Assignments due on **07/02/2022**

Assignment 2a.

1. Run the code of 2 layer neural network from

https://stackabuse.com/creating-a-neural-network-from-scratch-in-python-adding-hidden-layers/

1. Go through the code and understand each line.
2. Read the dataset as shown there.
3. Two classes are there. Label 0 for one class, 1 for another class
4. Take 70% data for training. 30% data for testing
5. Build a two-layer neural network from from scratch where zo is the output.
   1. You will have two layers.
      1. One is hidden layers ( use 4 neurons)
      2. One output layer (use one output neuron)
   2. Use sigmoid function as activation function in each neuron.
   3. Use sum of square error as your loss function.
   4. Calculate the derivatives for back propagation.
6. Write codes training module for 2000 epochs to train the neural network.
7. Now classify the test data using the trained neural network.
   1. During the test, you will do only forward pass.
   2. If the forward pass score is >=0.5, make the class label 1. Otherwise 0.
   3. Report your accuracy.
8. Draw data points for training data and also **plot the class boundary** in 3D plot
9. Draw data points for test data and also **plot the class boundary** in 3D plot
10. You cannot use any built-in deep learning functions

Assignment 2b (same as 2a. But the input is 3X32X32 image instead of 3 numbers.

* Select any two classes from CIFAR 10 dataset. For example: Airplane, cat.
* So, airplane means numerical label 0, cat means 1
* Now, read one image, reshape it to row vector **x**. **x** is our input data now with dimension 1X3072.
* **x**’s class label would be either 0 or 1.
* **Now implement the same network architecture as above.**
* Build a two-layer neural network from scratch where Y\_pred is the output.
  + You will have two layers.
    - One is hidden layers ( use 4000 neurons)
    - One output layer (use one output neuron)
  + Use adam initializer to initialize the weights.
  + Use sigmoid function as activation function in each neuron.
  + Use the sum of square error as your loss function.
  + Calculate the derivatives for back propagation.
* Write codes training module for 10 epochs to train the neural network.
* Then, use the test images of airplane and cats from the test set.
* Supply the test images to the trained neural network and classify the images by using the Y\_pred as follows
  + If (y\_pred < 0.5) then class label is 0 else class label is 1.
* Report your accuracy on the test set
* You cannot use any built-in deep learning functions