# Homework 09 – Language

Arthur J. Redfern
arthur.redfern@utdallas.edu

---

# 0 Outline



# 1 Reading

1. Language
   Motivation: understand common uses of xNNs in language applications
   https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Lectures/xNNs_090_Language.pdf

   Complete

2. Open sourcing BERT: state-of-the-art pre-training for natural language processing
   Motivation: an alternative presentation of BERT
   https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

   Complete

3. The illustrated BERT, ELMo, and co. (how NLP cracked transfer learning)
   Motivation: an alternative presentation of language modeling
   http://jalammar.github.io/illustrated-bert/

   Complete

# 2 Theory

None

# 3 Practice

4. Word embeddings. Understand the following example (https://www.tensorflow.org/tutorials/text/word_embeddings) and run it in Google Colab. It trains an embedding layer using word2vec style training.

Complete

5. [Optional] Sentence embeddings. BERT is available as a pre trained model in TensorFlow Hub. Using a similar strategy as in a previous homework, use the pre trained BERT model for sentence embedding with a shallow decoder trained for a specific application. A quick online search should give you some ideas. Google's TensorFlow BERT code can be found here: https://github.com/tensorflow/models/tree/master/official/nlp/bert

Complete

6. Language modeling. Understand the following example (https://www.tensorflow.org/tutorials/text/text_generation) and run it in Google Colab. It uses a RNN to predict the next character and recursively calls it to generate text.

Complete

7. Language translation. Understand the following example (https://www.tensorflow.org/tutorials/text/nmt_with_attention) and run it in Google Colab. It uses an encoder – attention – decoder style network with RNNs for the encoder and decoder.

Complete

8. Language translation. Understand the following example (https://www.tensorflow.org/tutorials/text/transformer) and run it in Google Colab. It uses an encoder – attention – decoder style network with self attention for the encoder and decoder.

Complete