

Projekt lab notes

2016

Project

Things we plan to do

19 december 2016

We have to day decided to try two approaches to the classification problem. One: we use the states given in the training data to create two markow models, one for the positive data and one for the negative data. We the use this model to calculate the probability of the sequence we want to classify, if it is more likely with the positive model, we classify it as positive, else it is negative. Two: we are also going to try to train a neural network for the classification problem. We are going to do this using an RNN(recurrent neural network) with LSTM(long short-term memory) units. In the first step we are going to disregard the data we have regarding the underlying statets and only look at the binary classification problem. In a later stage we might try to train an ANN with the hidenstate data as target.

```
model = Sequential()
model.add(Embedding(27, 24, input_length=max_len))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
model.fit(X_train_padded, Y_train, nb_epoch=3, batch_size=64)
# Final evaluation of the model
scores = model.evaluate(X_test_padded, Y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

Figure 1: Code for the first run.

This model was trained without dropout and with an embedding layer that was probably unnecesary.

```
Epoch 1/3
2388/2388 [=====] - 2725s - loss: 0.6689 - acc: 0.5917
Epoch 2/3
2388/2388 [=====] - 2827s - loss: 0.6630 - acc: 0.6591
Epoch 3/3
2388/2388 [=====] - 2889s - loss: 0.6408 - acc: 0.6642
Accuracy: 66.17%
```

Figure 2: Stats from the first run
