

$$\frac{\partial J}{\partial z} = -(2y-1) * (1 - \sigma((2y-1)z)) \tag{6}$$

Step-6 The derivative of J with respect to z can be expressed as:

$$\frac{\partial J}{\partial z} = -(2y-1) * (1 - \sigma((2y-1)z)) \tag{7}$$

1. z **is large and positive and y = 1.** Substituting the value of y = 1 in the equation above, we get:

$$\lim_{z\to\infty} -(2*1-1) * (1 - \sigma((2*1-1)z)) = \lim_{z\to\infty} -(1 - \sigma(z)) = -(1-1) = 0$$

   Hence, the derivative $\frac{\partial J}{\partial z}$ approaches 0 and the gradient vanishes.

2. z **is large and positive and y = 0.** Substituting the value of y = 0 in the equation, we get:

$$\lim_{z\to\infty} -(2*0-1) * (1 - \sigma((2*0-1)z)) = \lim_{z\to\infty} (1 - \sigma(-z)) = (1-0) = 1$$

   Hence, the derivative $\frac{\partial J}{\partial z}$ approaches 1 and the gradient does not vanish.

3. z **is large (in absolute value) and negative and y = 1.** Substituting the value of y = 1 in the equation, we get:

$$\lim_{z\to-\infty} -(2*1-1) * (1 - \sigma((2*1-1)z)) = \lim_{z\to-\infty} -(1 - \sigma(z)) = -(1-0) = -1$$

   Hence, the derivative $\frac{\partial J}{\partial z}$ approaches -1 and gradient does not vanish.

4. z **is large (in absolute value) and negative and y = 0.** Substituting the value of y = 0 in the equation, we get:

$$\lim_{z\to-\infty} -(2*0-1) * (1 - \sigma((2*0-1)z)) = \lim_{z\to-\infty} (1 - \sigma(-z)) = (1-1) = 0$$

   Hence, the derivative $\frac{\partial J}{\partial z}$ approaches 0 and the gradient vanishes.

In conclusion, we notice that the partial derivative of the cost function with respect to its input z vanishes only when the prediction is correct (ie y = 1 and z tends to infinity or y = 0 and z tends to negative infinity). When the prediction is incorrect, the gradient tries to pull z in the direction that increases the cost function since it points in the direction of the steepest ascent, as we know from introductory calculus classes. Consequently, the negative of the gradient is pointing in the direction z should go in to reduce the error (Not quite gradient descent since we never update z but the same principle applies). For example, if z tends to negative infinity and y = 1, then the gradient of -1 is telling us z should be increased since $-\frac{\partial J}{\partial z}$ = -(-1) = 1. Similarly, if z tends to positive infinity and y = -1, z should be decreased as $-\frac{\partial J}{\partial z}$ = -(1) = -1. The derivative thus vanishes progressively as the predictions get better and better.

Another point to note is that all gradients are between -1 and 1, or between 0 and 1 in absolute value. Consequently, however bad the predictions are, the gradient in absolute value will never exceed 1. This means that, in the best case where the prediction is infinitely wrong, the activation function won't lead to the gradient vanishing when multiplying by $\frac{\partial J}{\partial z}$ in the chain rule, while in all other cases it will cause a problem for the input layers of the MLP.

All of the above leads to the conclusion that sigmoid is only a suitable activation function for the output layer, since it will only lead to the gradient vanishing if the prediction of the output layer (Which we only care about) becomes correct. Using sigmoid in intermediate layers means that we will be multiplying multiple chained gradients like the one above, which will lead to little to no change during the gradient descent's weight updates in the MLP's earlier layers.