INTRODUCTION:

Text Classification using CNN, RNN and LSTM

OBJECTIVE:

To implement RNN, LSTM and CNN for text classification – sentiment

Prediction will be like :

“I like it very much”

“It is so interesting”

“I don’t like it”

“It isn’t interesting”

DATASET:

IMDB dataset and then process it accordingly as you can see in “imdb_corpus_data_processing.py “where I took max length of corpus as 200.

APPROACH:

Implemented RNN, LSTM and CNN. All models accept pre-trained word embedding inputs. Used corpus.py to build corpus.

PARAMETERS:

N_hidden = 128, n_emb=128. Batch _size = 32. Conv_size =5

Pooling type ='mean' or 'max', Model_type='lstm,rnn,cnn', w2v_fn=None,

Model_save_fn=None, disp_proc = True

Use_w2v : to use pre- trained embeddings from word2vec

EVALUATION AND DISCUSSION:

We see that :

a) For max – pooling
Accuracy for RNN =0.9038 , CNN =0.8765 and LSTM = 0.8993
b) For Mean – Pooling :
Accuracy for CNN = 0.8883, RNN=0.8765 AND LSTM=0.8345

RNN is better for max-pooling and CNN is better for Mean - Pooling