Module 1

Loss functions:

What Are Loss Functions in Machine Learning?

The loss function helps determine how effectively your algorithm model the featured dataset. Similarly loss is the measure that your model has for predictability, the expected results. Losses can generally fall into two broad categories relating to real world problems: classification and regression. We must predict probability for each class in which the problem is concerned. In regression however we have the task of forecasting a constant value for a specific group of independent features.

What is Loss Function in Deep Learning?

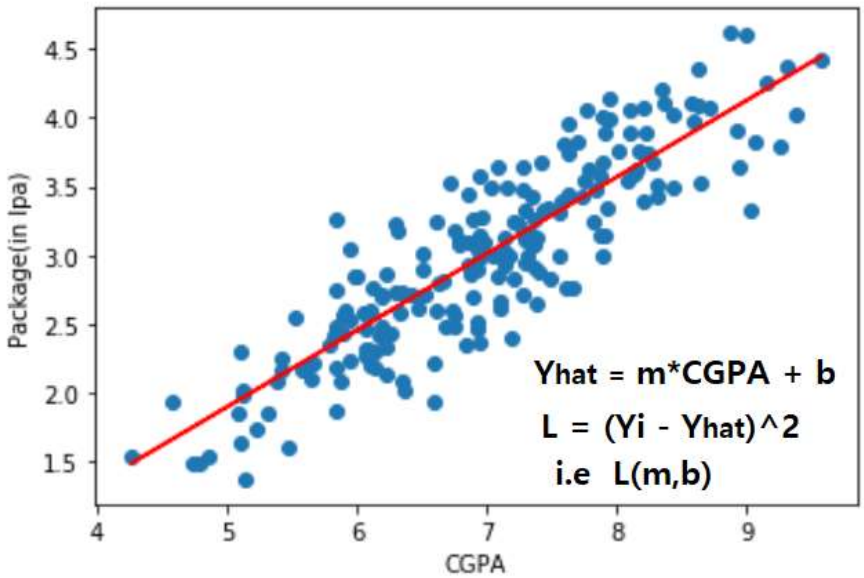
In mathematical optimization and decision theory, a loss or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

In simple terms, the Loss function is a method of evaluating how well your algorithm is modeling your dataset. It is a mathematical function of the parameters of the machine learning algorithm.

In simple linear regression, prediction is calculated using slope (m) and intercept (b). The loss function for this is the (Yi – Yihat)^2 i.e., loss function is the function of slope and intercept. Regression loss functions like the MSE loss function are commonly used in evaluating the performance of regression models. Additionally, objective

Module 1

functions play a crucial role in optimizing machine learning models by minimizing the loss or cost. Other commonly used loss functions include the Huber loss function, which combines the characteristics of the MSE and MAE loss functions, providing robustness to outliers in the data.



Why is the Loss Function Important in Deep Learning?

In mathematical optimization and decision theory, a loss or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

In simple terms, the Loss function is a method of evaluating how well your algorithm is modeling your dataset. It is a mathematical function of the parameters of the machine learning algorithm.

Module 1

In simple linear regression, prediction is calculated using slope (m) and intercept (b). The loss function for this is the (Yi – Yihat)^2 i.e., loss function is the function of slope and intercept. Regression loss functions like the MSE loss function are commonly used in evaluating the performance of regression models. Additionally, objective functions play a crucial role in optimizing machine learning models by minimizing the loss or cost. Other commonly used loss functions include the Huber loss function, which combines the characteristics of the MSE and MAE loss functions, providing robustness to outliers in the data.

Loss Functions in Deep Learning

Regression Loss Functions

1. Mean Squared Error/Squared loss/ L2 loss

The Mean Squared Error (MSE) is a straightforward and widely used loss function. To calculate the MSE, you take the difference between the actual value and the model prediction, square it, and then average it across the entire dataset.

$$MSE = \frac{1}{N}\sum_{i}^{N}(Yi - \hat{Y}i)^2$$

Advantage

- **Easy Interpretation:** The MSE is straightforward to understand.

Module 1

- **Always Differential:** Due to the squaring, it is always differentiable.
- **Single Local Minimum:** It has only one local minimum.

Disadvantage

- **Error Unit in Squared Form:** The error is measured in squared units, which might not be intuitively interpretable.
- **Not Robust to Outliers:** MSE is sensitive to outliers.

Note: In regression tasks, at the last neuron, it's common to use a linear activation function.

2. Mean Absolute Error/ L1 loss Functions

The Mean Absolute Error (MAE) is another simple loss function. It calculates the average absolute difference between the actual value and the model prediction across the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |Y_i - \hat{Y}_i|$$

Advantage

- **Intuitive and Easy:** MAE is easy to grasp.
- **Error Unit Matches Output Column:** The error unit is the same as the output column.

Module 1

- **Robust to Outliers:** MAE is less affected by outliers.

Disadvantage

- **Graph Not Differential:** The MAE graph is not differentiable, so gradient descent cannot be applied directly. Subgradient calculation is an alternative.

Note: In regression tasks, at the last neuron, a linear activation function is commonly used.

3. Huber Loss

The Huber loss is used in robust regression and is less sensitive to outliers compared to squared error loss.

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i - \hat{y}_i)^2 \qquad |y_i - \hat{y}_i| \leq \delta$$

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\delta\left(|y_i - \hat{y}_i| - \frac{1}{2}\delta\right) \qquad |y_i - \hat{y}_i| > \delta$$

- **n:** The number of data points.
- **y:** The actual value (true value) of the data point.
- **ŷ:** The predicted value returned by the model.
- **δ:** Defines the point where the Huber loss transitions from quadratic to linear.

Module 1

Advantage

- **Robust to Outliers:** Huber loss is more robust to outliers.
- **Balances MAE and MSE:** It lies between MAE and MSE.

Disadvantage

- **Complexity:** Optimizing the hyperparameter δ increases training requirements.

Classification Loss

1. Binary Cross Entropy/log loss Functions in machine learning models

It is used in binary classification problems like two classes. example a person has covid or not or my article gets popular or not.

Binary cross entropy compares each of the predicted probabilities to the actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

- yi – actual values
- yihat – Neural Network prediction

Module 1

Advantage –

- A cost function is a differential.

Disadvantage –

- Multiple local minima
- Not intuitive

Note – In classification at last neuron use sigmoid activation function.

2. Categorical Cross Entropy

Categorical Cross entropy is used for Multiclass classification and softmax regression.

loss function = -sum up to k(yjlagyjhat) where k is classes

$$\text{Loss} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

where k is number of classes in the data

cost function = -1/n(sum upto n(sum j to k (yijloghijhat))

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}[y_{ij}\log(\hat{y}_{ij})]$$

where

Module 1

- k is classes,
- y = actual value
- yhat – Neural Network prediction

Note – In multi-class classification at the last neuron use the softmax activation function.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

if problem statement have 3 classes

softmax activation – f(z) = ez1/(ez1+ez2+ez3)

When to use categorical cross-entropy and sparse categorical cross-entropy?

If target column has One hot encode to classes like 0 0 1, 0 1 0, 1 0 0 then use categorical cross-entropy. and if the target column has Numerical encoding to classes like 1,2,3,4….n then use sparse categorical cross-entropy.

Which is Faster?

Sparse categorical cross-entropy faster than categorical cross-entropy.

Module 1

Conclusion

The significance of loss functions in deep learning cannot be overstated. They serve as vital metrics for evaluating model performance, guiding parameter adjustments, and optimizing algorithms during training. Whether it's quantifying disparities in regression tasks through MSE or MAE, penalizing deviations in binary classification with binary cross-entropy, or ensuring robustness to outliers with the Huber loss function, selecting the appropriate loss function is crucial. Understanding the distinction between loss and cost functions, as well as their role in objective functions, provides valuable insights into model optimization. Ultimately, the choice of loss function profoundly impacts model training and performance, underscoring its pivotal role in the deep learning landscape.