Machine Learning Engineer Nanodegree

Capstone Proposal

Sentiment Analysis Using convolutional Neural Networks

Ahmed Saafan

October 1st, 2019

Domain Background

Natural language processing (NLP) is one of the active brunches of the Artificial Intelligence/Machine Learning field that attract lots of research and attention. NLP as the name suggests deals with the natural human language using computers to analysis and find patterns in written and spoken language. NLP applications can be found in almost all language related technologies, like Siri or Alexa (speech recognition), Google's search suggestions, text symmetrization and lots of other applications. But one very interesting application is sentiment analysis (SA), where a model finds the sentiment behind a text, whether it's a sentence, a paragraph or document. SA can be used to classify reviews on movies or products, whether they are positive or negative (polarity), or to find the emotion behind them, angry, sad, happy ..etc. Historically NLP used hard-rules for analyzing text, then it used statistical methods which gave good results, but in recent years machine learning and deep learning proved to have better potential to solve the NLP challenging problems(Liu. 2012).

Sentiment analysis can be very useful for applications that are related or affected by the "mode" of the people/users. Companies for example may be interested to know how the people are reacting to a product or the company as a whole. A politician running for elections will be interested in how people are reacting to his to the opponents campaign. SA in other words can be used as a metric on how people "feel" about something, this is made easy by the amount of huge raw data that can be found on social media, where users are talking and sharing their opinions and feelings all the time(Pang and Lee. 2008).

For me personally I find the NLP very interesting field that I want to explore more by starting with one of the important branches of this field with its huge potential and applications that are growing with time.

Problem Statement

Given a movie review, the model should determine whether the review was a positive or a negative. The reviews are real ones taken form popular movie review websites that you usually read when looking a movie up. The reviews are tagged with their polarity and the model will find patterns to classify the reviews after training, which make this a supervised learning problem, that can be measured by the accuracy of the model.

Datasets and Inputs

For this project a large movie review dataset to be used. The data is collected from the IMDB website known as Large Movie Review Dataset 1. The data was collected by Stanford as a new and larger benchmark dataset, the dataset contains 50K movie reviews, 25k for training and 25k for testing. Each half is divided into positive and negative reviews. The dataset also contains unlabeled movie reviews, but this set will not be used in this project. The dataset also is balanced when dealing with popular movies, as the collectors limited the review/movie to be 30 max (Maas et al. 2011)

Training data:

- 25k reviews in total, in 2 files {positive and negative}
    - 12.5k are positive reviews
    - 12.5k are negative reviews

Testing data:

- 25k reviews in total, in 2 files {positive and negative}
    - 12.5k are positive reviews
    - 12.5k are negative reviews

The dataset will be used to train and test both of the benchmark and the deep learning model performing sentiment analysis on different movie reviews. The dataset is large enough to train deep learning models as they require huge amounts of data, also the dataset is very polar, which means that each review is strongly positive or negative, which will make life easier for the classification task.

Solution Statement

The presented solution to this problem is to use deep learning architecture for this task. Convolutional Neural Networks (CNN) proved to be a good architecture working with image processing where lots of features extracting is expected to get useful results, the same principle may be applied to text to extract useful features to help find the sentiment of the text. Keras provides good implementation for CNN dealing with text data which will be very useful for this project. The data first will need to be represented properly to work with the model, this process is known as vectorization and Teutonization. Then the data will be feed into the model so it can learn important features that will make it classify the text of the reviews and find the correct sentiment. This can be measured using the accuracy score as the data is labeled, and we can check the quality of the predictions easily with simple score.

Benchmark

Logistic Regression is chosen to be the benchmark model as it's a historical model that has been used in NLP tasks for a long time, also the model is easy to use and fast to train compared with different models. The baseline model will follow the general steps presented in the solution statement with special considerations to the difference between the two models.

The baseline model will train on the same training data and test the trained model on the testing set, then the accuracy score will be compared with the deep learning model.

Evaluation Metrics

As mentioned before the accuracy score will be used as an evaluation metric for this task, as the task is a supervised learning task with clearly polar data that are labeled. The accuracy score will point the percentage of the correctly classified reviews to the total classified reviews, we are interested in how well the model can classify the data and the accuracy score gives a simple, easy yet a good metric to evaluate our task.

Project Design

The workflow will be similar to what we have done in previous project during the nanodegree. Basically the model will be a simple one in terms of layers: there will be an embeding layer to vectorize the words locally, 1d CONV layer/s with different filter sizes and connects to a DENS layer to output the label. drop out and maxpooling will be used as well and different configurations will be tested to find the best architucture. Transfare learing in NLP is relativly new and is acheving good results aginst the state of art models, but for me I'm more intrested to work with models from scratch to learn more as I'm planing to work with Arabic in the future. The workflow will consists of major steps that are presented as follows:

Loading the data

First the data is loaded and assigned properly in order to work with it. The data can be obtained by Keras directly, but loading the data from the source directly will be a good training for me to explore working with raw data personally.

Clean and preprocess the data

The data is then cleaned from the HTML debris that can be found in a lot of the reviews. The data will be then vectorized and tokenized to be ready to work with machine learning models as the raw text data should be changed to a numerical data to be feed to the ML model. * optional : the data may be normalized using NLP techniques later as a fine-tuning step.

split the data for training/testing

The data will be split into training and testing sets using skit-learn.

train and evaluate the baseline model

The baseline model will be trained using the skit-learn package for fast implementation. Then the baseline model will be evaluated using the accuracy score.

train evaluate the CNN model

The CNN model will be trained using Keras, the model design is similar to the image processing one we applied before. The text data will require 1d conv nets as the only difference. The model will be evaluated using the accuracy score.

try different architectures

Deep learning is an applied field with each task has its proper architecture, this was seen while training the dog breeding project as different architectures were needed to find a final good one. So a number of architectures will be trained and evaluated

compare all models and choose the final model

After evaluating all the architecture, a final one will be chosen that proved to be the best architecture for this task.

Refrences

B. Liu, Sentiment analysis and opinion mining . San Rafael, CA: Morgan and Claypool Publishers.(2012)

B. Pang1,L. Lee Opinion mining and sentiment analysis.(2008)

A. Maas, R. Daly, P. Pham, D. Huang, A. Ng,and C .Potts. Learning Word Vectors for Sentiment Analysis. (2011). In The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).