Department of Computational and Data Sciences
DS226: Introduction to Computing for Artificial Intelligence and Machine Learning
November 13, 2022

Assignment 4

Instructions:

- **Submit a typed report in PDF format on Moodle. Use LaTeXpreferably. Handwritten reports will not be accepted.**
- **Solutions should be in the same order as the questions.**
- **Discussion is encouraged, but answers should be your own. Do not copy answers. Plagiarism will be penalised severely.**
- **For late submissions, the following late submission policy:**

Delay	% of Credit that will be considered
0-24 hours	**$99 - x$, example if late by 1.1 hour $x = 2$ (so will vary from 98 to 75)**
24-48 hours	**50**
48 hours - 1 week	**25**
Beyond 1 week	**No credit**

Deadline: 22nd November, 11:59 AM
Total Credits: 100

Nomenclature

α	Learning rate
$\mathbf{x}$	Input features, $\mathbf{x} = (x_1, x_2, \cdots, x_M)$
a_i^k	Activation of i^{th} neuron of k^{th} layer
h_i	Hidden neurons
K	Number of targets
M	Number of features
N	Number of samples (or observations)
w_0^i	Bias for i^{th} layer
w_{ij}^k	Weight between i^{th} neuron of k^{th} layer and j^{th} neuron of $(k+1)^{th}$ layer
y_i	Output of Neural Network

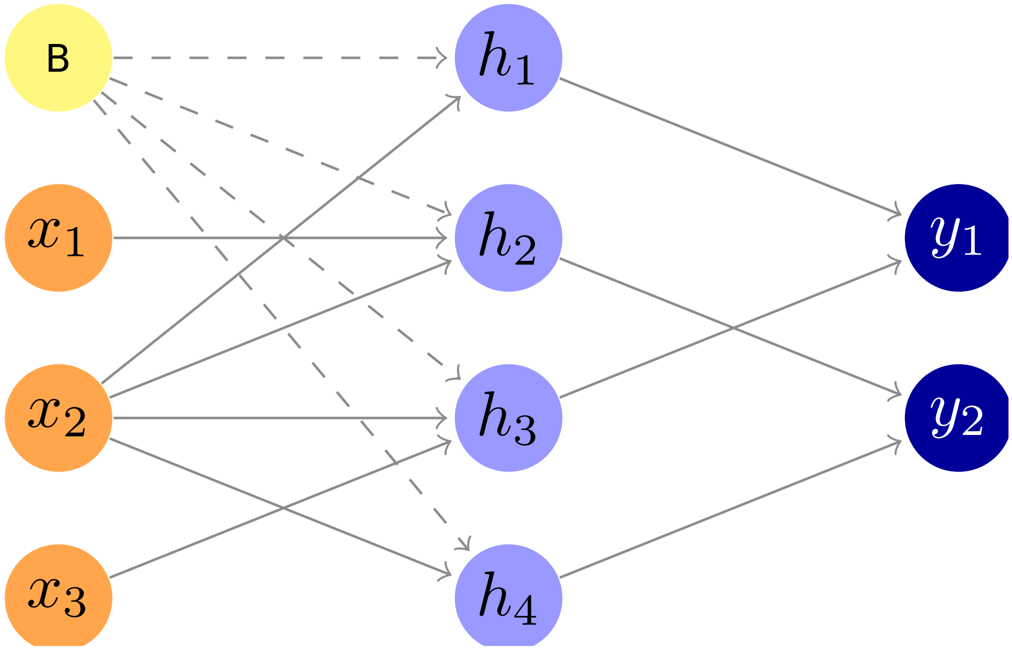

FIGURE 1. Neural network design for question 1

Question 1. Feed forward neural networks and backpropagation (20)

Consider the neural network shown in Figure 1. Here x_1, x_2, x_3 are the inputs, and outputs are the result of y_1, y_2 neurons. There is one hidden layer with neurons h_1, h_2, h_3, h_4 and bias B is applied only to the hidden layer.
Recall that an activation function decides whether a neuron should be *activated* or not, by caluclating the weighted sum of inputs to a neuron and further adding bias to it. Consider the following activation functions

$$identity(z) = z$$
$$sigmoid(z) = \frac{1}{1+\exp(-z)}$$
$$ReLU(z) = \begin{cases} 0 & , z < 0 \\ z & , z \geq 0 \end{cases}$$

The neurons h_1 and h_2 have **sigmoid** activation function, h_3 and h_4 have **ReLU** activation function. Output layer neurons y_1 and y_2 have **identity** activation function.
For the given neural network, the inputs, initial weights and outputs are as follows -

1) Inputs: $x_1 = 0.55$, $x_2 = 0.1$, $x_3 = 0.05$,
2) Weights between 1^{st} and 2^{nd} layer: $w^1_{12} = 0.15$, $w^1_{21} = 0.25$, $w^1_{22} = 0.2$, $w^1_{23} = 0.1$, $w^1_{24} = 0.05$,$w^1_{33} = 0.6$,
3) Weights between 2^{nd} and 3^{rd} layer: $w^2_{11} = 0.7$,$w^2_{22} = 0.45$,$w^2_{31} = 0.33$,$w^2_{42} = 0.8$,
4) Bias: $w^2_0 = 0.6$,
5) Target : $y_1 = 0.31$, $y_2 = 0.27$,
6) Learning rate : $\alpha = 0.5$

Solve the following with respect to the given neural network -

1) Derive the predicted outputs $\hat{y_1}$ and $\hat{y_2}$ as a function of the inputs, and calculate their values after one forward pass. (6)

2) Calculate the mean-squared error between the target and predicted outputs. Then, using one pass of backpropagation, compute the updated weights w^2_{31} and w^1_{21} (14)

Question 2. Reverse mode automatic differentiation (10)

For the function,

$$f(x_1, x_2) = \tanh\left(\frac{x_1}{x_2}\right) + sigmoid(x_1)$$

demonstrate the steps involved in calculating $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$ using Reverse Mode Automatic Differentiation.

Question 3. Neural Networks in Practice (30)

Consider the follow data-set: http://astro.utoronto.ca/ bovy/Galaxy10/Galaxy10.h5

The data-set contains 21785 pictures of 69×69 RBG pictures of galaxies. The galaxies are hand-labeled into the following 10 classes:

```
Class 0 : Disk, Face-on, No Spiral : 3461 images
Class 1 : Elliptical, Completely round : 6997 images
Class 2 : Elliptical, in-between round : 6292 images
Class 3 : Elliptical, Cigar shaped : 349 images
Class 4 : Disk, Edge-on, Rounded Bulge : 1534 images
Class 5 : Disk, Edge-on, Boxy Bulge : 17 images
Class 6 : Disk, Edge-on, No Bulge : 589 images
Class 7 : Disk, Face-on, Tight Spiral : 1121 images
Class 8 : Disk, Face-on, Medium Spiral : 906 images
Class 9 : Disk, Face-on, Loose Spiral : 519 images
```

Implement the following:

1) Randomly select 10 images from the data-set and display them along with their labels. Display the number of galaxies in each category in the data. Display the shape of the image data. [5]
2) Split the data into a 80 : 20 training and testing set and prepare it to be fed into a Convolution Neural Network(CNN) of the structure described below. Print the shape of the training and testing data. [5]
3) Construct a Convolution Neural Network(CNN) with three convolutional and four dense layers. Report which activation functions you have chosen for each layer and why. Print a summary of the model used. [10]
4) Train the model on the training data. Report which optimizer you have used along with the learning rate and the loss used. Provide justification for your choices. [10]
5) Report the training, validation and testing accuracies of your model. What conclusions can you draw from the training-validation accuracy plot? Can you identify any problems with the model from the plot? If so, how do you suggest to improve the model? [10]
6) Create two new models, one with an extra convolution layer and one with an extra dense layer. Report the model summaries. Draw comparisons between the change in the number of trainable parameters in both cases and explain the observed difference due to the introduction of the new layers. [10]

Submit the code and the plots and the answers to the questions in a detailed and well documented .ipynb notebook. Once you save your jupyter notebook make sure all the outputs are seen when it is reopened (without the need for compiling it again).