Calculation Exercise 1: Multilayer Perceptron (MLP)

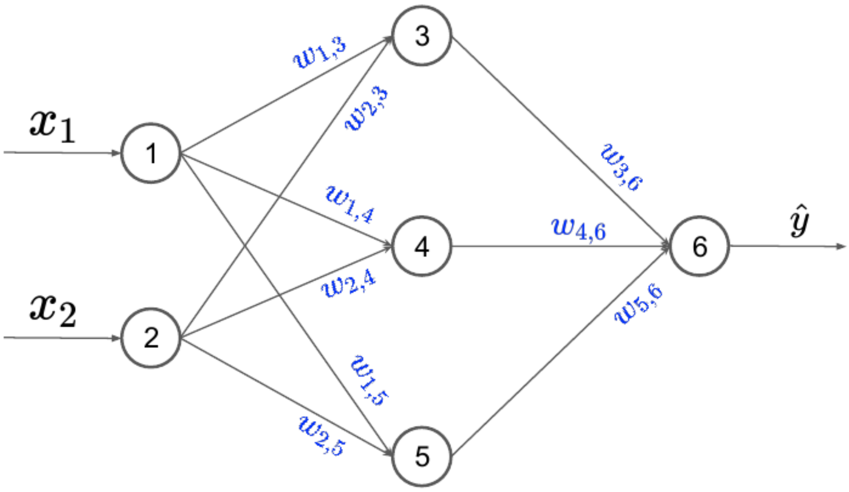

Figure 1: An MLP with one hidden layer (Question 1).

Parameter	Value
$w_{1,3}$	0.4
$w_{1,4}$	-0.2
$w_{1,5}$	-0.3
$w_{2,3}$	0.0
$w_{2,4}$	0.7
$w_{2,5}$	0.1
$w_{3,6}$	-0.2
$w_{4,6}$	0.5
$w_{5,6}$	-0.6

Table 1: Parameter values of the MLP (Figure 1).

Neuron	Activation function
a_1	None
a_2	None
a_3	ReLU
a_4	ReLU
a_5	ReLU
a_6	Sigmoid

Table 2: Activation functions of the MLP (Figure 1).

1.1 Compute the output of the network for $x = (x_1, x_2)^T = (1, 2)^T$

$$a_1 = 1, \quad a_2 = 2$$

$$a_3 = \text{ReLu}(0.4 \times 1 + 0 \times 2) = 0.4, \quad a_4 = \text{ReLu}(-0.2 * 1 + 0.7 * 2) = 1.2, \quad a_5 = \text{ReLu}(-0.3 * 1 + 0.1 * 2) = 0$$

The function inside the ReLu of a_5 -0.1, the Relu makes it 0.

$$a_6 = -0.2 \times 0.4 + 0.5 \times 1.2 + -0.6 \times 0 = 0.52, \quad \hat{y} = \sigma(a_6) = \frac{1}{1 + \exp(-0.52)} \approx 0.627$$

1.2 Assume the label of $x = (x_1, x_2)^T = (1, 2)^T$ is $y = 0$. If we use the Binary Cross Entropy (BCE) loss to train our MLP, what will be the value of the loss for (x, y) ?

$$L(y, \hat{y}) = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y})$$

Subbing $y = 0$ and $\hat{y} = 0.627$ we get:

$$L(0, 0.627) = 0 \times \ln(0.627) - (1 - 0) \ln(1 - 0.627) = -\ln(1 - 0.627) \approx 0.986$$

1.3 Now assume the label of $x = (x_1, x_2)^T = (1, 2)^T$ is $y = 1$. For BCE loss, what will be the value of the loss for (x, y)? Do you expect the loss to be bigger or smaller compared to the previous part? Why? Explain your answer and your observation.

The loss equation is the same as above. Subbing $y = 1$ and $\hat{y} = 0.627$ we get:

$$L(1, 0.627) = -1 \ln(0.627) - (1 - 1) \ln(1 - 0.627) = -\ln(0.627) \approx 0.467$$

The predicted output 0.6271 is closer to 1 than to 0. The BCE loss penalizes more when the prediction is far from the true label:

- Since the prediction was closer to 1, the loss for $(y = 1)$ is smaller.
- When $(y = 0)$, the prediction of 0.627 is more off, leading to a higher loss.

1.4 Assume the learning rate of the SGD is $lr = 0.1$. For a training sample $x = (x_1, x_2)^T = (1, 2)^T$ and $y = 0$, obtain the updated value of $w_{3,6}$.

$$\frac{\partial L}{\partial w_{36}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_6} \cdot \frac{\partial a_6}{\partial w_{36}}$$

$$a_6 = w_{3,6} \times a_3 + w_{4,6} \times a_4 + w_{5,6} \times a_{5}, \quad \hat{y} = \sigma(a_6) = \frac{1}{1 + \exp(-a_6)}$$

$$\frac{\partial L(y, \hat{y})}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}} \approx 2.68, \quad \frac{\partial \hat{y}}{\partial a_6} = a_6(1 - a_6) \approx 0.234, \quad \frac{\partial a_6}{\partial w_{36}} = a_3 = 0.4$$

Combining these all together we get:

$$\frac{\partial a_6}{\partial w_{36}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_6} \cdot \frac{\partial a_6}{\partial w_{36}} = 2.68 \times 0.234 \times 0.4 \approx 0.250$$

Now we can do a step with the optimiser:

$$w_{3,6(new)} = -0.2 - 0.1 \times 0.250 \approx -0.225$$

1.5 Using the assumptions from the previous part (i.e., the learning rate of the SGD is $lr = 0.1$, (i.e., the training sample is $x = (x_1, x_2)^T = (1, 2)^T$ and $y = 0$), obtain the updated value of $w_{2,5}$.

$$\frac{\partial L}{\partial w_{36}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_6} \cdot \frac{\partial a_6}{\partial a_5} \cdot \frac{\partial a_5}{\partial w_{2,5}}$$

$$\frac{\partial a_6}{\partial a_5} = w_{5,6} = 0.1, \quad \frac{\partial a_5}{\partial w_{2,5}} = 0$$

In this scenario, $a_5 = 0$ making $\frac{\partial a_5}{\partial w_{2,5}} = 0$.

$$\frac{\partial L}{\partial w_{36}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_6} \cdot \frac{\partial a_6}{\partial a_5} \cdot \frac{\partial a_5}{\partial w_{2,5}} = 2.68 \times 0.234 \times 0.1 \times 0 = 0$$

This makes the new updated weight the same:

$$w_{2,5(new)} = 0.1 - 0.1 \times 0 = 0.1$$

Calculation Exercise 2: Activation Function

$$z(x) = \begin{cases} e^x - e^{-x}, & -1 \leq x \leq 1 \\ e^1 - e^{-1}, & x > 1 \\ e^{-1} - e^1, & x < -1 \end{cases}$$

If we made the function $z_{1000}(x)$ it would look close to a step function with the following properties:

- $x < 0$ would result in $z_{1000}(x) = e^{-1} - e^1$
- $x > 0$ would result in $z_{1000}(x) = e^1 - e^{-1}$
- $x = 0$ would result in $z_{1000}(0) = 0$

  The resulting activation function would be:

$$\lim_{n \to \infty} z_n(x) = \begin{cases} e^{-1} - e^1, & x < 0 \\ 0, & x = 0 \\ e^1 - e^{-1}, & x > 0 \end{cases}$$

As the $z(x)$ is nested more it approaches the above function.