Programming Test:

Learning Activations in Neural Networks

Monk AI

Ml, AI, DNN

Bangalore, Goa, Singapore

By: AVIRUP CHAKRABORTY

horizontal line

**Abstract -**

Artificial Neural Networks are inspired from the human brain and the network of neurons present in the brain. The information is processed and passed on from one neuron to another through neuro synaptic junctions. Similarly, in artificial neural networks there are different layers of cells arranged and connected to each other. The output/information from the inner layers of the neural network are passed on to the next layers and finally to the outermost layer which gives the output. The input to the outer layer is provided nonlinearity to inner layers’ output so that it can be further processed. In an Artificial Neural Network, activation functions are very important as they help in learning and making sense of non-linear and complicated mappings between the inputs and corresponding outputs.

Activation Functions are specially used in artificial neural networks to transform an input signal into an output signal which in turn is fed as input to the next layer in the stack. In an artificial neural network, we calculate the sum of products of inputs and their corresponding weights and finally apply an activation function to it to get the output of that particular layer and supply it as the input to the next layer. A Neural Network’s prediction accuracy is dependent on the number of layers used and more importantly on the type of activation function used. There is no manual that specify the minimum or maximum number of layers to be used for better results and accuracy of the neural networks but a thumb rule shows that a minimum 2 layers to be used. Neither is there any mention in literature of the type of activation function to be used. It is evident from studies and research that using a single/multiple hidden layer in a neural network reduces the error in predictions. A neural network’s prediction accuracy is defined by the type of activation function used. A neural network works just like a linear regression model where the predicted output is same as the provided input if an activation function is not defined. Same is the case if a linear activation function is used where the output is similar as the input fed along with some error. A linear activation function’s boundary is linear and if they are used, then the network can adapt to only the linear changes of the input but, in real world the errors possess non-linear characteristics which in turn with the neural networks ability to learn about erroneous data. The most appealing property of Artificial Neural Networks is the ability to adapt their behavior according to the changing characteristics of the system. In the last few decades many researchers and scientists have performed studies and investigated a number of methods to improve the performance of Artificial Neural Networks by optimizing the training methods, hyperparameter tuning, learn parameters or network structures but not much attention has been paid towards activation functions.

**ALGORITHM -**

1. **Importing required Libraries**
2. **Loading MNIST Dataset from Tensorflow.Keras**
3. **Splitting Data into Training and Validation Set**
4. **Building model using Keras Classifier**
5. **Performing RandomizedSearch to get the best tuned model**
6. **Using the best parameters to build a simpler model**
7. **Plotting Accuracy and Loss vs Epoch Plot**
8. **Predicting the values**
9. **Printing the required metrics to determine the performance of the model, which includes - F1, Precision,Recall,Validation Accuracy,Validation Loss, Confusion matrix.**

**GitHUB link to the code:**

https://github.com/Avich-It/ANN/blob/main/Optimized\_and\_Tuned\_simple\_ANN\_Model.ipynb