Task

1. Goal: train a multiple-label classification neural network based on collected data.
2. Method: handwritten the whole process.
3. Expected output: the 'difference' between the prediction and the actual can gradually decrease.

Therefore, this blog will show you:

- **Forward propagation** process with **activation function** in the hidden layer and in the output layer.
- **Backward propagation** process with **derivative calculation** and **cross-entropy**.
- **Update parameters simultaneously**.

Do not worry, each item will be shown later. Okay, let us go through the whole process step by step.

Dataset

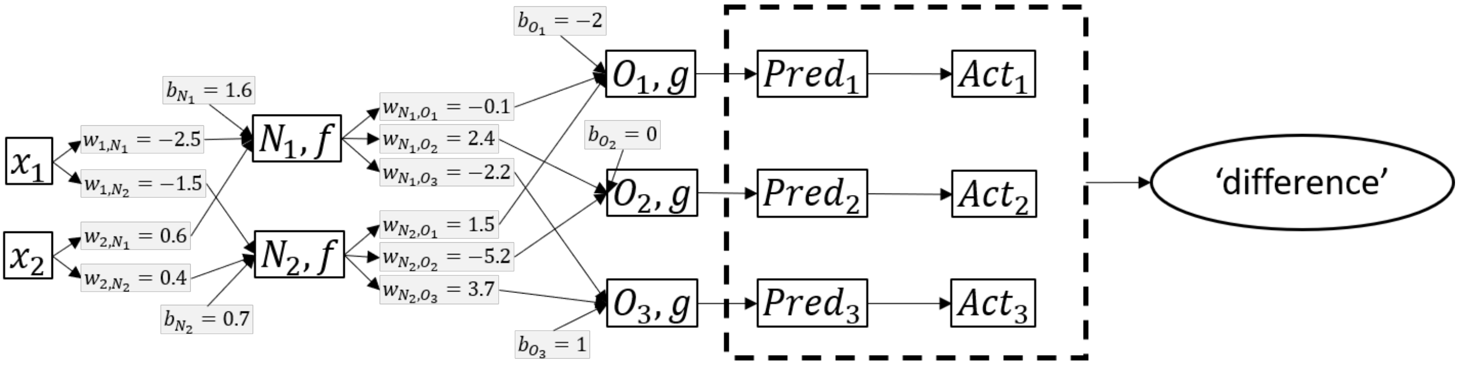
- 3 samples.
- 2 features.
- 3 target labels.

sample	x_1	x_2	Target(label)
1	0.04	0.42	0
2	1	0.54	1
3	0.5	0.65	2

Neural Network Architecture

A common neural network architecture involves:

1. Three layers and each layer has a number of nodes/neurons:
   - Input layer has 2 nodes x_1 and x_2. The number of nodes in the Input layer is the number of features in the dataset. In this example, each sample has 2 features.
   - Hidden layer has 2 nodes N_1 and N_2. The number of nodes in the Hidden layer is customized by us. In this example, we specify that there are two neurons.
   - Output layer has 3 nodes O_1, O_2 and O_3. The number of nodes in the Output layer is the number of unique targets. In this example, the dataset has 3 targets $(0, 1, 2)$.
2. Parameters (Weights and bias): In this example, parameters exist:
   - between Input layer and Hidden layer:
     - W_{1,N_1}, W_{2,N_1}, b_{N_1}
     - W_{1,N_2}, W_{2,N_2}, b_{N_2}
   - between Hidden layer and Output layer:
     - W_{N_1,O_1}, W_{N_2,O_1}, b_{O_1}
     - W_{N_2,O_1}, W_{N_2,O_2}, b_{O_2}
     - W_{N_2,O_3}, W_{N_2,O_3}, b_{O_3}
3. Two kinds of functions (one in the Hidden layer, and one in the Output layer):
   - Activation function f in the Hidden layer for each node. In this example, we use ReLu function $f(x) = \max(0, x)$, mainly because it will simplify the computational calculation.
   - Activation function g in the Output layer for each node. In this example, we use softmax functioin $g(x_i) = \frac{e^{x_i}}{\mathrm{sum}(e^{x_i})}$, mainly because this is a classification task.



Forward propagation

Taking $sample_1$ ($x_1 = 0.04$,$x_2 = 0.42$, target=0) as the example.

1. **Step1**: Get the values of nodes after the activation operation f in the hidden layer.

   - The input value $N_i^{input}, i = 1, 2$ of the node N_i on $sample_1$.

$$\begin{aligned} N_1^{input} &= x_1 w_{1,N_1} + x_2 w_{2,N_1} + b_{N_1 2} \\ &= 0.04 \times (-2.5) + 0.42 \times 0.6 + 1.6 \\ &= 1.752 \end{aligned} \tag{1}$$

$$\begin{aligned} N_2^{input} &= x_1 w_{1,N_2} + x_2 w_{2,N_2} + b_{N_1} \\ &= 0.04 \times (-1.5) + 0.42 \times 0.4 + 0.7 \\ &= 0.808 \end{aligned} \tag{2}$$

   - The output value $N_i^{output}, i = 1, 2$ of the node N_i on $sample_1$ with ReLU activation function $f(x) = \max(0, x)$.

$$\begin{aligned} N_1^{output} &= \max(0, N_1^{input}) \\ &= \max(0, 1.752) \\ &= 1.752 \end{aligned} \tag{3}$$

$$\begin{aligned} N_2^{output} &= \max(0, N_2^{input}) \\ &= \max(0, 0.808) \\ &= 0.808 \end{aligned} \tag{4}$$

2. **Step2**: Get the values of nodes after the softmax function g in the output layer.

   - The input value $O_i^{input}, i = 1, 2, 3$ of the node O_i in the hidden layer.

$$\begin{aligned} O_1^{input} &= N_1^{output} w_{N_1,O_1} + N_2^{output} w_{N_2,O_1} + b_{O_1} \\ &= 1.752 \times (-0.1) + 0.808 \times 1.5 - 2 \\ &= -0.9632 \end{aligned} \tag{5}$$

$$\begin{aligned} O_2^{input} &= N_1^{output} w_{N_1,O_2} + N_2^{output} w_{N_2,O_2} + b_{O_2} \\ &= 1.752 \times 2.4 + 0.808 \times (-5.2) + 0 \\ &= 0.0032 \end{aligned} \tag{6}$$

$$\begin{aligned} O_3^{input} &= N_1^{output} w_{1,N_3} + N_2^{output} w_{2,N_3} + b_{O_3} \\ &= 1.752 \times (-2.2) + 0.808 \times 3.7 + 1 \\ &= 0.1352 \end{aligned} \tag{7}$$

- The output value $O_i^{output}, i = 1, 2, 3$ of the node O_i in the hidden layer.

$$
\begin{aligned}
O_1^{output} &= \text{softmax}(O_1^{input}) = \frac{e^{O_1^{input}}}{e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}} \\
&= \frac{e^{-0.9632}}{e^{-0.9632} + e^{0.0032} + e^{0.1352}} \\
&= 0.1508
\end{aligned} \tag{8}
$$

$$
\begin{aligned}
O_2^{output} &= \text{softmax}(O_2^{input}) = \frac{e^{O_2^{input}}}{e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}} \\
&= \frac{e^{0.0032}}{e^{-0.9632} + e^{0.0032} + e^{0.1352}} \\
&= 0.3965
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
O_3^{output} &= \text{softmax}(O_3^{input}) = \frac{e^{O_3^{input}}}{e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}} \\
&= \frac{e^{0.1352}}{e^{-0.9632} + e^{0.0032} + e^{0.1352}} \\
&= 0.4525
\end{aligned} \tag{10}
$$

Till now, we know that $sample_1$ ($x_1 = 0.04$,$x_2 = 0.42$, target=0) goes through the forward propagation of the neural network and generates three predictions that are

- $Pred_1 = O_1^{output} = 0.1508$ means the 'probability' that $sampe_1$ is assigned with Target=0.
- $Pred_2 = O_2^{output} = 0.3965$ means the 'probability' that $sampe_1$ is assigned with Target=1.
- $Pred_3 = O_3^{output} = 0.4525$ means the 'probability' that $sampe_1$ is assigned with Target=2.

3. **Step3**: Calculate the 'difference' between the prediction and the actual value via Cross Entropy. The actual target observation of $sample_1$ is

   - $Act_1 = 1$ means the 'probability' that $sampe_1$is assigned with Target=0.
   - $Act_2 = 0$ means the 'probability' that $sampe_1$is assigned with Target=1.
   - $Act_3 = 0$ means the 'probability' that $sampe_1$is assigned with Target=2.

   Therefore, the Cross-Entropy(CE) of $sample_1$ with Target=0 is

$$
\begin{aligned}
CE^{sample_1} &= -\sum_i^M Act_i \log(Pred_i), M = \text{number of nodes in the hidden layer} \\
&= -Act_1 \times \log(Pred_1) - Act_2 \times \log(Pred_2) - Act_3 \times \log(Pred_3) \\
&= -1 \times \log(Pred_1) - 0 \times \log(Pred_2) - 0 \times \log(Pred_3) \\
&= -1 \times \log(Pred_1) = -1 \times \log(0.1508) = 1.8918
\end{aligned} \tag{11}
$$

We repeat **Step1**,**Step2** and **Step3** on $sample2$ ($x_1 = 1$,$x_2 = 0.54$, target=1) and $sample3$ ($x_1 = 0.5$,$x_2 = 0.65$, target=2), and we get followings:

- $sample2$ ($x_1 = 1$,$x_2 = 0.54$, target=1)
  1. $Pred_1 = 0.03511$, $Pred_2 = 0.2594$, $Pred_3 = 0.7053$
  2. $Act_1 = 0$, $Act_2 = 1$, $Act_3 = 0$
  3. $CE^{sample_2} = -0 \times \log(Pred_1) - 1 \times \log(Pred_2) - 0 \times \log(Pred_3) = -1 \times \log(Pred_2) = 1.34938$
- $sample3$ ($x_1 = 0.5$,$x_2 = 0.37$, target=2)

1. $Pred_1 = 0.0408$, $Pred_2 = 0.65338$, $Pred_3 = 0.30581$
2. $Act_1 = 0$, $Act_2 = 0$, $Act_3 = 1$
3. $CE^{sample_3} = -0 \times \log(Pred_1) - 0 \times \log(Pred_2) - 1 \times \log(Pred_3) = -1 \times \log(Pred_3) = 1.1847$

$CE^{total} = CE^{sample_1} + CE^{sample_2} + CE^{sample_3} = 1.8918 + 1.34938 + 1.1847 = 4.42588$
Therefore, the 'difference' between the predicted values and the target/actual values based on the whole dataset can be measured by the total cross-entropy $CE^{total} = 4.42588$. Next, in the backward propagation, we want to decrease the CE^{total} by adjusting parameters (weights and bias).

Backward Propagation

- The backward propagation aims to adjust the parameters simultaneously.
- The method to adjust the parameters simultaneously is to update them one by one.
- For example, when updating parameter b_{O_1}, other parameters will keep its previous values.
- The core idea is the updated b'_{O_1} is the previous b_{O_1} subtract the learning rate$r = 0.5$ times the 'difference', which is $b'_{O_1} = b_{O_1} - r \times$'difference', where 'difference'$= \frac{\mathrm{d}CE^{total}}{db_{O_1}}$ and r is the learning rate, and we can customize it as 0.5.
- $\frac{\mathrm{d}CE^{total}}{db_{O_1}}$ can be calculated as Equation.12 by the rule of the derivative of the sum is the sum of derivatives.

$$\frac{\mathrm{d}CE^{total}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}} + \frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}} + \frac{\mathrm{d}CE^{sample3}}{\mathrm{d}b_{O_1}} \tag{12}$$

In other words, the derivative of the total cross-entropy with respect to one scheduled parameter is the sum of the derivative of the cross-entropy generated from each sample with respect to the same schedule parameter.

1. Taking $\frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}}$ as an example. The key is to find the connection between the numerator $CE^{sample1}$ and the dominator b_{O_1}, which is $\frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}CE^{sample1}}{\mathrm{d}Pred_1} \times \frac{\mathrm{d}CE^{Pred_1}}{\mathrm{d}O_1^{input}} \times \frac{\mathrm{d}O_1^{input}}{\mathrm{d}b_{O_1}}$, and then solve it by the chain rule.
   (a) $CE^{sample1} = -\log(Pred_1^{sample1})$ as Equation.11, therefore,
   $$\frac{\mathrm{d}CE^{sample1}}{\mathrm{d}Pred_1^{sample1}} = \frac{\mathrm{d}}{\mathrm{d}Pred_1^{sample1}} - \log(Pred_1^{sample1}) = \frac{-1}{Pred_1^{sample1}} \tag{13}$$

(b) $Pred_1^{sample1} = \text{softmax}(O_1^{input})$ as Equation.8 by the quotient rule., therefore,

$$\begin{aligned}
\frac{\mathrm{d}Pred_1^{sample1}}{\mathrm{d}O_1^{input}} &= \frac{\mathrm{d}}{\mathrm{d}O_1^{input}}\text{softmax}(O_1^{input}) \\
&= \frac{\mathrm{d}}{\mathrm{d}O_1^{input}}\left(\frac{e^{O_1^{input}}}{e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}}\right) \\
&= \frac{\frac{\mathrm{d}}{\mathrm{d}O_1^{input}}e^{O_1^{input}}(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - e^{O_1^{input}}\frac{\mathrm{d}}{\mathrm{O_1^{input}}}(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})^2} \\
&= \frac{e^{O_1^{input}}(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - (e^{O_1^{input}})^2}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})^2} \\
&= \frac{e^{O_1^{input}}[(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - e^{O_1^{input}}]}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})^2} \\
&= \frac{e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})} \times \frac{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})} \\
&= \frac{e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})} \times \left(1 - \frac{e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})}\right) \\
&= Pred_1^{sample1} \times (1 - Pred_1^{sample1})
\end{aligned} \tag{14}$$

(c) $O_1^{input} = N_1^{output}w_{N_1,O_1} + N_2^{output}w_{N_2,O_1} + b_{O_1}$ as Equation.5, therefore,

$$\begin{aligned}
\frac{\mathrm{d}O_1^{input}}{\mathrm{d}b_{O_1}} &= \frac{\mathrm{d}}{\mathrm{d}b_{O_1}}(N_1^{output}w_{N_1,O_1} + N_2^{output}w_{N_2,O_1} + b_{O_1}) \\
&= 0 + 0 + 1 = 1
\end{aligned} \tag{15}$$

Therefore,

$$\begin{aligned}
\frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}} &= \frac{-1}{Pred_1^{sample1}} \times Pred_1^{sample1} \times (1 - Pred_1^{sample1}) \times 1 \\
&= Pred_1^{sample1} - 1
\end{aligned} \tag{16}$$

2. Taking $\frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}}$ as another example. That is $\frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}CE^{sample2}}{\mathrm{d}Pred_2} \times \frac{\mathrm{d}CE^{Pred_2}}{\mathrm{d}O_1^{input}} \times \frac{\mathrm{d}O_1^{input}}{\mathrm{d}b_{O_1}}$.

(a) Same as Equation.13

$$\frac{\mathrm{d}CE^{sample2}}{\mathrm{d}Pred_2^{sample2}} = \frac{\mathrm{d}}{\mathrm{d}Pred_2^{sample2}} - \log(Pred_2^{sample2}) = \frac{-1}{Pred_2^{sample2}} \tag{17}$$

(b) Similiar with Equation.14

$$\begin{aligned}
\frac{\mathrm{d}Pred_2^{sample2}}{\mathrm{d}O_1^{input}} &= \frac{\mathrm{d}}{\mathrm{d}O_1^{input}}\text{softmax}(O_2^{input}) \\
&= \frac{\mathrm{d}}{\mathrm{d}O_1^{input}}\left(\frac{e^{O_2^{input}}}{e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}}\right) \\
&= \frac{\frac{\mathrm{d}}{\mathrm{d}O_1^{input}}e^{O_2^{input}}(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - e^{O_2^{input}}\frac{\mathrm{d}}{\mathrm{d}O_1^{input}}(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})^2} \\
&= \frac{0 \times (e^{O_2^{input}} + e^{O_2^{input}} + e^{O_3^{input}}) - e^{O_2^{input}} \times e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})^2} \\
&= -\frac{e^{O_2^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})} \times \frac{e^{O_1^{input}}}{(e^{O_1^{input}} + e^{O_2^{input}} + e^{O_3^{input}})} \\
&= -Pred_2^{sample2} \times Pred_1^{sample2}
\end{aligned} \tag{18}$$

(c) Same Equation.15

$$\frac{\mathrm{d}O_1^{input}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}}{\mathrm{d}b_{O_1}}(N_1^{output}w_{N_1,O_1} + N_2^{output}w_{N_2,O_1} + b_{O_1}) \\ = 0 + 0 + 1 = 1 \tag{19}$$

Therefore,

$$\frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}} = \frac{-1}{Pred_2^{sample2}} \times (-Pred_2^{sample2} \times Pred_1^{sample2}) \times 1 \\ = Pred_1^{sample2} \tag{20}$$

3. Same as Equation.17, 18, 19, therefore,

$$\frac{\mathrm{d}CE^{sample3}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}CE^{sample3}}{\mathrm{d}Pred_3} \times \frac{\mathrm{d}CE^{Pred_3}}{\mathrm{d}O_1^{input}} \times \frac{\mathrm{d}O_1^{input}}{\mathrm{d}b_{O_1}} \\ = \frac{-1}{Pred_3^{sample3}} \times (-Pred_3^{sample3} \times Pred_1^{sample3}) \times 1 \\ = Pred_1^{sample3} \tag{21}$$

Therefore,

- $\frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}} = Pred_1^{sample1} - 1 = 0.1508 - 1 = -0.8492$. $Pred_1^{sample1}$ is the 'probability' that $sampe1$ (the superscript $sample1$ of $Pred_1^{sample1}$) is assigned with Target=0 which is the first (the subscript 1 of $Pred_1^{sample1}$) node in the Output layer.
- $\frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}} = Pred_1^{sample2} = 0.03511$. $Pred_1^{sample2}$ is the 'probability' that $sampe2$ (the superscript $sample2$ of $Pred_1^{sample2}$) is assigned with Target=0 which is the first (the subscript 1 of $Pred_1^{sample2}$) node in the Output layer.
- $\frac{\mathrm{d}CE^{sample3}}{\mathrm{d}b_{O_1}} = Pred_1^{sample3} = 0.0408$. $Pred_1^{sample3}$ is the 'probability' that $sampe3$ (the superscript $sample3$ of $Pred_1^{sample3}$) is assigned with Target=0 which is the first (the subscript 1 of $Pred_1^{sample2}$) node in the Output layer.

Therefore, the b_{O_1} will be updated as

$$\frac{\mathrm{d}CE^{total}}{\mathrm{d}b_{O_1}} = \frac{\mathrm{d}CE^{sample1}}{\mathrm{d}b_{O_1}} + \frac{\mathrm{d}CE^{sample2}}{\mathrm{d}b_{O_1}} + \frac{\mathrm{d}CE^{sample3}}{\mathrm{d}b_{O_1}} \\ = -0.8492 + 0.03511 + 0.0408 = -0.77329 \tag{22}$$

$b_{O_1}^{'} = b_{O_1} - r \times$ 'difference' $= -2 - 0.5 * (-0.77329) = -1.6133$

Till now, we have the newly updated b_{O_1} which is -1.6133. Congratulate!

If we goes through the forward propagation with this newly updated parameter b_{O_1}, we can calculate the new total cross-entropy $CE^{total} = CE^{sample1} + CE^{sample2} + CE^{sample3} = 1.5733 + 1.3657 + 1.2039 = 4.1429$ which is smaller than the previous $CE^{total} = 4.42588$! That is great! It means the parameters are going to be optimal for a smaller difference between the predicted value and the target value!