Assignment 4 Introduction to Deep Learning – CAP 4613

Due: Tuesday March 31 before 3:59pm.

Instructions:

• The entire assignment has to be submitted either "electronically on Canvas" or on "paper and should be turned in at the beginning of the class" before the deadline.

• Each assignment should have the following information on the first page: assignment number, student name and znumber, and a shareable link to the final version of your Python code in Colab.

> To get a shareable link, in your Colab notebook click 'Share' on the upper right corner, then click 'Get shareable link' and copy the link.

• The Python submission should include the codes and the generated outputs.

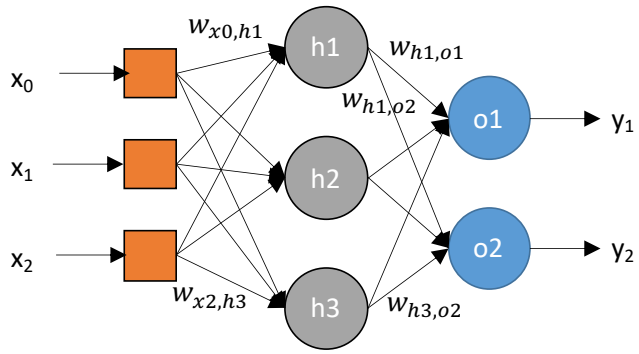
> To generate the PDF submission file: in Colab, go to 'File'=>'Print' then change the 'Destination' to 'Save as PDF' and save.

• **Only one file** has to be submitted electronically. Combine the handwritten and Python parts before submission. A penalty of 10% will be applied if the instructions are not followed.

> You can upload the pictures of your handwritten answers to the drive, and then import and show them in Colab notebook using matplotlib library.

• Filename for electronic submission: Student_Name_Assignmentxx.pdf or doc.

Problem 1) Forward-backward propagation: Consider the following forward feed neural network:




Initial weights are given as below. Learn rate is 0.5. The activation function is sigmoid function with $\alpha = 1$ for all the neurons.

$$w_{x_0,h_1} = 0.18, \quad w_{x_1,h_1} = 0.32, \quad w_{x_2,h_1} = 0.42$$

$$w_{x_0,h_2} = 0.51, \quad w_{x_1,h_2} = 0.64, \quad w_{x_2,h_2} = 0.12$$

$$w_{x_0,h_3} = 0.43,\ \ w_{x_1,h_3} = 0.72,\ \ \ w_{x_2,h_3} = \ 0.33$$

$$w_{h_0,O_1} = 0.53,\ \ w_{h_1,O_1} = 0.22,\ \ \ w_{h_2,O_1} = \ 0.19\ , w_{h_3,O_1} = \ 0.61$$

$$w_{h_0,O_2} = 0.61,\ \ w_{h_1,O_2} = 0.38,\ \ \ w_{h_2,O_2} = \ 0.21\ , w_{h_3,O_2} = \ 0.15$$

For a training, the inputs of the features ($x_1$=1, $x_2$=0) and the label of class $y_1$, perform the forward-backward steps as instructed below.

HINT: The data sample belongs to class $y_1$. This means that for this data sample, $y_1$=1 and $y_2$=0.

**NOTE: If you are not typing your answer, you need to submit your clean and neat handwritten solution with all the details. Otherwise, the assignment will not be graded.**

a) Perform the forward propagation from the left to the right side of the network. Show the detailed calculations in each step. Report the computed outputs in the table below:

| $y_{h_1}$ | $y_{h_2}$ | $y_{h_3}$ | $y_1$ | $y_2$ |
|---|---|---|---|---|
| | | | | |

b) Compute the local gradients of each node from the right to the left side of the network. Show the detailed calculations in each step. Report the computed local gradients in the table below:

| $\delta_{h_1}$ | $\delta_{h_2}$ | $\delta_{h_3}$ | $\delta_{O_1}$ | $\delta_{O_2}$ |
|---|---|---|---|---|
| | | | | |

c) Only for the bias weights of each neuron: compute the change in the weight and the updated weight. Show the detailed calculations in each step. Report the computed values in the table below:

| $\Delta w_{x_0,h_1}$ | $\Delta w_{x_0,h_2}$ | $\Delta w_{x_0,h_3}$ | $\Delta w_{h_0,O_1}$ | $\Delta w_{h_0,O_1}$ |
|---|---|---|---|---|
| | | | | |

| $w_{x_0,h_1}$ | $w_{x_0,h_2}$ | $w_{x_0,h_3}$ | $w_{h_0,O_1}$ | $w_{h_0,O_1}$ |
|---|---|---|---|---|
| | | | | |

Problem 2) Application of Keras to build, compile, and train a neural network to perform XOR operation:

a) Create an np array of shape 4x2 for the inputs and another 4x1 array for the labels of XOR.

| input | | desired |
|---|---|---|
| $x_1$ | $x_2$ | label |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

b) Plot the given data points with two different markers for each group.
c) Based on the plot from part (b), what is the minimum number of layers and nodes that is required to classify the training data points correctly? Explain.
d) Build the network that you proposed in part c using the Keras library.
e) Compile the network. Make sure to select a correct loss function for this classification problem. Use stochastic gradient descent learning (SGD, learning rate of 0.1). Explain your selection of the loss function.
f) Train the network for 200 epochs and a batch size of 1.
g) Use the trained weights and plot the final classifier lines in the plot of part (b).
h) Plot the training loss (i.e., the learning curve) for all the epochs.
i) Repeat steps (d) to (g) after adding 2 more nodes to the first layer and training for 400 epochs.
j) What behavior do you observe from the classifier lines after adding more nodes? Which number of nodes is more suitable in this problem? Explain.

Problem 3) Application of Keras to build, compile, and train a neural network as a three-class classifier for MNIST dataset (0 vs. 1 vs. 2):

a) Use *mnist* function in *keras.datasets* to load MNIST dataset and split it into training and testing sets. Then, randomly select 20% of the training images along with their corresponding labels to be the validation data.
b) Feature extraction: average the pixel values in the quadrants in each image to generate a feature vector of 4 values for each image.
c) Convert the label vectors for all the sets to binary class matrices using *to_categorical()* Keras function.
d) Build, compile, train, and then evaluate:
    i. Build a neural network with 1 layer that contains 10 nodes using the Keras library.
    ii. Compile the network. Make sure to select a correct loss function for this classification problem. Use stochastic gradient descent learning (SGD, learning rate of 0.0001). Explain your selection of the loss function.
    iii. Train the network for 50 epochs and a batch size of 16.
    iv. Plot the training loss (i.e., the learning curve) for all the epochs.

v. Use the *evaluate()* Keras function to find the training and validation loss and accuracy.

e) Repeat step (d) for each of the following networks:

| Model # | Details | Training | | Validation | |
|---|---|---|---|---|---|
| | | loss | accuracy | loss | accuracy |
| 1 | 1 layer<br>10 nodes | | | | |
| 2 | 1 layer<br>50 nodes | | | | |
| 3 | 1 layer<br>100 nodes | | | | |
| 4 | 2 layers<br>100 nodes,<br>10 nodes | | | | |
| 5 | 2 layers<br>100 nodes,<br>50 nodes | | | | |

f) What behavior do you observe in the training loss and the validation loss when you increase the number layers and nodes in the previous table. Which model is more suitable in this problem? Explain.

g) Evaluate the selected model in part (e) on the testing set and report the testing loss and accuracy.