COMP 237 - Online lab assignment “Neural Networks”

Due date: End of week # 10

Purpose:
The purpose of this Lab assignment is to:

1. To get hands-on experience of training feed forward neural networks.
2. To experiment with different sizes of training data.
3. To experiment with different # of hidden layers.
4. To note the effect of the optimization algorithms.

General Instructions:
Be sure to read the following general instructions carefully:

1. This assignment must be completed individually by all the students.
2. Only provide the requested screenshots and make sure to have a complete screenshot, partial screenshots will not earn any marks.
3. You will have to provide a **demonstration video for your solution** and upload the video together with the solution on **eCentennial** through the assignment link. See the **video recording instructions** at the end of this document.
4. In your 5-minute demonstration video you should explain your solution clearly, going over the main code blocks and the purpose of each module/class/method also demoing the execution of exercises #1 & 2. Youtube links and links to google drive or any other media are not acceptable, the actual recording must be submitted.
5. Any submission without an accompanying video will lose 70% of the grade.
6. In your analysis report make sure you provide an introduction and clearly state the facts and findings. Any submission missing Analysis report will lose 70%.

Pre-requisite to carrying out the assignment:

1. Go through and watch all “Neural networks” lecture and lab tutorials related to modules # 8 & 9 to understand the concepts and the presented code.

Submission: Combine the code related to the exercises that requires code, into one python script and screenshot of output, as needed. Name the folder “Exercise_nn_firstname”, where firstname is your first name. (In total 1 folders for this assignment).

For all questions that require written or graphic response create one word document and indicate the exercise number and then state your response. Name the document “Written_responses_firstname”, where firstname is your firstname. (In total one word or pdf document).

Note: In all coding amendments, make sure to comment your code with explanations.

Assignment - exercises:

Exercise # 1: Single layer feed forward to recognize sum pattern (20 marks)

Requirements:

1. Create the training data(input) :
2. Using numpy random generate two sets of 10 numbers drawn from the uniform distribution, make sure to set the numbers to fall between -0.6 and +0.6. Save these numbers in a 10 by 2 ndarray where each set is considered a feature. Name the ndarray input_firstname. *where firstname is your firstname*.
3. Create the target data (output):

    i. The target is the sum of the two random values for each instance of the input data, for example:

Input data	
0.1	0.45
0.035	0.21
..	..

    1. Then the output data should be

Output data
0.55
0.245
……

    ii. i.e (y = x1+x2)

    iii. Store the output in a ndarray 10 by 1. Name the ndarray output_firstname. *Where firstname is your firstname*.

4. Set the seed = 1.
5. Using neurolab, create a simple neural network with two inputs, 6 neurons in the single layer and one output.
6. Train the network using the input, output data you created in points 1&2 above. Set the following parameter:
    a. show=15
    b. goal=0.00001
7. Train the network using the 10 data points.
8. Test / simulate the network by passing the following test values 0.1 and 0.2. Record the result under result #1.

Exercise # 2: Multi-layer feed forward to recognize sum pattern (20 marks)

1. Repeat steps 1-8 except for step #5 create a two layer feed forward network i.e two hidden layers the first with 5 neurons and the second with 3 neurons. Set the parameters as follows:
    a. epochs=1000
    b. show=100
    c. goal=0.00001

2. Record the result under result #2
3. Set the training algorithm to Gradient descent backpropogation
4. In your written response compare the result #1 to result #2 to the actual result explain your findings.

Exercise # 3: Single-layer feed forward to recognize sum pattern with more training data (20 marks)

1. Repeat steps 1-3 in exercise #1 but generate this time 100 random instances.
2. Repeat steps 4-8 in exercise #1
3. Record the result as result #3
5. In your written response compare result #1 to result #3 to the actual result explain your findings.

Exercise # 4: Multi-layer feed forward to recognize sum pattern with more training data (20 marks)

1. Repeat step #1 in exercise #3 (i.e. generate 100 samples)
2. Create a two layer feed forward network i.e two hidden layers the first with 5 neurons and the second with 3 neurons. Set the parameters as follows:
   a. epochs=1000
   b. show=100
   c. goal=0.00001
3. Set the training algorithm to Gradient descent backpropogation
4. Train the network using the 100 data points
5. Plot the error training size graph
6. Test / simulate the network by passing the following test values 0.1 and 0.2. Record the result under result #4.
7. In your written response compare the result #3 to result #4 to the actual result, explain your findings.

Exercise # 5: Three input multi-layer feed forward to recognize sum pattern with more training data (20 marks)

8. Repeat exercises # 1 but instead of having two inputs generate <u>three</u> inputs.
9. Test/Simulate the following test sample [0.2,0.1,0.2] record the results in result #5
10. Repeat exercise #4 but instead of having two inputs generate <u>three</u> inputs.
11. Test/Simulate the same test in point 10 i.e [0.2,0.1,0.2] record the results in result #6
12. In your written response compare the result #5 to result #6 to the actual result, explain your findings.

Write some final conclusion.

------ End of Exercises ------

Rubric applies to all exercises

Evaluation criteria	**Not acceptable**	**Below Average**	**Average**	**Competent**	**Excellent**
	0% - 24%	**25%-49%**	**50-69%**	**70%-83%**	**84%-100%**
Model building Validation &Testing 35%	No evidence of testing and evaluation of the requirements.	Minor evaluation and testing efforts.	Some of the requirements have been tested & evaluated.	Majority of requirements are tested & evaluated.	Realistic evaluation and testing, comparing the solution to the requirements.
Code Documentation 5%	No comments explaining code.	Minor comments are implemented.	Some code is correctly commented.	Majority of code is correctly commented.	All code is correctly commented.
Written analysis Content 35%	Missed all the key ideas; very shallow.	Shows some thinking and reasoning but most ideas are underdeveloped.	Indicates thinking and reasoning applied with original thought on a few ideas.	Indicates original thinking and develops ideas with sufficient and firm evidence.	Indicates synthesis of ideas, in-depth analysis and evidences original thought and support for the topic.
Written analysis Format and organization 5%	Writing lacks logical organization. It shows no coherence and ideas lack unity. Serious errors. No transitions. Format is very messy.	Writing lacks logical organization. It shows some coherence but ideas lack unity. Serious errors. Format needs attention, some major errors.	Writing is coherent and logically organized. Some points remain misplaced. Format is neat but has some assembly errors.	Writing is coherent and logically organized with transitions used between ideas and paragraphs to create coherence. Overall unity of ideas is present. Format is neat and correctly assembled.	Writing shows high degree of attention to logic and reasoning of all points. Unity clearly leads the reader to the conclusion. Format is neat and correctly assembled with professional look.
Demonstration Video 25%	Very weak no mention of the code changes. Execution of code not demonstrated.	Some parts of the code changes presented. Execution of code partially demonstrated.	All code changes presented but without explanation why. Code demonstrated.	All code changes presented with explanation, exceeding time limit. Code demonstrated.	A comprehensive view of all code changes presented with explanation, within time limit. Code demonstrated.

Demonstration Video Recording

Please record a short video (max 4-5 minutes) to explain/demonstrate your assignment solution. You may use the Windows 10 Game bar to do the recording:

1. Press the Windows key + G at the same time to open the Game Bar dialog.
2. Check the "Yes, this is a game" checkbox to load the Game Bar.
3. Click on the Start Recording button (or Win + Alt + R) to begin capturing the video.
4. Stop the recording by clicking on the red recording bar that will be on the top right of the program window.

(If it disappears on you, press Win + G again to bring the Game Bar back.)
You'll find your recorded video (MP4 file), under the Videos folder in a subfolder called Captures.

Or you can use any other video recording package freely available.