CISC 867: Project 2 (10 points)

Data Preparation

1. In this project, you will use the Fashion-MNIST dataset using a CNN neural network architecture.
   <u>You can use keras/tensorflow or write your own routines</u>.

1) First, download the *data* file, load it, and
    1. Describe the data
    2. Clean the data
    3. Check the data for missing values or duplicates and carry out proper correction methods
    4. Visualize the data using proper visualization methods.
    5. Draw some of the images
    6. Carry out required correlation analysis
2) Carry out any required preprocessing operations on the data
3) Encode the labels

Training a CNN neural network

In this exercise you need to implement a LeNet-5 network to recognize the Fashion-MNIST digits.

- Modify hyperparameters to get to the best performance you can achieve.
- Evaluate the model using 5-fold cross-validation.
- In the report, provide a plot of accuracy improvement using the previously mentioned techniques. Also plot the convergence curve for LeNet-5.
- Comment on why you think LeNet-5 further improves the accuracy if any at all. And if it doesn't, why not?
- Note that this is an open-ended problem. So, there could be very good solutions.
- Try to use other two CNN models (using transfer learning) and compare the results with the full trained LeNet-5.

Use Google's Colab to save time.

Submission: Please include the following files (project_2.zip):

- project_2.pdf/docx and project_2.py/ipynb

CISC 867: Bonus Project 3 (10 points)

Building a recurrent language model

A language model can predict the probability of the next word in the sequence, based on the words already observed in the sequence. In this problem, you will need to prepare text for developing a word-based language model, design and fit a neural language model with a learned embedding and a recurrent hidden layer, and to use the learned language model to generate new text with similar statistical properties as the source text.

You will use *The Republic by Plato* as the source text. A cleaned version (no hyphens or punctuations, removed nonalphabetic words, and all words in lowercase) can be downloaded from https://www.gutenberg.org/cache/epub/1497/pg1497.txt. The file should be about 15,802 lines of text. Now we can develop a language model from this text. You need to:

1. Organize the text into sequences each of 50 input words and 1 output word. You will have exactly 118,633 training patterns to fit our model.
2. **T**rain a statistical language model using a recurrent architecture from the prepared data that
    a. uses a distributed representation for words so that different words with similar meanings will have a similar representation.
    b. learns the representation at the same time as learning the model.
    c. learns to predict the probability for the next word using the context of the last 100 words.
3. Try different types of recurrent nodes (GRU, LSTM) and different number of nodes/layers and report how this has affected the results

Submission: Please include the following files in the archive: (project_3.zip):

- Report_3.pdf/docx
- Project_3.py/ipynb