

Function-Approximation-using-Artificial-Neural-Network

This small self-project aims to approximate a non-linear function in two variables (within in a certain interval) through ANN. A simple approach is to use standard python libraries like TensorFlow, Scikit-Neural Network etc. However, this activity aims to implement a neural network architecture from scratch with deploying these high-level ML-libraries.

Below steps were followed:

- Program a python script to implement an ANN with one (or even two for additional complexities) hidden layer and **ReLU** activation.
- Generate 500 randomly selected data points within a specific interval for training and another 200 data points for testing.
- With a learning rate of 0.00001, train the above model for about 1 lakh epochs through forward propagation and backpropagation algorithms to optimize your weights and biases for each layer.
- Use Test Cost function to evaluate model predictions and hence analyse performance.

Function to Approximate:

$$f(x_1, x_2) = (1 - x_1)^2 + 100 \times (x_2 - x_1^2)^2$$

Within an interval of $[-1,1]$

For a fun activity, following can be tried as an extension of above exercise:

1. $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ within interval $[-5,5]$.
2. $f(x_1, x_2) = \sin(x_1) \times \exp(-x_2^2)$ within interval $[-10,10]$.