Name, Vorname: Wang, Jiahui	**Fach:** B.Sc.Mathematik	**Martriknr.**2992080
Name, Vorname: Zhao, Jiaqi	**Fach:** M.Sc.Autonome System	**Martriknr.** 3470190
Name, Vorname: Wang, Ziyin	**Fach:** M.Sc.Informatik	**Martriknr.** 3435397

Assignment 8

Task 1: Formalizing Neural Networks

Model chain-based architecture:

Input layer: $x = h^{(0)}, x \in \mathbb{R}^{12\times 1}$
First layer: $h^{(1)} = g^{(1)}(W^{(1)\top}x + b^{(1)}) \in \mathbb{R}^{n\times 1}$ with $W^{(1)} \in \mathbb{R}^{12\times n},\ b^{(1)} \in \mathbb{R}^{n\times 1}$
Second layer: $h^{(2)} = g^{(2)}(W^{(2)\top}h^{(1)} + b^{(2)}) \in \mathbb{R}^{m\times 1}$ with $W^{(2)} \in \mathbb{R}^{n\times m},\ b^{(2)} \in \mathbb{R}^{m\times 1}$
Output layer: $\hat{y} = h^{(3)} = softmax(W^{(3)\top}h^{(2)} + b^{(3)}) \in \mathbb{R}^{3\times 1}$ with $W^{(3)} \in \mathbb{R}^{m\times 3},\ b^{(3)} \in \mathbb{R}^{3\times 1}$

Active function: $g^{(i)}(z) = tanh(z)$ with $i = 1, 2$.
Other fuction: $softmax(z)_i = \frac{e^{z_i}}{\sum\limits_j e^{z_i}}$
Loss function(cross entropy): $L = \frac{1}{N}\sum\limits_i^N - \sum\limits_{c=1}^M \hat{y}_{ic}log(p_{ic}) = -\frac{1}{32}\sum\limits_{i=1}^{32}\sum\limits_{c=1}^{3}\hat{y}_{ic}log(p_{ic})$

Task 2: Feedforward Neural Network: Theory

1. Compute the forward-pass for the input $x = (2,\ -0.5)$:
   From the given task we know:

$$
\begin{aligned}
&x = h_0 = (x_1, x_2) = (2,\ -0.5)\\
&w_{11}^{(h)} = 0.5,\ w_{12}^{(h)} = -0.2,\ w_{21}^{(h)} = 0.6,\ w_{22}^{(h)} = 0.5,\\
&w_1^{(o)} = 1,\ w_2^{(o)} = -1
\end{aligned}
$$

   and for calculating we use the ReLu active function $g(z) = max\{0, z\}$.

$$
\begin{aligned}
First\ layer:\ h_1 &= w_{11}^{(h)}x_1 + w_{21}^{(h)}x_2 = 0.5\cdot 2 + (-0.5)\cdot 0.6 = 0.7 \Longrightarrow h_1' = g(h_1) = 0.7\\
Second\ layer:\ h_2 &= w_{12}^{(h)}x_1 + w_{22}^{(h)}x_2 = (-0.2)\cdot 2 + 0.5\cdot(-0.5) = -0.65 \Longrightarrow h_2' = g(h_1) = 0\\
Output:\ o &= w_1^{(o)}h_1' + w_2^{(o)}h_2' = 1\cdot 0.7 + (-1)\cdot 0 = 0.7 \Longrightarrow o' = g(o) = 0.7
\end{aligned}
$$

2. Compute the squared error loss for true value $y = 1.5$:

$$L(o',\ y) = (o' - y)^2 = (0.7 - 1.5)^2 = 0.64$$

3. Adjust the weight $w_{11}^{(h)}$ via back-propagation:
   We know:

$$\frac{\partial g(z)}{\partial z} = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Then we can calculate:

$$\frac{\partial L(o',\, y)}{\partial o'} = \frac{\partial (o'-y)^2}{\partial o'} = 2(o'-y) = -1.6$$

$$\frac{\partial o'}{\partial h_1'} = \frac{\partial g(o)}{\partial h_1'} = \frac{\partial g(w_1^{(o)} h_1' + w_2^{(o)} h_2')}{\partial h_1'} = w_1^{(o)} = 1$$

$$\frac{\partial h_1'}{\partial w_{11}^{(h)}} = \frac{\partial g(h_1)}{\partial w_{11}^{(h)}} = \frac{\partial g(w_{11}^{(h)} x_1 + w_{21}^{(h)} x_2)}{\partial w_{11}^{(h)}} = x_1 = 2$$

So we can compute the combination of gradient descent:

$$\frac{\partial L(o',\, y)}{\partial w_{11}^{(h)}} = \frac{\partial L(o',\, y)}{\partial o'} \frac{\partial o'}{\partial h_1'} \frac{\partial h_1'}{\partial w_{11}^{(h)}} = (-1.6) \cdot 1 \cdot 2 = -3.2$$

Now we can get the new weight $w_{11}^{(h)+}$:

$$w_{11}^{(h)+} = w_{11}^{(h)} - \eta \frac{\partial L(o',\, y)}{\partial w_{11}^{(h)}} = 0.5 - (0.1) \cdot (-3.2) = 0.82$$