

Assignments due on 13/02/2022

### Assignment 3a.

- 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 shown in the blog.
- 5) Take 70% data for training. 30% data for testing
- 6) Build a two-layer neural network from scratch where  $z_o$  is the output.
  - a) You will have two layers.
    - i) One is hidden layers ( use 4 neurons)
    - ii) three output layer (use one output neuron)
  - b) Use sigmoid function as activation function in hidden layers
  - c) Use softmax for output layer
  - d) Use negative log likelihood loss
  - e) Derive the derivative for softmax
  - f) 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.
  - a) During the test, you will do only forward pass.
  - b) Argmax the forward pass output
  - c) 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 32X32X3 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  $\mathbf{x}$ .  $\mathbf{x}$  is our input data now with dimension 1X3072.
- ❖  $\mathbf{x}$ '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 layer ( use 4000 neurons)**
    - **Second hidden layer (use 400 neurons)**
    - **Output layer (10 output neurons)**
  - ☐ So. Output-probability:  
$$\underline{\mathbf{s}} = \text{softmax} ( \mathbf{W}_o * \text{sigmoid} ( \mathbf{W}_{h2} * \text{sigmoid}(\mathbf{W}_{h1} * \mathbf{x} + \mathbf{b}_{h1}) + \mathbf{b}_{h2} ) + \mathbf{b}_o )$$
  - ☐ Where  $\mathbf{x}$  is input vector,  $\mathbf{W}_{h1}$ , and  $\mathbf{W}_{h2}$  are weight matrices for hidden layers.  $\mathbf{W}_o$  is the weight matrix for output layers.  $\mathbf{b}_{h1}$ ,  $\mathbf{b}_{h2}$ , and  $\mathbf{b}_o$  are bias vectors for corresponding layers.
  - ☐ Use the sum of cross entropy as your loss function.  $\underline{\mathbf{s}}$  is the output vector of size 10.
  - ☐ Derive and Calculate the derivatives for back propagation for the three layer neural network.

- ❖ Write codes for a 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 the 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 a 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.