#What is loss functions?
It's a mathematical method of evaluating how well specific algorithm models the given data. If predictions deviates too much from actual results, loss function would cough up a very large number

Most useful Loss Functions

1 Multinomial cross-entropy loss or Categorical Cross Entropy:
By this loss function we can able to calculate loss value for any no of classes that's why it is called as multinomial cross entropy
Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label

M(θ)=−1/n∑i,j yijlog(pij). Where i= 1 to n and j = 1 to m
Where i indexes samples/observations and j indexes classes

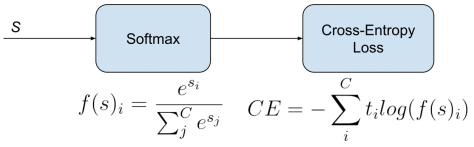
Pros:
Good for multi-class classification problem

2 Binary Cross Entropy:
Binary cross entropy is just a special case of categorical cross entropy. The equation for binary cross entropy loss is the exact equation for categorical cross entropy loss with one output node.
B(θ) = −1/n∑i=1 to n[yilog(pi)+(1−yi)log(1−pi)]

Pros:
Good for two-class classification problem

3 Multinomial Logistic Loss or Softmax Loss
It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a CNN to output a probability over the CC classes for each image. It is used for multi-class classification.
Formula:

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$



4 Sigmoid Cross-Entropy Loss

Sigmoid Cross-Entropy loss. It is a Sigmoid activation plus a Cross-Entropy loss.
Unlike Softmax loss it is independent for each vector component (class), meaning that the loss computed for every CNN output vector component is not affected by other component values. That's why it is used for multi-label classification

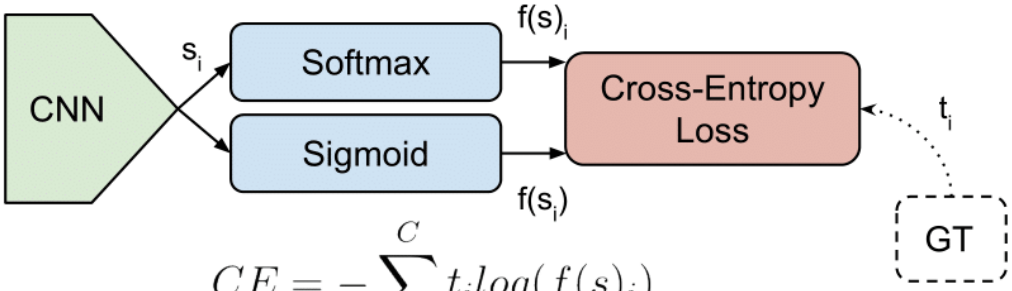
Formula

$$\frac{\partial}{\partial s_i}(CE(f(s_i)) = \begin{Bmatrix} f(s_i) - 1 & if & t_i = 1 \\ f(s_i) & if & t_i = 0 \end{Bmatrix}$$

$$CE = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1))$$

$$f(s_i) = \frac{1}{1 + e^{-s_i}}$$

Overall,



$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1))$$

5 Focal Loss

Focal Loss was introduced by Lin et al., from Facebook. They claim to improve one-stage object detectors using Focal Loss to train a detector they name RetinaNet. Focal loss is a Cross-Entropy Loss that weighs the contribution of each sample to the loss based in the classification error. The idea is that, if a sample is already classified correctly by the CNN, its contribution to the loss decreases. With this strategy, they claim to solve the problem of class imbalance by making the loss implicitly focus in those problematic classes.
Moreover, they also weight the contribution of each class to the lose in a more explicit class balancing. They use Sigmoid activations, so Focal loss could also be considered a Binary Cross-Entropy Loss. We define it for each binary problem as:
Formula:

$$FL = -\sum_{i=1}^{C=2}(1-s_i)^{\gamma} t_i log(s_i)$$

Where (1−si)γ, with the focusing parameter γ>=0, is a modulating factor to reduce the influence of correctly classified samples in the loss. With γ=0, Focal Loss is equivalent to Binary Cross Entropy Loss.

6 Least absolute error (L1)
L1 loss function are also known as Least Absolute Deviations
It is given by: ∑ (y-y`)
Where y = Real ground value
Y'= calculated value

7 Least square error (L2)
It is used to minimize the error which is the sum of all the absolute differences in between the true value and the predicted value.

It is given by: $\sum (y-y`)^2$
Where y = Real ground value
Y'= calculated value

8 Mean square error (L3)
MSE is the average of the squared error that is used as the loss function for least squares regression:
It is given by: $\sum (y-y`)^2/n$
Where y = Real ground value
Y'= calculated value
N = no of terms, i=1,2…n

9 Root mean square error:
RMSE is the square root of MSE. MSE is measured in units that are the square of the target variable, while RMSE is measured in the same units as the target variable. Due to its formulation, MSE, just like the squared loss function that it derives from, effectively penalizes larger errors more severely.
It is given by:
L = √(∑y-y´)^2/N)

Pros:
It is very popular in finding regression loss problem

10 Huber Loss
Huber Loss is often used in regression problems. Compared with L2 loss, Huber Loss is less sensitive to outliers(because if the residual is too large, it is a piecewise function, loss is a linear function of the residual).

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y-f(x))^2 & \text{for} |y-f(x)| \le \delta, \\ \delta\,|y-f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Among them, δ is a set parameter, yy represents the real value, and f(x) represents the predicted value.

The advantage of this is that when the residual is small, the loss function is L2 norm, and when the residual is large, it is a linear function of L1 norm

11 Hinge Loss

Hinge loss is often used for binary classification problems, such as ground true: t = 1 or -1, predicted value y = wx + b
L = max(0, 1-ty), t for no of classes