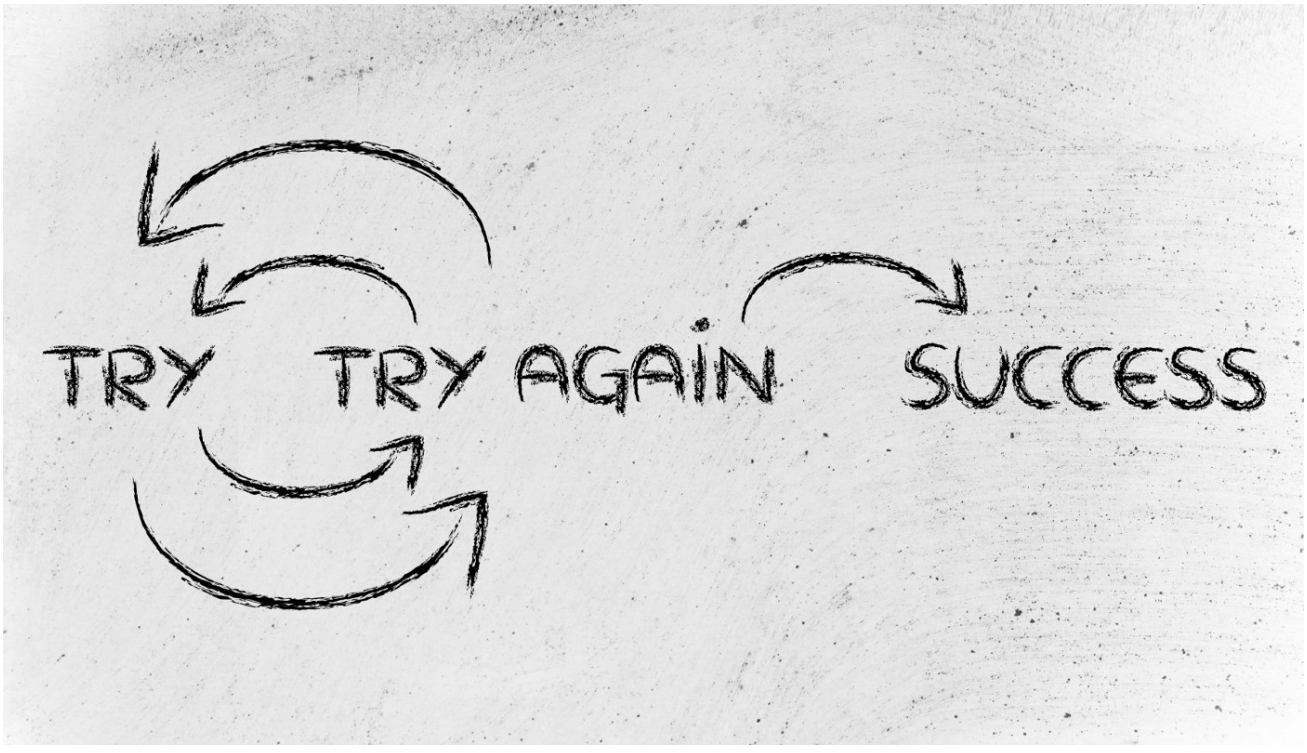
Common loss functions that you should know!

SOWMYA YELLAPRAGADA FEBRUARY 14, 2020

> Try try try until you succeed

That is the winning motto of life. Unsurprisingly, it is the same motto with which all machine learning algorithms function too. The model tries to learn from the behavior and inherent characteristics of the data, it is provided with. It then applies these learned characteristics to unseen but similar (test) data and measures its performance. It continually repeats this process until it achieves a suitably high accuracy or low error rate — **succeeds.**



Thus measuring the model performance is at the crux of any machine learning algorithm, and this is done by the use of **loss functions.** Choosing the right loss function can help your model learn better, and choosing the wrong loss function might lead to your model not learning anything of significance.

In this article series, I will present some of the most commonly used loss functions in academia and industry.

Mathematical representation of loss function

A loss function L maps the model output of a single training example to their associated costs. It takes as input the model prediction and the ground truth and outputs a numerical value.

> Loss function L: P x T → ℝ

where P is the set of all predictions, T is the ground truths and ℝ is real numbers set

Types of loss functions

Loss functions for regression :

Regression models make a prediction of continuous value. For example, predicting the price of the real estate value or stock prices, etc. The most commonly used loss functions in regression modeling are :

1. **Mean squared error (MSE):**

$$Loss = \frac{\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2}{n}$$

- The MSE loss function penalizes the model for making large errors by squaring them.
- This could both beneficial when you want to train your model where there are no outliers predictions with very large errors because it penalizes them heavily by squaring their error.
- If there are very large outliers in a data set then they can affect MSE drastically and thus the optimizer that minimizes the MSE while training can be unduly influenced by such outliers. The MSE value will be drastically different when you remove these outliers from your dataset. Minimizing MSE loss in such a scenario doesn't tell you much about the model performance. In that sense, the MSE is not "robust" to outliers
- This property makes the MSE loss function **less robust to outliers**. Therefore, it should not be used, if our data is prone to many outliers.

- **MSE loss is stable**
- The stability of a function can be analyzed by adding a small perturbation to the input data points. If the change in output is relatively small compared to the perturbation, then it is said to be stable.
- In the case of MSE loss function, if we introduce a perturbation of $\triangle$ << 1 then the output will be perturbed by an order of $\triangle^2$ <<< 1. Hence, MSE loss is a stable function

2. Mean absolute error (MAE):

$$Loss = \frac{\sum_{i=0}^{n} |Y_i - \hat{Y}_i|}{n}$$

- MAE loss is the average of absolute error values across the entire dataset.
- Unlike MSE, MAE doesn't accentuate the presence of outliers. Hence, MAE loss is **more robust to outliers** than MSE loss. Hence it is extremely useful when the data is prone to outliers, and the goal of your model is to perform well on most data points and not focus on a handful of outliers.

- Introducing a small perturbation $\triangle$ in the data perturbs the MAE loss by an order of $\triangle$, this makes it less stable than the MSE loss

3. **Huber Loss**

- The Huber loss combines the best properties of MSE and MAE.
- It is quadratic for smaller errors and is linear for larger errors
- It is identified by its delta parameter:

$$Loss_{\delta} = \frac{\sum_{i=0}^{n} L_i}{n}$$

$$where,\ if\ |L_i - \hat{L}_i| < \delta\ ,\ L_i = \frac{(L_i - \hat{L}_i)^2}{2}$$

$$else\ L_i = \delta|L_i - \hat{L}_i| - \frac{\delta^2}{2}$$

- Huber loss is more robust to outliers than MSE because it exchanges the MSE loss for MAE loss in case of large errors (the error is greater than the delta threshold), thereby not amplifying their influence on the net loss.
- If you would like your model to not have excessive outliers, then you can increase the delta value so that more of these are covered under MSE loss rather than MAE loss.
- It is used in **Robust Regression, M-estimation and Additive Modelling**.

Loss functions for binary classifications

Binary classification is a prediction algorithm where the output can be either one of two items, indicated by 0 or 1, (or in case of SVM, -1 or 1). The output of many binary classification algorithms is a prediction score. The score indicates the algorithm's certainty that the given observation belongs to one of the classes. For example, consider if the prediction is 0.6, which is greater than the halfway mark then the output is 1. Else, if the prediction is 0.3, then the output is 0. The most commonly used loss functions in binary classifications are —

1. **Binary Cross-Entropy or Log-loss error**

- Before we define cross-entropy loss, we must first understand **entropy**
- **Entropy** indicates disorder or uncertainty. It is measured for a random variable X with probability distribution p(X) as follows —

If x is continuous -

$$Entropy = -\int p(x)log(p(x))$$

If x is discrete -

$$Entropy = -\Sigma P(x)log(P(x))$$

- The negative sign is used to make the overall quantity positive.
- A greater value of entropy for a probability distribution indicates a greater uncertainty in the distribution. Likewise, a smaller value indicates a more certain distribution.

Binary Cross-Entropy or Log-loss error aims to reduce the entropy of the predicted probability distribution in binary classification problems. It is defined as follows —

$$Loss = \sum_{i=0}^{n} [-Y_i log(\hat{Y}_i) - (1 - \hat{Y}_i) log(1 - \hat{Y}_i)]$$

2. **Hinge Loss**

- Hinge Loss not only **penalizes the wrong predictions** but also the **right predictions that are not confident.**
- It is primarily used with Support Vector Machine (SVM) Classifiers with class labels -1 and 1, so make sure you change the label of your dataset are re-scaled to this range.
- It is used when we want to make real-time decisions with not a laser-sharp focus on accuracy.

Loss functions for multi-class classifications

Multi-class classification is an extension of binary classification where the goal is to predict more than 2 variables. A classic example of this is object detection from the ImageNet dataset. Most commonly used loss functions in multi-class classifications are —

1. **Multi-class Cross Entropy Loss**

- This is an extension to the binary cross-entropy or log-loss function, generalized to more than two class variables —

$$Loss = \sum_{i=0}^{n} max(0, 1 - Y_i \hat{Y}_i)$$

Here, C is the number of class labels

2. Kullback Leibler Divergence Loss (KL-Divergence)

- The Kullback-Liebler Divergence is a measure of how a probability distribution differs from another distribution.
- If the KL-divergence is zero, then it indicates that the distributions are identical
- For two probability distributions, P and Q, KL divergence is defined as —

- Note that KL divergence is not a symmetric function i.e.,
- Dkl(P||Q) != Dkl(Q||P)
- The goal of the **KL divergence loss** is to approximate the **true probability distribution P** of our target variables with respect to the input features, given some **approximate distribution Q**
- To do so, if we minimize Dkl(P||Q) then it is called **forward KL.** If we minimize Dkl(Q||P) then it is called **backward KL**

- KL-Divergence is functionally similar to multi-class cross-entropy and is also called relative entropy of P with respect to Q —

Here, H(P, P) = entropy of the true distribution P and H(P, Q) is the cross-entropy of P and Q

This concludes the discussion on some common loss functions used in machine learning.

Check out the next article in the loss function series here —

Also, head here to learn about how best you can evaluate your model's performance —

You may also reach out to me via sowmyayellapragada@gmail.com