**Lab Assignment #2 – Train and Use MLPs to solve various prediction problems**

Due Date: Sunday 11:59pm, Week 6

Purpose: The purpose of this Lab assignment is to:

1. Design and implement MLP networks to predict/classify:

* Practice Backpropagation algorithm
* Practice the design and implementation of MLPs
* Practice the use of TensorFlow to implement MLPs

References: Read textbook, chapter 10, Neural Networks and Deep Learning book, chapter 2, and the lecture slides. This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

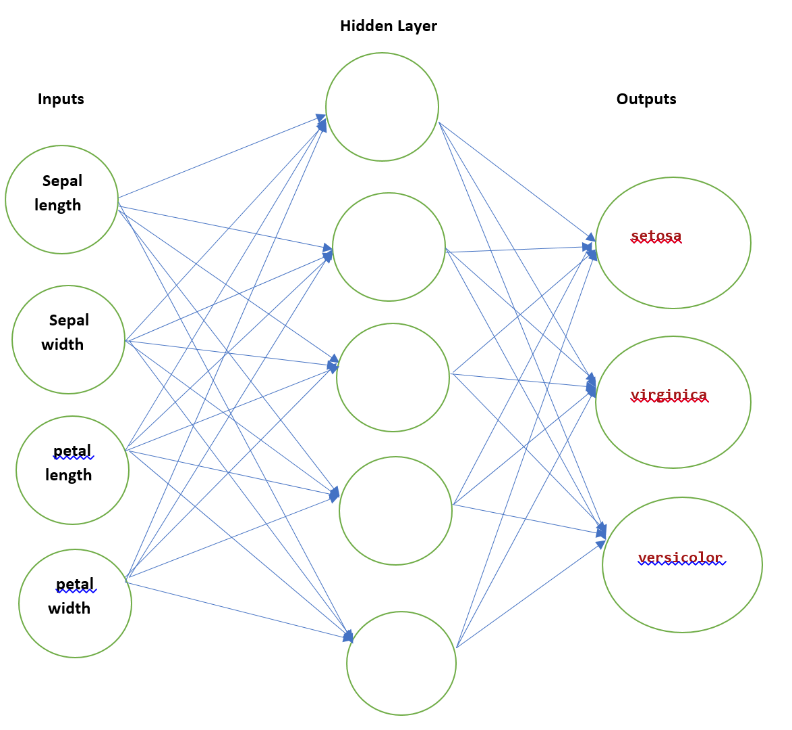
- This assignment must be completed individually by all the students.

- See the naming and **submission rules** at the end of this document

- 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.

**Exercise 1: Backpropagation Algorithm**

Write a Python program to **classify the different species** of the iris flower. The iris dataset is provided on eCentennial. You will **implement the backpropagation algorithm in a similar way to the code from BackPropEx1 example**. Use the following net architecture:



(4 marks)

Your output should **display the results of classification for the test examples in a friendly format**.

**Exercise 2: MLP Training Using Backpropagation for CIFAR10 dataset**

Write a Python program to train CIFAR10 dataset (<http://www.cs.toronto.edu/~kriz/cifar.html>) using Backpropagation. Use TensorFlow.

You will design an MLP network and train the network using backpropagation. Make the necessary adjustments to the net architecture to achieve a high accuracy. Build an image classifier using the **sequential API**. Compile, train, and evaluate the model. Use the model to make predictions.

1. Try different network architectures. What network architecture produces the best result?

* Fine-Tune Neural Network Hyperparameters (number of hidden layers, number of neurons per hidden layer, activation functions),

1. Try different learning algorithms. Which one worked better?
2. Try different parameters of the learning algorithms. Which ones produced the best results?

* Fine-Tune learning rate, batch size, etc.

Your output should **display the results of image classification in a friendly format**.

(6 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality:**   * Correct implementation of requirements * Code demonstration and brief explanation in a short video | 70%  10% |
| **Algorithm design**:   * correct design of classes and methods similarly to class examples * Correct use of naming guidelines and for classes, variables, methods. Good use of comments. | 15%  5% |
| **Total** | 100% |

You must **name your Jupyter notebook file** according to the following rule:

**YourFullname\_COMP258Labnumber\_Exercisenumber**.

Example: **JohnSmith\_COMP258Lab2\_Ex1**

Provide your **student number and full name as a comment** at the top of your code for each exercise.

**Submission rules:**

Submit your solution as a **zip file** that is named according to the following rule:

**YourFullname\_COMP258Labnumber.zip**

Example: **JohnSmith\_COMP258Lab2.zip**

Use 7-zip to compress files (https://www.7-zip.org/download.html).

**Demonstration Video Recording**

Please record a short video (max 3-4 minutes) to 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.

Submit the video together with your solution.