Machine Learning Engineer Nanodegree

Capstone Proposal

Nishkarsh Vardhan
May 4th, 2018

Proposal

Domain Background

Proposal for the project is based on Toxic Comments featured in feeds, tweets and comments on various social pages. Natural Language Processing and Recurring Neural Network are one of the most powerful domains of Machine Learning and are linked in many ways. Recently, it has been observed a decline in people expressing themselves due to continuous threat of abuse and harassment they face online. Seeking opinions and raising their voices over any discussion are becoming difficult day by day due to toxic comments. People are getting bullied with the toxic remarks. Existing platforms struggle to effectively facilitate conversations. If we can solve the problem for toxic comments using Machine Learning techniques it will be easy for people to express themselves without any reluctance. More opinions will lead to more ideas and more developments. Recently, many companies worked on similar projects and provided APIs like Perspective API from Google etc. But current models still make errors and they are not of much help. An attempt will be made on Wikipedia comment's data using Machine Learning Techniques . Personally, motivation behind the project is to let public speak and come out on major issues. More number of people will open up so we get to know real life issues and that's how other problems can also be dealt very easily.

Problem Statement

Toxic Comment Classification is the major goal for this project. I will use Deep Learning and Neural Networks to train the model and help to detect accurateness of toxic comments detection. At start I will start working with basic models to check if we are getting correct output then will tune the model in more elaborate manner with deep learning. This Problem is frequent nowadays and can be easily worked upon as it is quantifiable.

Datasets and Inputs

This project will use dataset from Kaggle available at
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.

This is a data collected by Google and Jigsaw, it's a data with large number of Wikipedia comments which have been labeled by human raters for toxic behavior.

Data set comprises of 2 files, these are:-
train.csv - the training set, contains comments with their binary labels

test.csv - the test set, you must predict the toxicity probabilities for these comments.

To deter hand labeling, the test set contains some comments which are not included in scoring.

Solution Statement

A deep learning algorithm will be developed using TensorFlow/Keras and with the training data it will be trained accordingly. A good start of the project can be done using basic Machine learning algorithms like Linear Regression or Naive Bayes then more specifically will focus on deep learning algorithms like Recurrent Neural Network basically LSTM (Long Short Term Memory) model and further optimization can be done accordingly. Predictions will be made on test data set and proper evaluation will be done accordingly.

A good reference for model is a paper on RNN and LSTM is Comment Abuse Classification with Deep Learning by Theodora Chu, Kylie Jue kyliej,Max Wang.

Benchmark Model

As a baseline model I am considering is linear regression and will also run the feed forward neural network as deep learning baseline.
Convolutional Neural Networks for Toxic Comment Classification by Spiros V. Georgakopoulos reported accuracy of approximately 90% compared to previous models like Linear Regression,SVM,kNN etc which were in the range of 60-80% accuracy. Ex Machina: Personal Attacks Seen at Scale by Google,Jigsaw and WikiMedia are using there existing methods like Linear Regression which are not up to the mark to clearly define the accuracy.

Main motivation will be to work around the accuracy and to create a good model at-least comparable to CNN's accuracy. So, using RNN based LSTM model will try to enhance the model and reach the accuracy which can cover CNN as well as Basic models in depth.

Google Jigsaw published a paper on using machine learning to automatically detect abusive comments. Titled ”Ex Machina: Personal Attacks Seen at Scale,” the paper detailed some of the algorithms behind Google’s recently released Perspective API, an API aimed at ”hosting better conversations.” In their paper, the researchers tested logistic regression and multi-layer perceptrons for their model, word and character n-grams for their features, and one-hot vectors and empirical distributions for their labels. They found that a multi-layer perceptrons (MLP) model with character n-grams and empirical distribution labels had the highest standard 2-class area under the receiver operating characteristic curve (AUC) score and an F1 score of 0.63.

I will try to enhance my model from Linear Regression to Recurrent Neural Network based LSTM model.

Evaluation Metrics

I will be using two baselines in order to get a better understanding of the problem from both a general machine-learning perspective and a deep-learning perspective. I will use linear regression with least squares loss function in order to provide the general machine-learning baseline.

Square lost function described as :

$$J(f(x);y) \text{ square} = \Sigma i*(yi - f(xi))^2$$

For deep-learning perspective will be using feed forward neural network classifier to provide a baseline. TensorFlow has DNN classifier built into its libraries, and its default loss function is logistic loss:

$$J(f(x);y) \log = \log[\ 1+\exp(-y*f(x))]$$

For our primary models, we used a recurrent neural network (RNN) with a long short-term memory (LSTM) cell and a convolutional neural network (CNN). I will use word based embedding and will test for word-level embedding on the LSTM for efficiency reasons.

Recurrent Neural Network with Long Short-Term Memory cell LSTMs in particular outperform RNNs with GRU cells on problems where understanding a broader, long-term context is important, and the LSTM cell also prevents the problem of a vanishing gradient. LSTMs are commonly used in sentiment classification because of their ability to use longer-term contexts. LSTMs uses sigmoid functions for the classification.

Following formulas based on RNN:-

$a(t) = b + Ws(t-1) + Ux(t)$
$s(t) = \tanh(a(t))$ (summary of past sequence of inputs upto t)
$o(t) = c + Vs(t)$
$p(t) = \text{softmax}(o(t))$

Based on above parameters and functions will define LSTM cells as well.

Project Design

First Stage of the project will be to download and preprocess the data provided from wikipedia comments. Comments might contain multiple acronyms, emoticons and unnecessary data like urls, images etc. So we need to first preprocess the data to represent correct emotions of public. Now, here I will do filtering like using tokens (spliting individual words based on space and symbols) for list of words. Remove unnecessary words which do not show any emotions like a, is, the etc. Remove unnecessary URLs or searching for words in special symbol based words like to identify the word say p0Le as pole or $w!pE as swipe (can be a kind of image based but will try to find the words if possible).

Second Stage of the project will be filtering and keeping the words in a bucket. So here I will make different categories to define the words. We can say classes like Happy, Alert, Good, Bad, Angry etc. The p-value measure can be of significant help here as it will help in predicting properly the sentiments.

Third Stage of the project will be to implement the above values and plugin to our models for learning purpose. Here first I will implement Linear Regression and based on the results I will compare and train my deep-learning model which is RNN based LSTM. Long Short-term memory cell will be helpful if words are recurring more frequently in comments which actually happens a same word keeps on coming to hurt the sentiments. Linear Regression will use supervised model so we will might approximate values but with Recurring Neural Network we will get exact values and words that hurt sentiments.

Here I will show all graphs and statistical analysis based on Basic Learning model as well as Deep-learning model.
Fourth Stage of the project will be to test the data based on given training values. I will try to test with limited set of data on trained model and will check if both the models are providing sufficient results or not.

References:-

1) https://iamtrask.github.io/2015/11/15/anyone-can-code-lstm/
2) https://www.kaggle.com/eliotbarr/text-mining-with-sklearn-keras-mlp-lstm-cnn
3) https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/
4) Comment Abuse Classification with Deep Learning https://web.stanford.edu/class/cs224n/reports/2762092.pdf
5) http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/]