TEK5040 Assignment, Text sequences

Narada Warakagoda

October 12, 2020

1 Introduction

This assignment deals with word prediction using Recurrent Neural Networks (RNN).

One Python/Tensorflow script `wordpred.py` has been provided.

The data set is provided in three *pickle* files: `lm_corpus.pkl`, `lm_dict.pkl` and `lmr_embeddings.pkl`. `lm_corpus.pkl` contains a set of words stored in a python list, `lm_dict.pkl` contains a python dictionary which maps each word into an index and `lmr_embeddings.pkl` contains a 2D python list where each row gives the embedding vector of size 5 corresponding to the word of that row index.

`wordpred.py` trains a word prediction system based on RNNs. More specifically it uses the current N words taken from the training set to predict the next word. Default N is 4 and it is called `time_steps` in the script (see line 66). There are two ways to represent the words: just use the word index or word vectors computed and stored in `lm_embeddings.pkl`. Note that dimensions of the word vectors are reduced to 5 using Principle Component Analysis (PCA) in order to reduce the computational burden.

2 Task

1. Run the script `wordpred.py` as it is. This version makes use of word indexes (i.e. scalar values) for representing words. Study how the loss evolves by examining the program output. Change `input_fea_dim` (line 67) to `embedding_dim` (line 62). Run the script again. How the loss function evolves this time compared to the previous run? Give a possible explanation.

2. In the provided version of `wordpred.py`, a single LSTM layer is used (line 127). Implement a version with two LSTM layers. Run the new version and observe the evolution of the loss function in training and comment!

3. In the provided version of `wordpred.py`, we predict the next word using the current N words (i.e. predict the word $N+1$ using the words $1, 2, \cdots, N$). But we can predict all

the words $(2, 3, \cdots, N+1)$ from the current words $(1, 2, \cdots, N)$ along the way. Modify the script to handle that situation.

4. How can we convert the model above to a bidirectional LSTM?