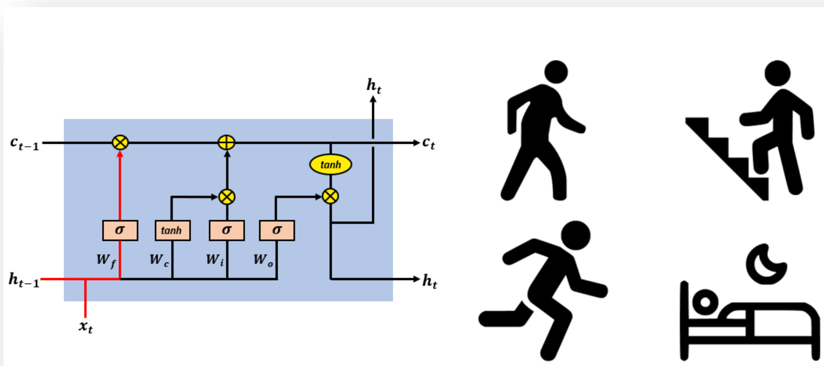
Dr. Eick

COSC 3337 *"Data Science I"* Fall 2022

Last Updated: Oct. 27, 10a

Task 3: Human activity detection using an LSTM network based on the sensor signals collected by a smartphone

Task3 Due: Wednesday, November 9, 11:59p (no extensions will be given this time!)
Responsible TA: Navid
Total points: 50 + (10 Bonus)



Learning objectives:

- Getting familiar with RNN and LSTM.
- Learn how to tune model parameters and hyper parameters using k-fold cross validation.
- Learn how to use majority voting for labeling a test sample based on the outputs of different networks.
- Getting familiar with Bi-directional LSTM.
- (Bonus) Learn how to visualize time-series data using an LSTM encoder-decoder structure.

Classification of time series data is usually a difficult task. Effectively generating features from the raw data for a machine learning model often requires extensive domain understanding and the use of methodologies that are borrowed from signal processing. By automatically extracting features from the unprocessed data, deep learning techniques, like **recurrent neural networks** (RNNs), have recently obtained cutting-edge outcomes for time-series classification tasks. RNN uses sequential data or time series data. Their **memory**, which allows them to use data from earlier inputs to affect the current input and output, sets them apart from other neural networks. Recurrent neural networks' outputs depend on the previous items in the sequence, in contrast to standard deep neural networks' assumption that orders of inputs do not matter.

In this task, we are going to use a variation of RNNs known as Long short-term memory (LSTM) to classify six different human activities based on the data collected by a smartphone. The dataset can be downloaded from:

https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

The data were conducted on a group of 30 volunteers ranging in age from 19 to 48. Each participant used a Samsung Galaxy S II smartphone while engaging in six different activities (WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, and LAYING). We recorded 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using its integrated accelerometer and gyroscope. The experiments were videotaped so that the data could be manually labeled. The resulting dataset was divided into two sets at random, with 30% of the volunteers chosen to create test data and 70% of the participants chosen to create training data. After applying noise filters as a pre-processing step, the accelerometer and gyroscope sensor data were sampled with fixed-width sliding windows of 2.56 seconds and 50% overlap (128 readings/window). The README.txt file contains more information about this dataset.

Download and extract the dataset. In this assignment, we use only **raw data** located in **"Inertial signal"** folder within "train" and "test" folders. The information about the corresponding subject for each row of the dataset can be found in subject_train.txt and subject_test.txt in train and test folders, respectively.

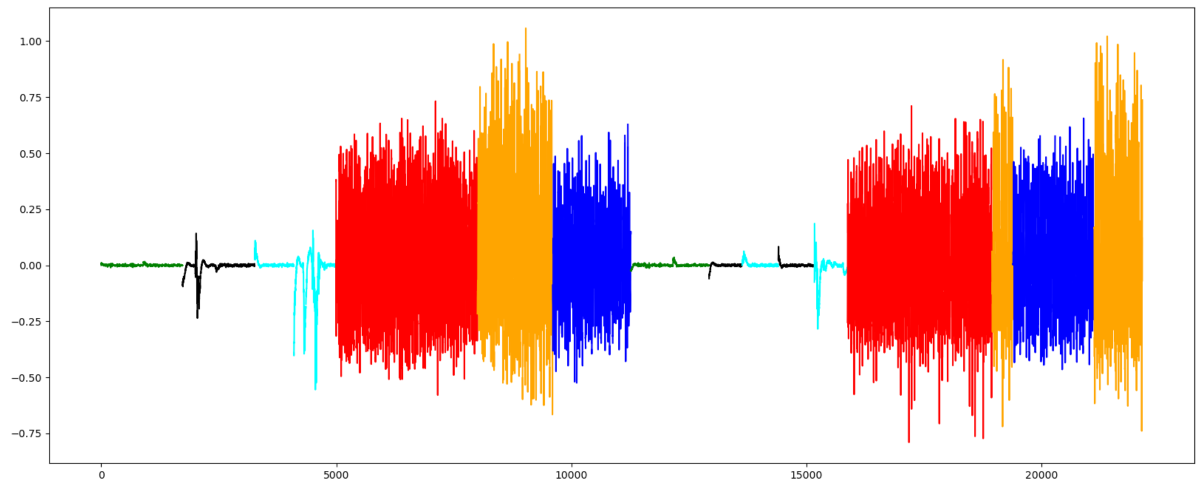
Write your code in **Python** or other language you prefer to answer the tasks listed below. There is a sample Python code for implementing RNN network in the folder of this task in MS Teams.

Tasks:

3.1. Briefly explain (in about one page) how an LSTM cell works. (How many gates does it have and what are the duties of these gates.) **Points: 5**

3.2. Explain (in about one paragraph) the advantage of LSTM networks over traditional RNNs. **Points: 2**

3.3. For subject 1 in training data, plot nine time series corresponding to nine different features. Note that there is a 50% overlap between each row of the data as mentioned above. The color of the line plot should change according to the corresponding activity. Use red, blue, orange, black, green and cyan for WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, and LAYING, respectively. Interpret the obtained plots for discriminating the different activities. The plot for the first feature (body_acc_x_train.txt) is shown in figure below. You need to obtain this for all nine features. **Points: 3**



3.4. Briefly explain k-fold cross validation. **Points: 2**

3.5. Design an RNN network to classify six different activities. Use the input with all nine features. Tune model parameters and hyper parameters and choose one that results the best accuracy based on the 5-fold cross validation. You must use grid search for tuning your parameters (hidden state and number of epochs). You can consider three values (or only two values if you have a slow machine) for each parameter and find the best settings based on the highest accuracy obtained using grid search. Report all confusion matrices, accuracies and recalls for cross validation step. (if you tune learning rate and the number of nodes for the first dense layer as well, you will get bonus points) **Points: 5**

3.6. Train your RNN network with selected settings found in previous task using all training samples (with all nine features) and report the confusion matrix, accuracy and recall. Also, report the confusion matrix, accuracy and recall for testing set. **Points: 3**

3.7. Replace RNN with LSTM structure and repeat 3.5 and 3.6 for designing and training an LSTM network. Compare obtained results. Discuss the reason(s). **Points: 10**

3.8. Now consider the data as nine separate univariate time-series. Design nine separate LSTM networks and feed each of them with the training samples with a single feature (nine features → nine separate LSTM networks). Use majority voting to label each sample in testing set. Report the confusion matrix, accuracy and recall for testing set. Discuss the reasons of the obtained result by comparing it with 3.7. **Points: 6**

3.9. Briefly explain what is a bi-directional LSTM? And why does it sometimes provide better performance over a simple LSTM network? **Points: 3**

3.10. Design and train a bi-directional LSTM. Report all confusion matrices, accuracies and recalls for 5-fold cross validation and testing set. Discuss the results. **Points: 6**

2.11. Write a conclusion or summary (at most 30 sentences!) about what you learned in this task. **Points: 5**

2.12. (BONUS*) In this task, you will learn to use LSTM structure to visualize the time series in a 2-dimensional space. The network consists of one encoder and one decoder. The encoder converts the time series into a vector. The decoder takes this vector as its initialization to produce the input sequence. The network will be trained based on the difference between the original time series and the predicted time series. Design and train an LSTM encoder-decoder. Obtain the encoded time series produced by the trained encoder for testing samples. Plot a supervised scatter plot using the obtained 2-dimensional vectors and the testing labels. Interpret the findings! **Points: 10**

Remark: *Most of the points of the tasks will be given to summarizing and interpreting the obtained results. Therefore, by showing only graphs and tables without any discussions you will get a few points of that task! Your discussions and explanations in your report must be* ***at least 2 pages long****. This excludes any tables and figures (only text!) except for subtask 1. You can include graphical representations of LSTM cell to support your explanations. Also, you can ignore to report all confusion matrices for cross validation step in subtask 3.8. But if you report all, it is going to be considered as bonus points!*

Attention! *Please* ***DO NOT*** *send me your codes for debugging before submission.*

How to submit:

You must separate your codes from your report. The report must include your discussions, explanations and your plots and tables. Do not combine it with your codes! Please upload a ***ZIP*** *file including your report in* ***PDF*** *format and your code files in Blackboard.*