Machine Learning Engineer Nanodegree
Capstone Proposal

Felix H.L Lou
June 15th, 2018

Multi-channel Convolutional Neural Network for Text Classification

1. Domain Background

1.1. Natural Language Processing & Deep Learning

Natural Language Processing, a.k.a NLP, is a study of automatic manipulation of natural language. In other words, it focuses on the interactions between human languages and computers, specifically on how to program computers to analyze and process a huge amount of natural language data. Although the study of Natural Language Processing has been around for more than 50 years, the problem of understanding text has never been really solved. With progressive technological advancement, NLP has grown of out of the field of linguistics, and people have been leveraging artificial intelligence to tackle the problem in recent year.

Deep Learning is one of the members of the Machine Learning family. Originally inspired by how the human brain functions, Deep Learning models can be enhanced by training them with more examples whilst increasing their depth (Layers), so-called Deep Neural Network. On top of that, Deep Learning models are really good at feature extraction/ feature learning, in which they turn raw data which could not have been used into desired form. For instance, people leverage these models to process words/ text to get word vectors, which are basically representations of certain groups of contextual-similar words.

There have been numbers of successful demonstrations of applying Deep Learning to real world NLP problems, including speech recognition, machine translation etc. These demonstrations encompass 3 reasons why Deep Learning is promising in NLP:

- Feature extraction/ Feature Learning - Features are learned by the model, but not pre-defined or specified
- Continued improvement - The performance of the model is highly data-driven and based on empirical results
- End-to-End models - It is a relatively general approach, and domain knowledge is not necessary in system construction

1.2. Motivation

The reason why I set out to investigate this domain is because I consider myself a highly customer-centric individual. Listening and understanding your customers comes second to none when running businesses; so if we can leverage technology to do this more effectively and efficiently with scalability, we can take customer experience to the next level. One of the popular applications is sentiment analysis where class labels represent

the emotional tone of the source text such as ‘positive’ or ‘negative’. In a real world setting, the text could be anything from movie, product review to forum comments and Tweets. In fact, this is my very first time tackling NLP problem with Machine Learning; so hopefully this would be a good taster.

2. Problem Statement

2.1. Description

In this project, we will be leveraging Deep Neural Network to perform supervised learning, in which the model will be learning to ‘read’ customers’ reviews to tell the relevant sentiment. In other words, the goal is to build a model to detect customers’ emotional tone in their reviews.

2.2. Breakdown

- Task: Classify reviews’ potential sentiment (Emotional tone)
- Performance: Accuracy/ F-score of the predictions on a test set
- Independent variable (Predictor): Customers’ review
- Dependent variable (Outcome): Customers’ sentiment (Positive/ Negative/ Neutral)

3. Datasets and Inputs

3.1. Source

We will be using a Kaggle dataset, which includes about 23000 Women’s E-Commerce Clothing reviews.

3.2. Dataset Content

This dataset includes 23486 observations and 10 independent variables. Each observation corresponds to a customer’s review, and includes other variables. Since this project primarily focuses on NLP, we will only be feeding the ‘Review Text’ as input into the model when building our Neural Network.

4. Solution Statement

4.1. Highlighted Techniques

We will be using multiple Convolutional Neural Networks with different kernel (gram) sizes to read customers’ reviews. Here are the highlighted techniques required for the approach:

- Word Embeddings - A learned representation for text where words that have the same meaning have a similar representation
- Convolutional Neural Networks (CNN) - Achieve feature extraction through weights sharing

4.2. Model Architecture

- Word2Vec - A statistical method for efficiently learning a standalone word embedding from a text corpus
- Multi-channel CNN - A combined CNN that reads text with different n-gram sizes (groups of words)
- Fully connected layer - A dense layer to interpret the features extracted by the CNN in terms of a predictive output

5. Benchmark Model

5.1. Compared Model

We will be comparing our results to the top kernel of this dataset on Kaggle:

- Predicted based on Bayes Algo

5.2. Details

The benchmark model adopts the Multinomial Naive Bayes Algorithm to make predictions. Although the person uses 'Recommended IND' and 'Rating' as target variables, the fact that it is a NLP classification problem does not change. We can test our Neural Network on the same target variables to compare the performance. For 'Recommended IND' and 'Rating', the person's model yielded a f-1 score of 0.88 and 0.94 respectively.

6. Evaluation Metrics

We will be evaluating and comparing the results based on the model's prediction accuracy on a test set. Specifically, since this is a classification problem, the metrics used to measure the results will be **accuracy** and **f-score**:

- **Accuracy -** Measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of predictions (the number of test data points)
- **Precision** - [True Positives/(True Positives + False Positives)]
- **Recall (sensitivity)** - [True Positives/(True Positives + False Negatives)]
- **F-score** - A metric that considers both precision and recall

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

- If $\beta = 0$, then we get **precision**
- If $\beta = \infty$, then we get **recall**
- For other values of β, if they are close to 0, we get something close to precision, if they are large numbers, then we get something close to recall, and if $\beta = 1$, then we get the **harmonic mean** of precision and recall

7. Project Design

7.1. Programming Language and Hightlighed Libraries

- **Python 3**
- **NumPy** - Open source scientific computing library
- **Pandas** - Open source library that provides high-performance, easy-to-use data structures and data analysis tools
- **StatsModels** - Open source library that provides classes and functions for the estimation of many different statistical models
- **Scikit-learn** - Open source Machine Learning library
- **Keras** - High-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.
- **TensorFlow** - Open source library for Deep Learning
- **Nltk** - Open source library for NLP

- **Gensim** - NLP library good for topic modelling

7.2. Exploratory Data Analysis & Statistical Inference

This will be the first part of the project, in which we will be exploring the dataset to see how customers like the products offered by the e-commerce store. We will be exploring independents variables like 'Rating' to see how customers are satisfied with their purchases. Statistical methodologies like Bootstrapping will be adopted to run simulations to draw probabilistic conclusions. On top of that, we will be using classical statistical models to fit the data to see if certain variables are predictive. The goal of this part of the project is to understand the dataset thoroughly and seek potential insights.

7.3. Text Preprocessing and Model Building & Training

This will be the second part of the project, in which we will be first splitting the dataset into training and testing set. We will then be leveraging NLP libraries to preprocess customers' reviews and create sentiment labels; so they will be in the desired form when being fed into our Neural Network. As for our model, here are the 3 main pieces of the architecture:

- Word2Vec - A statistical method for efficiently learning a standalone word embedding from a text corpus
- Multi-channel CNN - A combined CNN that reads text with different n-gram sizes (groups of words)
- Fully connected layer - A dense layer to interpret the features extracted by the CNN in terms of a predictive output

7.4. Testing and Model Refinement

This will be the last part of the project, in which we will be evaluating the trained model's performance on the test set. Refinement would be used for certain hyperparameters if necessary.

8. Reference

Convolutional Neural Networks for Sentence Classification

Fast and Robust Neural Network Joint Models for Statistical Machine Translation

Very Deep Convolutional Networks for End-to-End Speech Recognition

Word2Vec Tutorial - The Skip-Gram Model

Best Practices for Document Classification with Deep Learning