CSE616 Neural Networks and Their Applications
Assignment 1

Shahira Hany Hussein Mohamed Amin (2101341)

April 2022

1 Problem 1

1.1 Part a

$$\begin{bmatrix} b_1^{[1]} & w_{11}^{[1]} & w_{21}^{[1]} \\ b_2^{[1]} & w_{12}^{[1]} & w_{22}^{[1]} \\ b_3^{[1]} & w_{13}^{[1]} & w_{23}^{[1]} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Given the weights and biases of the first layer and the input vector:

$$\begin{bmatrix} 1 & 2 & 2 \\ 0 & 1 & -1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 9 \\ -2 \\ 5 \end{bmatrix}$$

The output of the network:

$$\hat{y} = \begin{bmatrix} b_1^{[2]} & w_{11}^{[2]} & w_{21}^{[2]} & w_{31}^{[2]} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 9 \\ -2 \\ 5 \end{bmatrix} = 36$$

1.2 Part b

Let $h(.)$ be the ReLU activation function:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = h\left(\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \right) \begin{bmatrix} max(0,9) \\ max(0,-2) \\ max(0,5) \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 5 \end{bmatrix}$$

The output of the network:

$$\hat{y} = h\left(\begin{bmatrix} b_1^{[2]} & w_{11}^{[2]} & w_{21}^{[2]} & w_{31}^{[2]} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \right) = max\left(0, \begin{bmatrix} 1 & 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 9 \\ 0 \\ 5 \end{bmatrix} \right) = 38$$

1.3 Part c

Given a squared error loss function:

$$J = (\hat{y} - y)^2, \quad \frac{\partial J}{\partial \hat{y}} = 2(\hat{y} - y)$$

where

$$\hat{y} = \sum_{i=0}^{3} w_{i1}^{[2]} a_i + b_1^{[2]}, \quad \frac{\partial \hat{y}}{\partial b_1^{[2]}} = 1, \quad \frac{\partial \hat{y}}{\partial w_{21}^{[2]}} = a_2$$

Using the Chain rule:

$$\frac{\partial J}{\partial b_1^{[2]}} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_1^{[2]}} = 2(\hat{y} - y) \tag{$*$}$$

$$\frac{\partial J}{\partial w_{21}^{[2]}} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_{21}^{[2]}} = 2(\hat{y} - y)a_2 \tag{$*$}$$

Since

$$a_2 = \sum_{i=1}^{2} w_{i2}^{[1]} x_i + b_2^{[1]}, \quad \frac{\partial a_2}{\partial b_2^{[1]}} = 1$$

Then, using the Chain rule:

$$\frac{\partial J}{\partial b_2^{[1]}} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_2} \frac{\partial a_2}{\partial b_2^{[1]}} = 2(\hat{y} - y)w_{21}^{[2]} \tag{$*$}$$

Since

$$a_3 = \sum_{i=1}^{2} w_{i3}^{[1]} x_i + b_3^{[1]}, \quad \frac{\partial a_3}{\partial w_{13}^{[1]}} = x_1$$

Then, using the Chain rule:

$$\frac{\partial J}{\partial w_{13}^{[1]}} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial w_{13}^{[1]}} = 2(\hat{y} - y)w_{31}^{[2]} x_1 \tag{$*$}$$

1.4 Part d

Forward pass is calculated in Part (a) to obtain the activation and network output. To update $b_2^{[1]}$:

$$\begin{aligned} b_2^{[1]} &= b_2^{[1]} - \eta \frac{\partial J}{\partial b_2^{[1]}} \\ &= b_2^{[1]} - 2\eta(\hat{y} - y) w_{21}^{[2]} \\ &= 0 - (2)(2)(36 - 32) \\ &= -16 \end{aligned}$$

To update $w_{13}^{[1]}$:

$$\begin{aligned} w_{13}^{[1]} &= w_{13}^{[1]} - \eta \frac{\partial J}{\partial w_{13}^{[1]}} \\ &= w_{13}^{[1]} - 2\eta(\hat{y} - y) w_{31}^{[2]} x_1 \\ &= 3 - (2)(2)(36 - 32)(2) \\ &= -29 \end{aligned}$$

1.5 Part e

Yes, checking multiple models on the test set and selecting the best performing model will be a good indicator of the out-of-sample error.
Justification: The reason for this is that a model may overfit the training data performing very well on the training set, but fails to predict on test data (out-of-sample data) which leads to large out-of-sample error. However, by splitting the available data set into train set and validation set and treating the model as a hyper-parameter, multiple models are trained on the training set and evaluated on the validation set. The best performing model (i.e. the model with the least validation loss) is selected. This helps to solve the problem of overfitting, and therefore, the model will be a good indicator of the out-of-sample error.

2 Problem 2

$$\frac{\partial f}{\partial x_1} = \sum_{j=1}^{m} \frac{\partial f}{\partial g_j}\frac{\partial g_j}{\partial x_1} = \frac{\partial f}{\partial g_1}\frac{\partial g_1}{\partial x_1} + \frac{\partial f}{\partial g_2}\frac{\partial g_2}{\partial x_1} = \cos(g_1)(e^{x_2}) + 2g_2$$

$$\frac{\partial f}{\partial x_2} = \sum_{j=1}^{m} \frac{\partial f}{\partial g_j}\frac{\partial g_j}{\partial x_2} = \frac{\partial f}{\partial g_1}\frac{\partial g_1}{\partial x_2} + \frac{\partial f}{\partial g_2}\frac{\partial g_2}{\partial x_2} = \cos(g_1)(x_1 e^{x_2}) + (2g_2)(2x_2)$$

3 Problem 3

3.1 Part 1

Using the quotient rule:

$$\frac{\partial f}{\partial z} = \frac{(1+e^{-z})(0)+(1)(e^{-z})}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}}\Big(1-\frac{1}{1+e^{-z}}\Big) = f(z)(1-f(z))$$

3.2 Part 2

Since

$$z = \sum_{j=1}^{D} w_j x_j, \quad \frac{\partial z}{\partial w_i} = x_i$$

Using the Chain rule and from Part (1):

$$\frac{\partial f}{\partial w_i} = \frac{\partial f}{\partial z}\frac{\partial z}{\partial w_i} = f(z)(1 - f(z))x_i, \quad i = 1, 2, \ldots, D$$

$$\frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f}{\partial w_1} & \frac{\partial f}{\partial w_2} & \cdots & \frac{\partial f}{\partial w_D} \end{bmatrix}^T$$

3.3 Part 3

Given

$$J(w) = \frac{1}{2}\sum_{i=1}^{m} |w^T x^{(i)} - y^{(i)}|$$

Let

$$u^{(i)} = w^T x^{(i)} - y^{(i)}, \quad \frac{\partial u^{(i)}}{\partial w_j} = x_j^{(i)}$$

Then,

$$J(w) = \frac{1}{2}\sum_{i=1}^{m} |u^{(i)}| = \frac{1}{2}\sum_{i=1}^{m} \sqrt{(u^{(i)})^2}, \quad \frac{\partial J}{\partial u^{(i)}} = \frac{1}{2}\frac{1}{2}\frac{2u^{(i)}}{\sqrt{(u^{(i)})^2}} = \frac{1}{2}\frac{u^{(i)}}{|u^{(i)}|}$$

Using the Chain rule:

$$\begin{aligned}
\frac{\partial J}{\partial w_j} &= \sum_{i=1}^{m} \frac{\partial J}{\partial u^{(i)}} \frac{\partial u^{(i)}}{\partial w_j} \\
&= \sum_{i=1}^{m} \frac{1}{2}\frac{u^{(i)}}{|u^{(i)}|} x_j^{(i)} \\
&= \frac{1}{2}\sum_{i=1}^{m} \frac{w^T x^{(i)} - y^{(i)}}{|w^T x^{(i)} - y^{(i)}|} x_j^{(i)}, \quad j = 1, 2, \ldots, D
\end{aligned}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_1} & \frac{\partial J}{\partial w_2} & \cdots & \frac{\partial J}{\partial w_D} \end{bmatrix}^T$$

3.4 Part 4

Let

$$J(w) = J_1(w) + J_2(w) \quad \text{and} \quad u^{(i)} = w^T x^{(i)} - y^{(i)}, \quad \frac{\partial u^{(i)}}{\partial w_j} = x_j^{(i)}$$

where

$$J_1(w) = \frac{1}{2}\sum_{i=1}^{m}(u^{(i)})^2, \quad \frac{\partial J_1}{\partial u^{(i)}} = \frac{1}{2}(2)u^{(i)} = u^{(i)}$$

And

$$J_2(w) = \lambda \left\|w\right\|_2^2, \quad \frac{\partial J_2(w)}{\partial w_j} = 2\lambda w_j$$

Using the Chain rule:

$$\begin{aligned}
\frac{\partial J(w)}{\partial w_j} &= \frac{\partial J_1(w)}{\partial w_j} + \frac{\partial J_2(w)}{\partial w_j} \\
&= \sum_{i=1}^{m} \frac{\partial J_1}{\partial u^{(i)}} \frac{\partial u^{(i)}}{\partial w_j} + \frac{\partial J_2(w)}{\partial w_j} \\
&= \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)}) x_j^{(i)} + 2\lambda w_j, \quad j = 1, 2, \ldots, D
\end{aligned}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_1} & \frac{\partial J}{\partial w_2} & \cdots & \frac{\partial J}{\partial w_D} \end{bmatrix}^T$$

3.5 Part 5

Let

$$J(w) = \sum_{i=1}^{m} y^{(i)} log(f^{(i)}(w)) + (1 - y^{(i)}) log(1 - f^{(i)}(w)),$$

$$\frac{\partial J(w)}{\partial f^{(i)}(w)} = \sum_{i=1}^{m} \frac{y^{(i)}}{f^{(i)}(w)} - \frac{1 - y^{(i)}}{1 - f^{(i)}(w)} = \frac{y^{(i)} - f^{(i)}(w)}{f^{(i)}(w)(1 - f^{(i)}(w))}$$

where

$$f^{(i)}(w) = \frac{1}{1 + e^{-w^T x^{(i)}}}, \quad \frac{\partial f^{(i)}(w)}{\partial w_j} = f^{(i)}(w)(1 - f^{(i)}(w)) x_j^{(i)} \quad \text{(Part (2))}$$

Using the Chain rule:

$$\begin{aligned}
\frac{\partial J(w)}{w_j} &= \sum_{i=1}^{m} \frac{\partial J(w)}{\partial f^{(i)}(w)} \frac{\partial f^{(i)}(w)}{\partial w_j} \\
&= \sum_{i=1}^{m} \frac{y^{(i)} - f^{(i)}(w)}{f^{(i)}(w)(1 - f^{(i)}(w))} f^{(i)}(w)(1 - f^{(i)}(w)) x_j^{(i)} \\
&= (y^{(i)} - f^{(i)}(w)) x_j^{(i)} \\
&= \Big(y^{(i)} - \frac{1}{1 + e^{-w^T x^{(i)}}} \Big) x_j^{(i)}, \quad j = 1, 2, \ldots, D
\end{aligned}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_1} & \frac{\partial J}{\partial w_2} & \cdots & \frac{\partial J}{\partial w_D} \end{bmatrix}^T$$

3.6 Part 6

Let

$$f(w) = tanh(z) = \frac{sinh(z)}{cosh(z)}$$

Using the quotient rule:

$$\frac{\partial f(w)}{\partial z} = \frac{cosh^2(z) - sinh^2(z)}{cosh^2(z)} = 1 - \frac{sinh^2(z)}{cosh^2(z)} = 1 - tanh^2(z)$$

where

$$z = w^T x, \quad \frac{\partial z}{\partial w_j} = x_j$$

Using the Chain rule:

$$\begin{aligned}
\frac{\partial f(w)}{\partial w_j} &= \frac{\partial f(w)}{\partial z}\frac{\partial z}{\partial w_j} \\
&= (1 - tanh^2(z))x_j \\
&= (1 - tanh^2(w^T x))x_j, \quad j = 1, 2, \ldots, D
\end{aligned}$$

$$\frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f}{\partial w_1} & \frac{\partial f}{\partial w_2} & \cdots & \frac{\partial f}{\partial w_D} \end{bmatrix}^T$$

4 Problem 4

I used Jupyter Notebook for this exercise.

- Since this problem is a classification problem, where classes are categorized according to different qualities of red wine, I used one hot encoding to map the wine quality values in the dataset into integer values. Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1. The final output of the network is a binary vector representing the wine quality.
- The loss function is the L_2 norm squared of the error between the predicted binary vector (predicted wine quality) and actual binary vector (actual wine quality).
- I calculated two performance metrics and printed them in a pdf file:
  - root mean squared error (calculated as the L_2 norm of the error between predicted and ground-truth binary vectors)
  - accuracy (calculated as percentage of predictions that match the true labels)

Below are the learning curves for three different values of learning rate, where we can see that as the learning rate increases, the loss converges faster.:

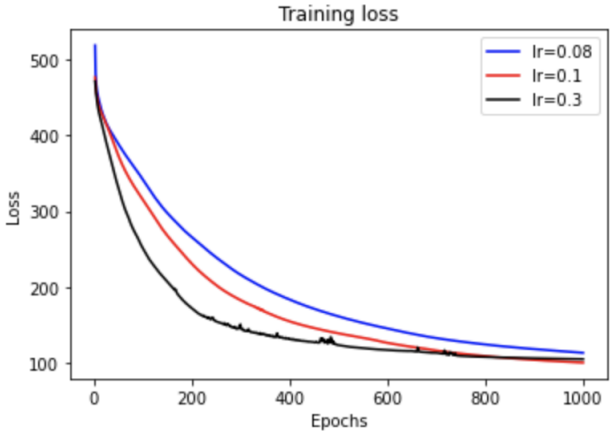

Figure 1: Learning curves for three different learning rates