Fake News!

Deep Learning and Reinforcement Learning: Final Project

Chosen dataset

A Kaggle competition dataset of text articles each labelled as fake or true. In the current social media climate news can reach people quicker than it can be validated. Misinformation distorts people's view on reality en masse resulting in large public movements, e.g. anti-vaxxers or flat earthers, that can do a lot of harm to society. The models build with this dataset try to fight misinformation by identifying articles as fake or true.

Link to dataset: https://www.kaggle.com/c/fake-news

Data preprocessing

One needs to *tokenize* the text data before passing to the model. *Tokenizing* a text is splitting the text into a list of words and converting each word into a unique number according to some word to number mapping. Here we used the *spaCy tokenizer* with the *en_core_web_sm* dictionary. One defines a maximum number of words taken from the dictionary resulting in one hot vectors with a size of that maximum plus two additional numbers corresponding to a word outside this set of words and padding. Padding are numbers used to pad text sequences to make them equally long.

Models

Three different binary text classification models have been trained and validated. The code of each model is implemented in a Jupyter notebook, made available on Github and can be run on Google Colab:

1. Simple RNN
2. LSTM
3. Fast Tex

Link to Github: https://github.com/pimverschuuren/FakeNews/

The Simple RNN and LSTM are both a recurrent neural networks that take the tokenized text as input. Recurrent neural networks learn from the word passed to the network but also take some form of the previous word as input, i.e. memorize the previously passed words. This allows the neural network to learn from one or multiple sentences. The FastTex model uses filters like in convolution neural networks to combine multiple words into one output and learn from that.

Performance

Simple RNN

The loss function does not diminish and the validation accuracy remains 50% i.e. the model has no classification power. Even after training the model for multiple epochs.

LSTM

After 5 epochs of training the final validation accuracy is 99.48%. This is a substantial improvement compared to the RNN. One needs at least 10 mins of training time on Google Colab GPUs.

FastTex

After 5 epochs of training the final validation accuracy is 93.79%. This is slightly lower than the LSTM. However, 5 epochs of training took 2.5 mins i.e. was four times faster to train than the LSTM.