Loss Functions in MLP (ANN)

* What is Loss Function?

In deep learning, the Loss Function is the feedback mechanism.

It is method for evaluating that how bad our model is performing.

High Loss → Model is performing bad.

Low Loss → Model is performing better.

Intution – Imagine ' A person giving an interview for Data science job role, so the feedback of interviewer in this case is the loss function of the model named as Data science Interview'.

* Loss vs Cost :-

(1) Loss Function – This calculates the error on single training example (one row of data).
For example calculating the difference between the predicted and actual salary for a student.

(2) Cost Function – This is the average of loss functions for entire training dataset (or batch)
It represent total error of model.

* How Loss Functions work in training? :-

(1) Forward Propagation - Input data (eg: cgpa, iq) enters in the Neural Network.
The network uses the random initial weights to make a prediction.

(2) Calculate Loss - We compare the prediction against the true value using loss function formula (eg. MSE)

(3) Backward Propagation - We use loss value to adjust the weights and biases using an optimizers (eg. gradient descent)

(4) Repeat - This happens for every student (data-point) and repeats multiple times (epochs) until the loss is minimized.

* Types of loss Functions :-

In MLP generally we have two types of Loss functions

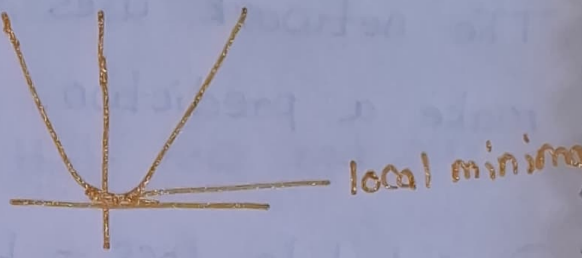
(1) For Regression

(2) For classification

(1) Regression Loss Functions :-

(i) Mean Squarred Error - It is the squarred distance of actual values from the predicted value.

$$L = (y_i - \hat{y}_i)^2$$



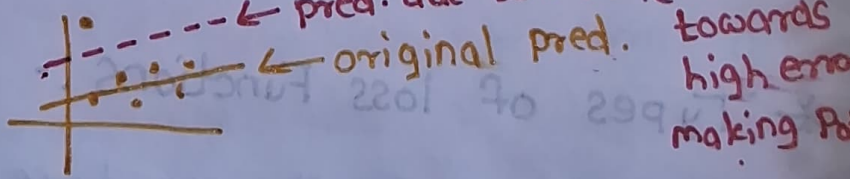
We take squares, because it ensures that during backpropagation the opposite signs does not cancel out each other.

Advantages -

It is easy to interpret

It is differentiable (The curve is smooth so it become easy to use during gradient descent)

It has only one local minima, so has only one perfect soln.



Disadvantages -

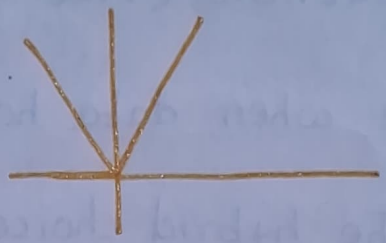
It is not robust to outliers (because due to squares it penalizes the errors more)

Unit mismatch due to squares. The errors are always in squarred units.

(ii) Mean Absolute Error (MAE) – It is the Positive difference between the actual and predicted value.

$$L = (|y_i - \hat{y}_i|)$$

$$C = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$



It is used when there are minor outliers in data, because it does not penelizes high errors highly.

Advantages –

Unit remain same as original data

Perform better when some outliers are present.

Disadvantages –

Not differentiable at ~~one~~ like MSE due to it is not a curve. (V shaped)

Time consuming.

(iii) Huber loss – This function attempts to fix the flaws of MSE & MAE by combining them.

It behaves like MSE when no outliers & behaves like MAE when too many outliers (Natural outliers)

Note -

MSE - Use for general regression with No or minimum outliers. ~~(True outliers)~~

MAE - Use when data have outliers (It is robust)

Hubber - The hybrid choice. Best of both world.

Use 'linear or No A.F' in output Layer while using these loss functions.

(2) Classification Loss Functions :-

(i) Binary cross Entropy - (Log Loss) -

$$L = -[y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})]$$

It is used for binary classification problem along with 'sigmoid' A.F. in output layer.

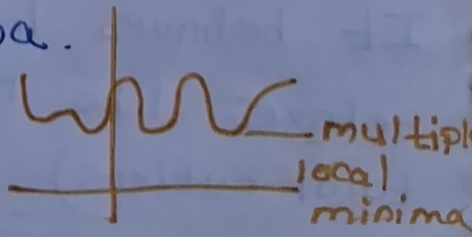
It penalizes the model heavily for making wrong class prediction.

Advantages -

It is differentiable.

Disadvantage -

It can have multiple local minima.



(ii) Categorical Cross Entropy :- It is used for Multiclass classification problem. (more than 2 classes)

$$L = -\sum_{j=1}^{k} y_j \cdot \log(\hat{y}_j)$$

where, k - No. of classes.

The output layer must have No. of Nuerons equal to classes.

i.e if No. classes = 10 then No. of Nuerons in out put layer = 10

The A.F must be 'Softmax' in output layer.

Advantages -

Can be useful in Multiclass classification problem.

Disadvantages -

It requires one-Hot encoding for target variable.

(iii) Sparse categorical cross entropy - Same as categorical cross entropy, but does not requires One-Hot encoding for target variable.

Note -

Binary classification → Binary cross entropy → sigmoid A.F. in O/P Layer

Multiclass classification → categorical cross entropy

↓

Softmax A.F in O/P layer.

(Requires One-Hot(y))

↓ skip

Sparse categorical cross Entropy.