Exercise 4 :

Also called **logarithmic loss**, **log loss** or **logistic loss.**

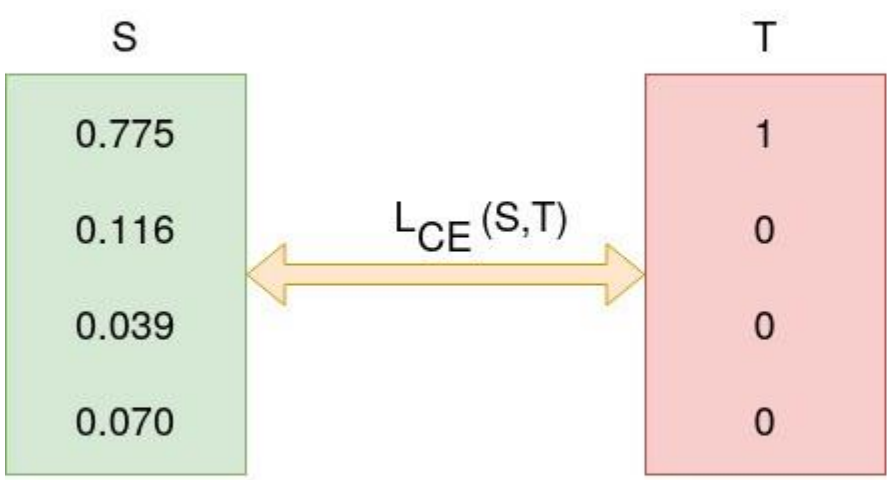
This function is mostly used in classification problems in machine learning. Because, for example, in binary classification, the answer to the problem is either 1 or 0. It can minimize the error by giving a penalty and points to each of these classes according to the probability of the predicted class. Penalizes the probability based on its distance from the expected true value. Binary cross entropy is defined as follows.

$$
\begin{aligned}
L &= -\sum_{i=1}^{2} t_i \log(p_i) \\
&= -[t_1 \log(p_1) + t_2 \log(p_2)] \\
&= -[t \log(p) + (1-t) \log(1-p)]
\end{aligned}
$$

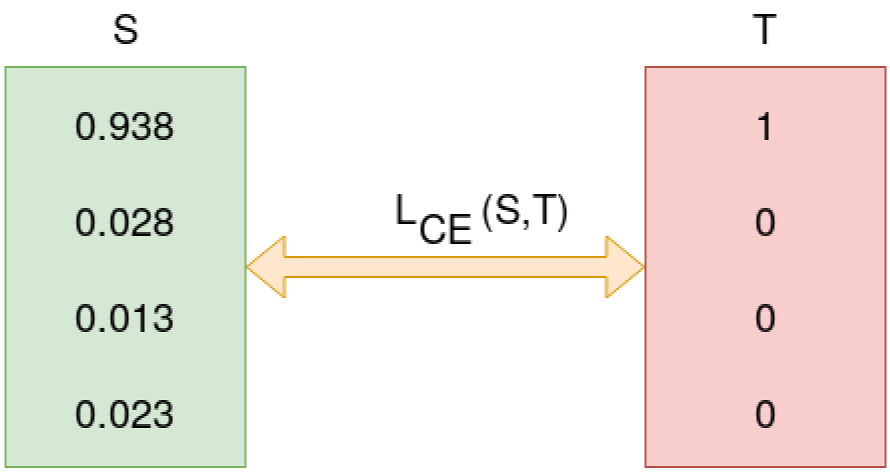
where t_i is the truth value taking a value 0 or 1 and p_i is the Softmax probability for the i^{th} class. Since we have two classes 1 and 0 we can have $t_1 = 1$ and $t_2 = 0$ and since $p's$ are probabilities then $p_1 + p_2 = 1 \implies p_1 = 1 - p_2$. For the convenience of notation, we can then let $t_1 = t, t_2 = 1 - t, p_1 = p$ and $p_2 = 1 - p$.

As we can see in the formula, when the probability of predicting the class is 0, it has no effect on our error rate, and it only calculates the error by giving a penalty to the class with a probability of 1, the closer the predicted value is to 1, according to the diagram The logarithm of its value is closer to 0 and the error rate is lower, and this model gives greater points to large differences close to 1.

On the other hand, it gives a small score for small differences that tend to 0.



$$
\begin{aligned}
L_{CE} &= -\sum_{i=1} T_i \log(S_i) \\
&= -\left[1\log_2(0.775) + 0\log_2(0.126) + 0\log_2(0.039) + 0\log_2(0.070)\right] \\
&= -\log_2(0.775) \\
&= 0.3677
\end{aligned}
$$



$$
\begin{aligned}
L_{CE} &= -1\log_2(0.936) + 0 + 0 + 0 \\
&= 0.095
\end{aligned}
$$

Cross entropy loss is used in classification tasks where we are trying to minimize the probability of a negative class by maximizing an expected value of some function on our training data

The idea of this loss function is to give a high penalty for wrong predictions and a low penalty for correct classification.

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

MSE :

But there is another loss function such as MSE which is used in regression models and its purpose is to minimize the predicted value of our data in the model. By calculating and summing the square of the difference between the actual and predicted data, the error can be obtained, which is always a value between 0 and infinity. It is not suitable for classified models because the impact of data that does not belong to the desired class can be high and destroy the model.

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$