**COMP 4190 Assignment 2: Machine Learning Using TensorFlow**

*Release Date: Wednesday, 14th Feb, 2018*

*Due Date: 11:30am on Friday, 9th March 2018*

*Demo Date: same as due.*

# **Content**

In this assignment, you must use the [TensorFlow](https://www.tensorflow.org/) library in Python, a high performance “programming language” for mathematical calculations developed by Google.

# **2. Linear Regression (20%)**

Extend the TensorFlow sample program we seen in class to fit a parabola (i.e., 2nd order polynomial, *y=ax2 + bx + c* ) instead of a straight line to a data set. You should also generate a plot of your approximation (save the plots).

Generate randomized training and test data using the following equation:-

y = 2x2 - 14x + 18 + 0.9\*random()

# **3. Overfitting: Polynomial Regression (30%)**

Given is the following [data](https://docs.google.com/spreadsheets/d/1VkNn-wuQXgMQmMFgAt18mMrgR0ny-tw0ttzS8kTo_r4/edit?usp=sharing) set (training data) which was generated from a polynomial with an unknown distribution. Extend the TensorFlow program to fit the following polynomials to the data.

* Straight line: *y = ax + b*
* Parabola: *y = ax2 + bx + c*
* Cubic: *y = ax3 + bx2 + cx + d*
* 4th order: *y = ax4 + bx3 + cx2 + dx+e*
* 5th order: *y = ax5 + bx4 + cx3 + dx2+ex+f*

**Create a simple text file (no word .doc file) "overfitting.txt" which**

a. Evaluatethe performance of your system using the test data in [data](https://docs.google.com/spreadsheets/d/1VkNn-wuQXgMQmMFgAt18mMrgR0ny-tw0ttzS8kTo_r4/edit?usp=sharing) set. Show the error of each degree of polynomial

b. Explain which order polynomial results in the **smallest** error on the test set.

c. Explain how these results relates to the concept of **overfitting** in machine learning.

# **4. Myo Armband Keyboard (50%)** In this part, you are trying to detect a key that was entered on a [paper keyboard](https://docs.google.com/document/d/1urHwRPF4KpEKS15sYqIhx5ueZwJUAHySiM-J5Qtlsyw/edit?usp=sharing) by the user using the [Myo armband](https://www.myo.com/). The keyboard layout allows entering five commands: Forward, Left, Backward, Right, and Enter. During an experiment, raw sensor [data](https://drive.google.com/file/d/1aXSGIHo2UglgUUS28LxpVDlb_CFFMsfy/view?usp=sharing) from the Myo armband was collected by pressing each key about 10 times in succession with short breaks in between. There is a directory each for each key as well as separate csv files for the accelerometer, gyroscope, orientation as quaternion, orientation using Euler angles, and the EMG ( Electromyography) data respectively. The EMG data is the most interesting data for keyboard detection as the arm did not move much between pressing a key.

## **4.1 Set Detection**

Firstly, **implement a program that separates each file into the different sets**. For example, your program should look for periods of no change in all three axes and then split the sets at this point. Not that your system has to be robust enough to ignore the changes in the accelerometer readings due to noise.

Implement a **multi-layer perceptron** in TensorFlow that uses [softmax](https://en.wikipedia.org/wiki/Softmax_function) to classify a data series into different commands. You should implement **at least** a standard 3 level perceptron with 4 hidden units.

But you can also experiment with different network architectures and optimization algorithms as provided by TensorFlow.

## **4.2 Multi-axis Composition**

So far, we have dealt with the sensor data separately (e.g., emg1, emg2, etc.). However, it may be advantageous to model a combination of several axis (e.g., a three dimensional vector of accelerometer readings). The three axis readings can be interpreted as vectors. This kind of representation emphasizes the relative strength of each axis.

# **4.3 Investigation of Classification Accuracy**

**Separate the data into training (90%) and test (10%) sets**. Train the system on the training set and test the performance of the system on the test set.

**Create a simple text file (no word .doc file) "myo\_investigation.txt" that** explain which representation (single dimension or multi-axis composition) works better and give a brief explanation for your observations.

If there are special "features" of your program that the marker should be aware of, you should mention them in a simple text file "readme.txt". This may include special features/bugs of your program. **It is in your interest to simplify the job of the marker. Also include information about collaboration with other groups in this file.**

# Submission

**This assignment must be done in group of two or individually (with permission).** You must submit all parts of the assignment before the due date and time via *UMLearn*. Create a zip or tgz archive which includes all source code of your project. Your submission should extract into a directory called COMP4190\_a2<student#\_student#>.  **ONLY ONE SUBMISSION PER GROUP.**

Place a copy of the honesty declaration (modified to include your own name and student number) in the COMP4190\_a2<student#\_student#> directory. Make sure that you understand that doing so and handing it in from your account both securely identifies you and holds you to the responsibilities of a paper honesty declaration ([Group](https://umanitoba.ca/faculties/science/resources/Science_HonestyDeclarationGroup_June2013.pdf)/[Individual](https://umanitoba.ca/faculties/science/resources/Science_HonestyDeclarationIndividual_June2013.pdf)).

**Your program must compile by just running the command make in the source directory or a similar step for your programming language of choice.**

**Remember there is a hard deadline on the due date, late submission will receive penalty of 10% per day.**

**You will also briefly demonstrate (in class) your assignment to the instructor.**