Assignments due on 10/03/2020

Assignment 3.

1. Run the code of 2 layer neural network from for multi class classification
2. <https://stackabuse.com/creating-a-neural-network-from-scratch-in-python-multi-class-classification/>
3. Go through the code and understand each line.
4. Build the artificial dataset as shon in the blog.
5. Take 70% data for training. 30% data for testing
6. 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. three output layer (use one output neuron)
   2. Use sigmoid function as activation function in hidden layers
   3. Use softmax for output layer
   4. Use negative log likelihood loss
   5. Derive the derivative for softmax
   6. Calculate the derivatives for back propagation.
7. Write codes training module for 2000 epochs to train the neural network.
8. Now classify the test data using the trained neural network.
   1. During the test, you will do only forward pass.
   2. Argmax the forward pass output
   3. Report your accuracy.
9. Draw data points for training data and also plot the class boundary in 3D plot
10. Draw data points for test data and also plot the class boundary in 3D plot
11. You cannot use any built-in deep learning functions

Assignment 3b (same as 3a. But the input is 3X32X32 image, output 10 classes.. And three layer neural network

* Download and read train data from CIFAR 10 from 10 classes
* 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 one hot encoded
* **Use two hidden layer and one output layer.**
* Build a **three-layer neural network** from scratch
  + You will have three layers.
    - **First hidden layers ( use 1000 neurons)**
    - **Second hidden layers (use 100 neurons)**
    - **Output layers (10 output neurons)**
  + So. Output-probability:

**ao** = softmax ( **Wo** \* sigmoid ( **Wh2**\* sigmoid(**Wh1** \* **x** + **bh1) + bh2 ) + bo )**

* + Where **x** is input vector, Wh1, and Wh2 are weight matrices for hidden layers. Wo is the weight matrix for output layers. **bh1,** **bh2**, and **bo** are bias vectors for corresponding layers.
  + Use sum of cross entropy as your loss function. **ao** is the output vector.
  + Derive and Calculate the derivatives for back propagation for the three layer neural network.
* Write codes for training module for 10 epochs to train the neural network.
* Report your accuracy on the test set
* Show confusion matrix of your test prediction
* You cannot use any built-in deep learning functions

Assignment 3c (use functions) –

* Download and read train data from CIFAR 10 from 10 classes
* Same as assignment 3b.
* Write forward(X) function to implement the forward calculation.
* Use forward function to pass the input data to calculate the probabilities.
* Calculate the loss using criterion function with cross entropy option.
* Using the loss, use autograd and backward() to calculate the derivatives automatically.
* Write codes for training module for 10 epochs to train the neural network.
* Report your accuracy on the test set
* Show confusion matrix of your test prediction
* Explore internet blogs to get more information.