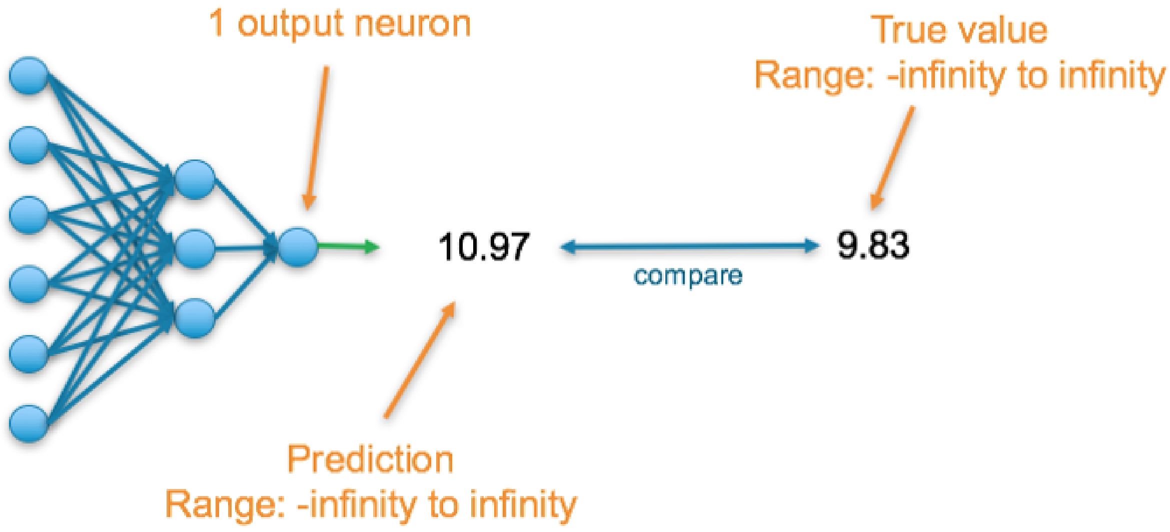
Which Loss and Activation Functions should I use?

- The motive of the blog is to give you some ideas on the usage of “Activation Function” & “Loss function” in different scenarios.
- Choosing an activation function and loss function is directly dependent upon the output you want to predict. There are different cases and different outputs of a predictive model. Before I introduce you to such cases let see an introduction to the activation function and loss function.
- The activation function activates the neuron that is required for the desired output, converts linear input to non-linear output. If you are not aware of the different activation functions I would recommend you visit my activation pdf to get an in-depth explanation of different activation functions click here : https://github.com/pratyusa98/ML_Algo_pdf/tree/main/01_Deep_Learning_PDF (https://github.com/pratyusa98/ML_Algo_pdf/tree/main/01_Deep_Learning_PDF).
- Loss function helps you figure out the performance of your model in prediction, how good the model is able to generalize. It computes the error for every training. You can read more about loss functions and how to reduce the loss https://github.com/pratyusa98/ML_Algo_pdf/tree/main/01_Deep_Learning_PDF (https://github.com/pratyusa98/ML_Algo_pdf/tree/main/01_Deep_Learning_PDF)..

Let’s see the different cases:

CASE 1: When the output is a numerical value that you are trying to predict

- Ex:- Consider predicting the prices of houses provided with different features of the house. A neural network structure where the final layer or the output later will consist of only one neuron that reverts the numerical value. For computing the accuracy score the predicted values are compared to true numeric values.



- Activation Function to be used in Output layer such cases,

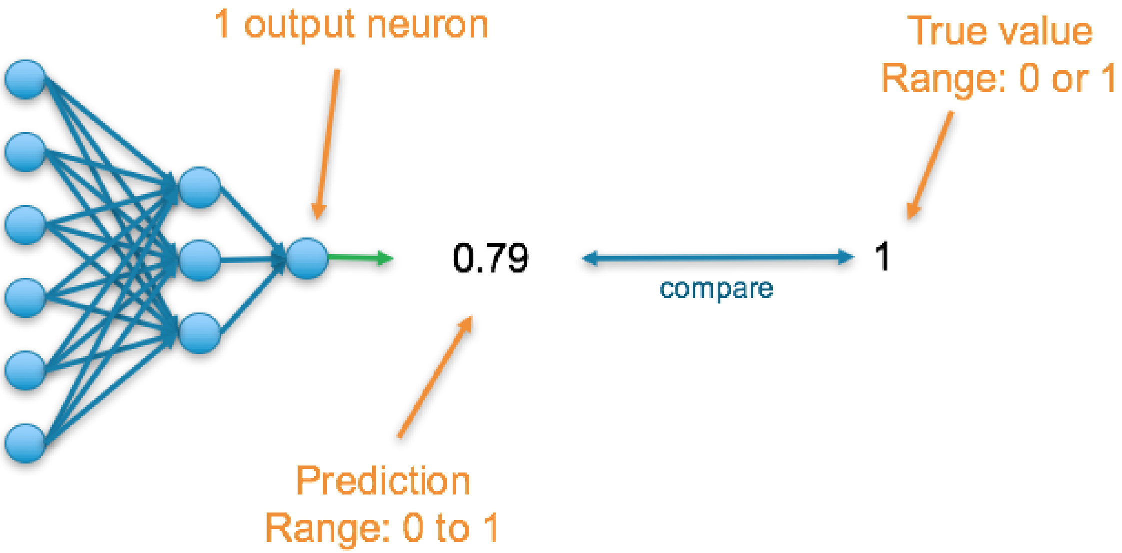
```
          * Linear Activation - it gives output in a numeric form th
at is the demand for this case. Or
          * ReLU Activation - This activation function gives you pos
itive numeric outputs as a result.
```

- Loss function to be used in such cases,

```
          * Mean Squared Error (MSE) - This loss function is respons
ible to compute the average squared difference    between the true v
alues and the predicted values.
```

CASE 2: When the output you are trying to predict is Binary

- Ex:- Consider a case where the aim is to predict whether a loan applicant will default or not. In these types of cases, the output layer consists of only one neuron that is responsible to result in a value that is between 0 and 1 that can be also called probabilistic scores.
- For computing the accuracy of the prediction, it is again compared with the true labels. The true value is 1 if the data belongs to that class or else it is 0.



- Activation Function to be used in Output layer such cases,

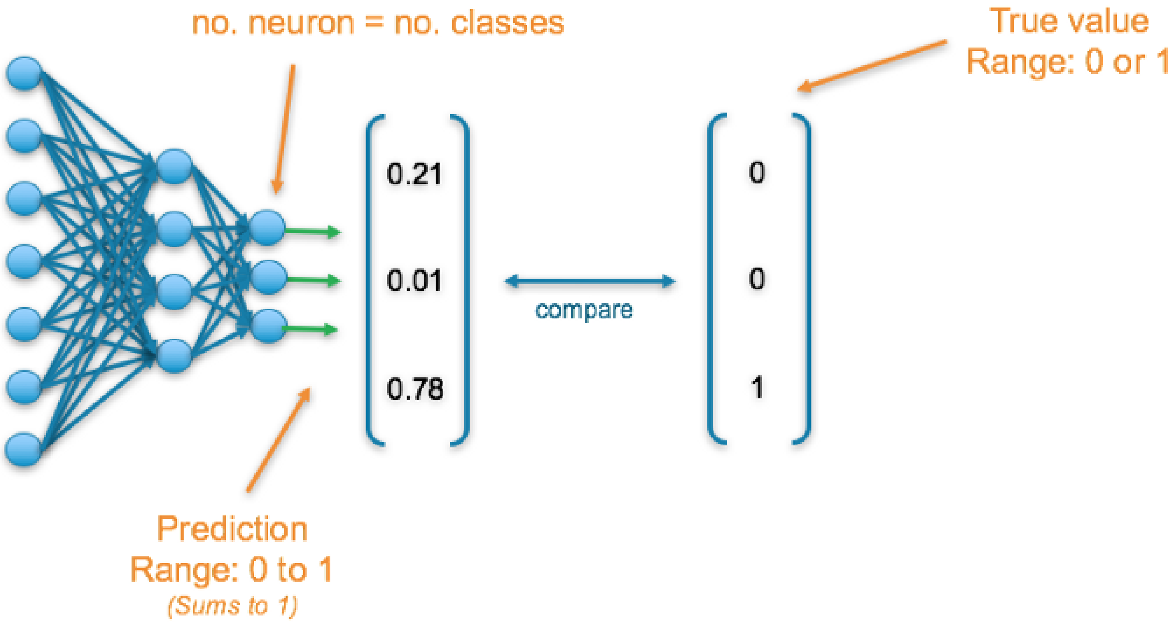
```
            * Sigmoid Activation -  This activation function gives
 the output as 0 and 1.
```

- Loss function to be used in such cases,

```
             * Binary Cross Entropy - The difference between the tw
o probability distributions is given by binary  cross-entropy. (p,1-
p) is the model distribution predicted by the model, to compare it w
ith true distribution, the   binary cross-entropy is used.
```

CASE 3: Predicting a single class from many classes

- Ex:- Consider a case where you are predicting the name of the fruit amongst 5 different fruits. In the case, the output layer will consist of only one neuron for every class and it will revert a value between 0 and 1, the output is the probability distribution that results in 1 when all are added.
- Each output is checked with its respective true value to get the accuracy. These values are one-hot-encoded which means if will be 1 for the correct class or else for others it would be zero.



- Activation Function to be used in Output layer such cases,

```
              * Softmax Activation -  This activation function gives
the output between 0 and 1 that are the probability scores which if
 added gives the result as 1.
```

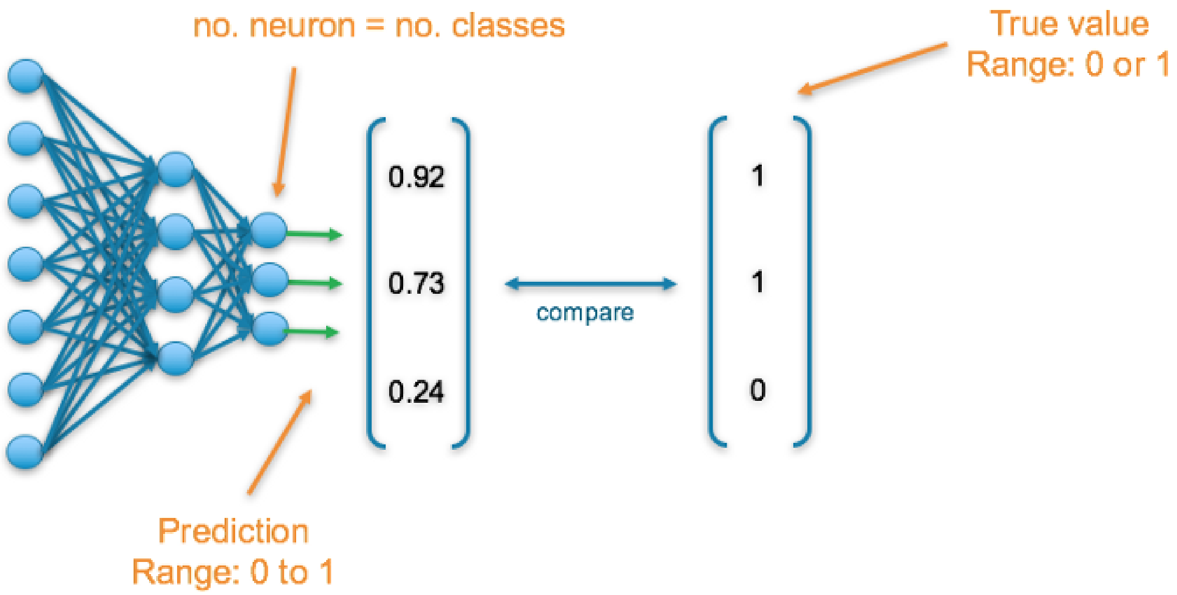
- Loss function to be used in such cases,

```
              * Cross-Entropy - It computes the difference between t
wo probability distributions.
              * (p1,p2,p3) is the model distribution that is predict
ed by the model where p1+p2+p3=1. This is compared with the true dis
tribution using cross-entropy.
```

CASE 4: Predicting multiple labels from multiple class

- Ex:- Consider the case of predicting different objects in an image having multiple objects. This is termed as multiclass classification. In these types of cases, the output layer consists of only one neuron that is responsible to result in a value that is between 0 and 1 that can be also

called probabilistic scores.
- For computing the accuracy of the prediction, it is again compared with the true labels. The true value is 1 if the data belongs to that class or else it is 0.



- Activation Function to be used in Output layer such cases,

```
                    * Sigmoid Activation -  This activation function gives the output as 0 and 1.
```

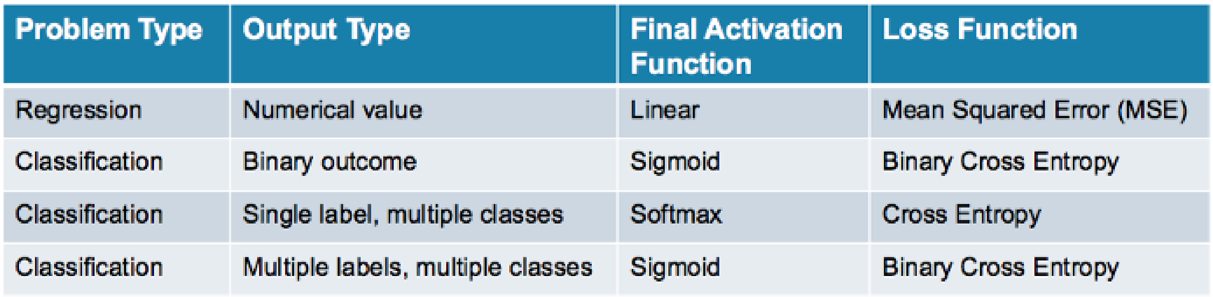
- Loss function to be used in such cases,

```
                    * Binary Cross Entropy - The difference between the two probability distributions is given by binary cross-entropy. (p, 1-p) is the model distribution predicted by the model, to compare it with true distribution, the binary cross-entropy is used.
```

All Losses : [https://keras.io/api/losses/ (https://keras.io/api/losses/)](https://keras.io/api/losses/)
All Activation : [https://keras.io/api/layers/activations/ (https://keras.io/api/layers/activations/)](https://keras.io/api/layers/activations/)

Summary

- This activation use only output layer and in hidden layer you can use Relu or Leaky Relu.
- The following table summarizes the above information to allow you to quickly find the final layer activation function and loss function that is appropriate to your use-case



Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy