**EIE4122 Deep Learning and Deep Neural Networks**

**Lab 2: RNN and LSTM for Text Analysis**

**A. Objectives and Outcomes**

After finishing this lab, you should be able to perform the following

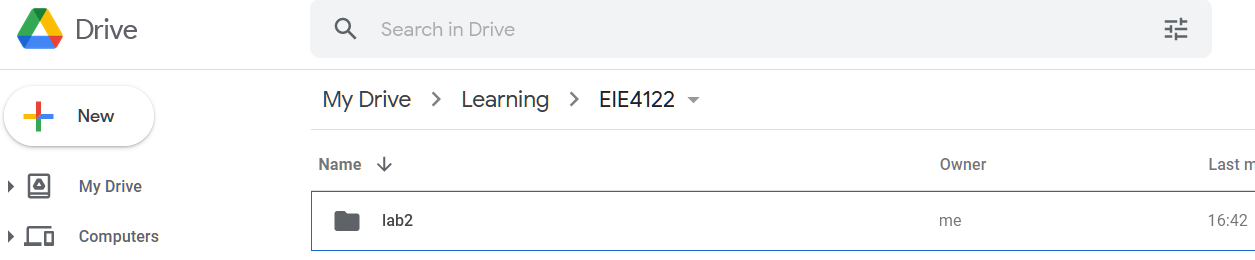
* Use PyTorch to develop recurrent neural network (RNN) for sentiment classification.
* Preprocess datasets for sentiment classification, learn how to split training materials, understand the concept of random seed, vocabulary and embeddings.
* Understand the architecture of RNN, LSTM, bidirectional-LSTM and multi-layer LSTM.
* Understand padding and packing sequences.

***B. Prepare Colab Environment***

1. Colab runs on browsers. You need a Google account to use Colab. If you do not have one, visit https://support.google.com/mail/answer/56256?hl=en.
2. Display the Google Drive (https://drive.google.com/drive/) page in your browser. Create the following directory structure in your Google Drive:

My Drive/Learning/EIE4122/lab2/

After creating the folders, you should see something like this:



Directory structure in Google Drive

***C.1 Conduct two experiments for sentiment classification***

1. Download ‘RNN\_SentimentAnalysis.ipynb’ from

<https://github.com/enmwmak/EIE4122> and drag ‘RNN\_SentimentAnalysis.ipynb’ in your computer to

“My Drive/Learning/EIE4122/lab1/” in your Google Drive.

Right click ‘RNN\_SentimentAnalysis.ipynb’ and open it with Colab lab, read the materials in the .ipynb file and follow the procedures in the file to conduct the experiment for sentiment classification using a basic RNN network.

1. Download ‘LSTM\_SentimentAnalysis.ipynb’ from

<https://github.com/enmwmak/EIE4122> and drag ‘LSTM\_SentimentAnalysis.ipynb’ in your computer to

“My Drive/Learning/EIE4122/lab1/” in your Google Drive.

Right click ‘LSTM\_SentimentAnalysis.ipynb’ and open it with Colab lab, read the materials in the .ipynb file and follow the procedures in the file to conduct the experiment for sentiment classification using a upgraded RNN (Mult-layer bidirectional LSTM).

5. When you conduct the experiments, remember capturing the outputs and results to your reports.

**References:**

[1] <https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>

[2]<https://github.com/bentrevett/pytorch-sentiment-analysis/blob/master/1%20-%20Simple%20Sentiment%20Analysis.ipynb>

[3]<https://github.com/bentrevett/pytorch-sentiment-analysis/blob/master/1%20-%20Simple%20Sentiment%20Analysis.ipynb>

Hints:

(1) There are two embedding dimensions in this lab: word-embedding dimension and embedding dimension. The former refers to the dimension of the 1-hot-encoding of each word. If you have 1000 words in your vocabulary, the word-embedding (also named INPUT\_DIM and vocab\_size in the program) dimension is 1000. This dimension can be changed by setting the maximum number of vocabulary (MAX\_VOCAB\_SIZE). The latter is the dimension of the dense vectors to be input to the RNN layer, which is set to 100 by default (EMBEDDING\_DIM = 100).

(2) The RNN layers can be stacked by using the code as follows:

In init(),

self.rnn1 = nn.RNN(embedding\_dim, hidden\_dim1)

self.rnn2 = nn.RNN(hidden\_dim1, hidden\_dim2)

Then, in forward()

output1, h1 = self.rnn1(embedding)

output2, h2 = self.rnn2(output1)

-- END --