TDT4171 - Exercise 5

Jens Waage

Report

Parameters and accuracies

Naive Bayes and Decision Tree

For the first part of the assignment the bag-of-words was performed using the following parameters for the HashingVectorizer:

- `stop_words = 'english'` -> basic dictionary of common english stop words. The dataset is all english reviews, so I figured this should suffice for the desired performance.
- `n_features = 2**18`. Recommended[1] number of features to avoid hashing collisions.
- `binary = True`. We want to know whether a word is present or not, so any word with a count >0 should be 1.

In addition, the decision tree was tweaked by changing default parameters `min_samples_leaf` to 10 and `min_impurity_decrease` to 0.01. This was done to ensure the model would not run for too long.

Accuracies for the Naive Bayes and Decision Tree classifiers were as follows:

```
Naive Bayes accuracy: 0.8476572076489336
Decision Tree accuracy: 0.7982655062515323
```

Keras LSTM

For the embedding layer the data was padded using the `max_len` parameter. The model was built with an `Embedding` layer with an output size of 10, an LSTM with 3 units and a Dense-layer with 1 unit and a tanh activation function.

The model uses mean-squared-error as its loss function and RMS-prop for gradient descent. It was trained for 1 epoch only due to time constraints.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html

The accuracy of the LSTM was as follows:

```
130272/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130304/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130336/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130368/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130400/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130432/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130464/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130496/130528 [============================>.] - ETA: 0s - loss: 0.0680 - accuracy: 0.9097
130528/130528 [==============================] - 264s 2ms/sample - loss: 0.0680 - accuracy: 0.9097
Accuracy: 0.9096899032592773
```

LSTM vs Naive Bayes/Decision Tree

The Naive Bayes and Decision Tree algorithms classify reviews as positive or negative based only on the presence of certain words. It does not take into account where those words occur in conjunction with other words in the review, i.e. the context. This could lead to errors and dilution of the classification through sentences such as “I like the food” and “I didn’t like the food” both containing the word ‘like’, but expressing two different opinions. With the LSTM and word embedding we can capture this context, and allow the network to train on the review as a whole to a much larger degree. In the example given, the LSTM would more easily capture the second review as negative because it would classify using the whole sentence, not just the individual words.