Comparison of Deep vs Shallow Learning using IMDB Review Classification

Sean Northcutt, Johnnie Oldfield, Andrew Russell

November 27, 2019

1 Problem

Talk is always being done on what methods are better and faster for a researcher to use when it comes to applying machine learning to some data that they have. This project consists of a comparison of a deep learning solution and a shallow learning solution to the same problem. That problem is the classification of IMDB movie reviews based on whether they are positive or negative. The models we will be using for shallow learning is a basic Naive Bayes model and for our deep learning portion the classification problem will be applied to a Keras deep learning model using 2 epochs.

2 Data Set: Source

The data-set used contains 50,000 movie reviews obtained from IMDB and imported using Keras. This data-set was already pre split into a training and testing set with a 50/50 split and we couldn't change that to a more preferred 80/20 split. Each word in a review is indexed by overall frequency in the data-set so each review is a sequence of integers where the integer '20' is considered to be the 20th most frequent word in the data-set and is done like that for convenience. We proceeded tog get those indexes and turn the reviews back into normal reviews so that it would work with our shallow learning Naive Bayes method. The reviews looked like what is seen below where BEGIN is the beginning of the review UNKNOWN and UNUSED are unused words in the data-set that couldn't be indexed.

```
'BEGIN rarely have i seen an action suspense movie that was so boring ...
   none of the action scenes are exciting the story line is nothing ...
   special and except for a couple of actors the acting is bad charlie ...
   sheen platoon major league stars as the white house chief of staff ...
   who gets himself in the middle of a conspiracy that wants him and ...
   more people dead donald sutherland a time to kill fallen plays a ...
   friend who he tells everything he can and linda hamilton linda ...
   hamilton terminator UNKNOWN peak plays a reporter who gets involved ...
   in the situation charlie sheen is ok as the star donald sutherland ...
   is a good actor who gives a good performance linda hamilton gives a ...
   poor performance with a bad movie i can actually like it if it has a ...
   good ending but this movie has a very cheesy ending with some almost ...
   laughable stuff'
```

This is what the same review looked like with the original indexed integers that keras provided us with.

```
'[1,1713,28,13,110, ... ,19,49,220,1322,538]'
```

3 Naive Bayes Method

```
constant = np.log(np.sum(np.where(np.array(y_train)==0))/
    np.sum(np.where(np.array(y_train)==1)))

pofx = {}

for(ix, review) in enumerate(reviews):
  words = cleanup(review)
  for word in words:
    if word not in pofx:
      counts = [0, 0]
      counts[y_train[ix]] = 1
      pofx[word] = counts
    else:
      pofx[word][y_train[ix]] += 1

pofx = {i:[pofx[i][0]/np.sum(pofx[i])+ 0.0000001 , ...
   pofx[i][1]/np.sum(pofx[i])+ 0.0000001 ]for i in pofx}

tables = [[i, np.log(pofx[i][0]/pofx[i][1])] for i in pofx]
tabs = {i[0]:i[1] for i in tables}
```

The code above is our implementation on the training portion of the Naive Bayes Method. What this is doing is taking all the words from each of the 25,000 training reviews and placing them in a dictionary where each word has a count of its occurrences associated with that word. We will then take each word and get the probability that it occurs in negative reviews and the probability that it occurs in positive reviews and run our tests with those probabilities. The testing portion was relatively simple where we would add the constant to a ratio and then the probability for each word in a review ignoring ones that did not appear in the training set as it would not affect the decision made by the algorithm. If the ratio was less than 0 it was branded a positive review and if it was greater than 0 it would be the opposite.

3.1 What is Naive Bayes

The Naive Bayes Classifier that we used on our shallow learning method is a probabilistic classifier that applies Bayes Theorem assuming that all the features, in the case of the problem used for our comparison the features were all the words in a review and our classes were positive and negative reviews, are independent. The formula for this is [1]:

$$p(C_K|X) = \frac{p(C_k)p(x|C_k)}{p(x)}$$

where X is a vector and this is for the basic Bayesian formula before we classify it. When classifying the formula evolves into

$$ln(p(C_k|x)) \propto ln(p(C_k)) + \sum_{i=1}^{n} p(x_i|C_k)$$

In laymans terms to figure out what class an unknown is apart of you take the largest of the probabilities and assign the unknown to that class.

4 Keras Deep Learning Method

```
max_words = 500
X_train = sequence.pad_sequences(x_train, maxlen=max_words)
X_test = sequence.pad_sequences(x_test, maxlen=max_words)

model = Sequential()
model.add(Embedding(top_words, 32, input_length=max_words))
model.add(Flatten())
model.add(Dense(250, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', ...
    metrics=['accuracy'])
model.summary()
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=2, ...
    batch_size=128, verbose=2)

scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

The above code implements the use of Keras in using neural networks with different layers to help realize the patterns to help predict whether or not reviews are positive or negative. Our model uses binary cross entropy since we are going with determining if a review is positive or negative so the computer reads it as 0 or 1 for it's decision making.

4.1 What is Keras?

Keras is a high-level API that has been adopted not only by Tensorflow but other back-end engines as well. The python library as three main focuses. Being modular, easy to extend, and user friendly. The latter is the reason it's so popular. Learning Keras is not very difficult because of it's ease of use and building models is rather intuitive. Although, the simplicity of Keras does not take away it's effectiveness. Simple models are very useful for quick prototyping but there are numerous features in Keras to expand models [2]

5 Evaluation

Our evaluation is done three ways in our comparison of the learning methods used for our data-set. Time, Accuracy, and a confusion matrix is what was decided best for the comparison of our models. What we found was interesting in our tests done for this project. While the confusion matrices were very similar as seen in figures 1 and 2 with the Keras model having a better time identifying

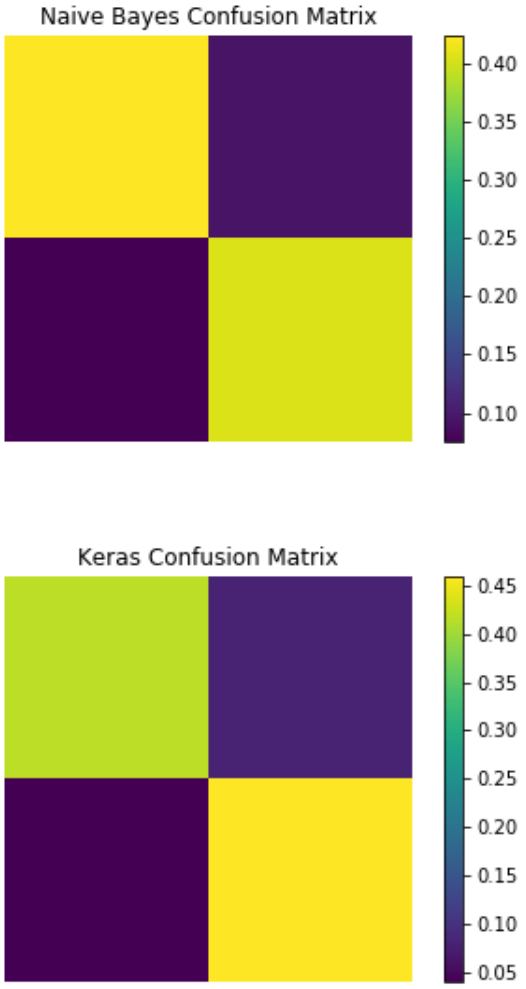

Figure 1: Confusion Matrices for both methods you can see how the Keras method has a better time identifying negative reviews and opposite for the Bayes method

negative reviews than positive and the opposite for our Naive model. Accuracy wise our methods were again similar but the Deep Learning Keras model was higher with an average of 88 percent and the Naive Bayes had an average accuracy of 82 to 84 percent. The most interesting metric for our evaluation was time and since time is always the most important factor in determining what model to use especially when the metrics for both models are pretty similar in accuracy and our confusion matrices. With time we found that the Naive Bayes model was the fastest with an average time of 20 seconds per run and with our Keras model we found that time was over double with an average time around 50 seconds, but that is because we are running 2 epochs because 1 epoch takes about 23 seconds on average to run. As a result making our time double for accuracy that is not too dissimilar to the Naive Bayes method.

6 Results

After evaluating both methods through multiple tests it is best to recommend using the shallow learning Naive Bayes method. This is solely because of time as with the Keras model applied to the same problem the time to solve the classification problem doubled to get a result that was just 4 percent better than the Naive Bayes method. Though if time was not an issue for you then the best solution would be to use the Keras Deep Learning Method but we do not recommend it for such a marginal accuracy difference.

7 Discussion

As we assumed, the deep learning approach did perform better in terms of accuracy as neural networks are very effective if implemented properly. We were unsure which method would perform better regarding time but hypothesised that the sequential model would take more time. We did not expect how significantly longer it would take. This could be attributed to the the fact that Keras' sequential model does not allow strings to be taken in as data, So the model had to have an embedding layer to go through every single string in the data set.

8 Conclusion

While this problem lead to some surprising results it did teach us a bit about the mechanics behind deep learning. Even though deep learning is slower using libraries like Keras makes the implementation of deep learning a lot easier to follow and understand than with figuring out the implementation of shallow learning models like Naive Bayes. This project also led us to see results that proved that even though deep learning is better with accuracy it is not always the best way as when looking at the time between the implementation of both methods to achieve similar accuracy the shallow learning model is 2 times as fast as the deep learning model. Though that was with a small data-set of 50,000 reviews these tests would best be performed with larger data-sets and ones where there are more classes too see how each method deals with significantly more complex problems than a simple polarity separator on reviews.

References

[1] https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[2] https://en.wikipedia.org/wiki/Keras.