

## MA 721: Project 1 (due Friday Dec 8, 2017)

Write a Keras/Theano/Tensorflow program to implement an MLP (convolutional layers allowed) and an RNN (LSTM allowed) for the IMDB dataset listed below. Train your MLP and RNN and present your best results in a table that includes the minibatch size, the number of hidden layers, the numbers of hidden units given as a listing, the total number of parameters, the optimization methods used, the learning rate, the initializer, the regularization if used, the number of training epochs, the final training accuracy, the final testing accuracy, and **the ratio of the test accuracy to the 10th root of the total number of parameters**. This ratio as well as **the test accuracy** will be used to score your results. Also, plot the convergence curve of the loss function, the training accuracy and the testing accuracy against the training epochs.

IMDB Movie reviews sentiment classification dataset:

(see <http://ai.stanford.edu/~amaas/data/sentiment/>)

1. In Keras, download the dataset using these options with `vocab_size=20000` and `maxLen=250`  

```
(x_train, y_train), (x_test, y_test) = imdb.load_data(path="imdb.npz",  
num_words=vocab_size, skip_top=0, maxlen=maxLen, seed=113, start_char=1,  
oov_char=2, index_from=3)
```
2. Pad the sequences to the same length of `maxLen=250` using  

```
x_train = sequence.pad_sequences(x_train, maxlen=maxLen)  
x_test = sequence.pad_sequences(x_test, maxlen=maxLen)
```
3. Use word embedding to represent words in a dense vector representation; see <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/> for an explanation. In Keras, this is implemented by using an embedding layer as the first layer for inputs:  

```
model.add(Embedding(vocab_size, 128, input_length=maxLen))
```

### Notes:

1. Submit your codes and results through Canvas in a single PDF file.
2. Submission received up to 3 days late will receive 50% credits. Submission of more than three days late will not be accepted .