Machine Learning Engineer Nanodegree

Capstone Proposal

Sujay Bhowmick

August 8th 2018

Proposal

Domain Background

In NLP Sentiment Analysis is a process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative or neutral.

There are three main approaches to Sentiment Analysis

1. Lexicon based - considers lexicon dictionary for identifying polarity of the text
2. Machine learning based approach - Needs to develop classification model, which is trained using prelabeled dataset of positive, negative, neutral content
3. Combined approach - Which uses lexicon dictionary along with pre-labelled data set for developing classification model (Combien approach 1 and approach 2)

My project will be based on approach 2 where I have classification model which is trained using prelabeled dataset of positive and negative tweets.

Sentiment Analysis is useful in many ways. In my use case we are classifying Tweets of some relevance to domain like Finance related to both Micro and Macro events for which various credible financial analyst, activist, famous investors and financial news publishers are talking about in Twitter through their twitter handles (can not provide the Twitter handles of the users here due to privacy concerns). These Tweets are then used to guage the sentiment of the investors on certain topics and can be used to assess the investment decisions on that particular financial asset or stock after sentiments are combined with other analytical data dervided using different methodologies.

My motivation for this project has come from the fact that I have been working in a company which is working on Sentiment Analysis of News, Blogs and Tweets. I am doing this project to work with the data science team to understand and deploy the machine learning models at scale.

Problem Statement

The goal of this Machine Learning Nanodegree Capstone project is to analyse the sentiment of various twitter tweets which is publicly available.

The tweets are related to financial news which have been labelled by a human for training and testing purpose of the Machine Learning model. There are approximately 8000 tweets which have been labelled with labels **Positive** and **Negative** for tweets indicating a **Positive** sentiment and **Negative** sentiment respectively.

Data Sets and Inputs

I have prepared a list of approximately 8000 Twitter tweets and labelled them accordingly with **Positive** and **Negative** labels.

The training and test dataset contains *total 8351* labelled Tweets, with *3843 positive* and *4508 negative* tweets. The input data feature in this case is the Twitter Tweet content and I will try to predict the Sentiment of the Tweet content.

Link for the labelled dataset can be found here

Note: Neutral sentiments are not included in the labelled data.

> **References:**
>
> https://machinelearningmastery.com/quick-and-dirty-data-analysis-for-your-machine-learning-problem/ https://www.r-bloggers.com/how-to-prepare-and-apply-machine-learning-to-your-dataset/

Solution Statement

I have decided to use Convolutional Neural Network (CNN) classifier to predict the sentiment (positive or negative) of a tweet

Benchmark Model

I am using the Afinn word list based approach of Twitter sentiment analysis which is currently in use in my company as my benchmark to compare my CNN based model.

How Afinn model works

1. Methodology is keyword-matching,
2. Dictionaries of keywords and their Sentiment Value are pre-defined,
3. Input message is split by all non-alphanumeric characters into individual Tokens,
4. Each token is matched against the dictionary in the appropriate language,
5. Afinn's Sentiment Score is the sum of all Sentiment Values of the matched Tokens in the input message.

Afinn's weakness

1. Dictionary is more suited for analyzing product reviews
2. Methodology cannot reliably deal with even slightly complex language patterns (e.g. "not good")

The original form factor from Afinn model is an integer-value formula as Afinn returns the sum of values of all tokens in a message. Based on a small simulated test, this form factor made it more difficult to translate from Sentiment Score to Sentiment Label as the results may vary infinitely.

> **References:**
>
> http://www.stratio.com/blog/benchmarking-machine-learning-prediction-models/ https://blog.dominodatalab.com/benchmarking-predictive-models/

Evaluation Metrics

Once you have built your model, the most important question that arises is how good is your model? So, evaluating your model is the most important task in the data science project which delineates how good your predictions are.

I will use a confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. In the confusion matrix there are four parameters which needs to be looked at to determine if the model is good enough. These are

1. **True Positives [TP]** - These are the correctly predicted positive values, which means that the value of actual class and the value of predicted class is same
2. **True Negatives [TN]** - These are the correctly predicted negative values which means that the value of actual class and the value of predicted class is same
3. **False Positives [FP]** - These values occurs when value of predicted class is positive and actual class is negative
4. **False Negatives [FN]** - These values occurs when value of predicted class is negative and actual class is positive

Once you understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all Tweets that are labeled as Positive for sentiment, how many actually Positive? High precision relates to the low false positive rate

Recall - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - Positive. The question recall answers is: Of all the Tweets that have a Positive sentiment, how many did we label?

F1-Score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

> **Pro Tips:**
>
> https://towardsdatascience.com/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428 https://towardsdatascience.com/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

Project Design

Data Collection, Labelling & Preprocessing

First step is to collect the data i.e. Twitter Tweets and label them with positive and negative labels based on the sentiment of the tweet. Once I have the labeled data set, I will encode the labels to 0 for negative and 1 for positive sentiment respectively. Next step is to preprocess the tweet content and remove punctuations, links (with tag), tweet handles(with), hash tag (with) and certain special characters.

Feature Extraction

Once we have the labelled data with Content and Label as columns in the data set. I looked at the distribution of the words in the Tweet content. For this I have used a custom tokenizer using Keras text preprocessing tokenizer and observe the distribution of words. We can then determine the maximum number of tokens in the training dataset. This is a good input feature which can be used for building the classifier.

Splitting data

I will split the tweet dataset into training and test datasets using sklearn's split function. I going to do 80 to 20 split for training and test data set respectively

Model

I am choosing a 1-D Convolutional Neural Network (CNN) using Keras and Tensorflow to build and train the model

> **References**:
>
> Afinn - https://github.com/fnielsen/afinn
>
> Another Twitter sentiment analysis with Python - https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-11-cnn-word2vec-41f5e28eda74
>
> Sentiment Analysis Using Convolutional Neural Network - https://ieeexplore.ieee.org/document/7363395/