Capstone Proposal

Machine Learning Engineer Nanodegree

Sunil Kumar Rajput

March 01, 2019

Table of Contents

Sentiment Analysis on Amazon Reviews

Abstract

Analyzing and predicting consumers behaviour has already been a promising area of study with great value of research. By analyzing the polarity of the text, decision makers like companies who sell their products can effectively inspect the strength and weakness of their products. Reviews posted on Amazon are not only related to the product that are sold but also the service given to the customers. In this project I propose a system that performs the classification of the reviews.

Domain background

Sentimental analysis often requires to using a combination of techniques like Natural Language Processing and text analysis to identify the positive or negative sentences or emotions. The whole process has parts such as tokenization, stop word filtering, stemming, classification and prediction of sentiment. The more detailed process will be described in the Project design part of the proposal.

This technique holds a great hidden business value as it allows us to quantize the subjective information for further investigation. For example, by analyzing the polarity of the text, decision makers can effectively understand the strength and weakness of their product or even anticipate the complaints and the sales amount. Due to its vast potential there are a lot of researchers applying the topic of sentimental analysis.

I was always fascinated by the thought of making a machine understand the Natural Language the way human consume process and understand. It is very easy for a human to differentiate *"This product is good"* and *"This product is not so good"* but for machines it is a quite a work to understand that these are 2 opposite class of statements.

Problem statement

My goal is to implement sentimental analysis on the Amazon Review Dataset using transfer learning technique. I believe the modern transfer learning technique can help achieve a higher accuracy than traditional method of training the models from scratch. The model will be based on Neural networks, the good thing about neural networks is that they are really good at understanding pattern in data but the bottleneck is data, they need a lot data to train on.

This requires both huge data and longer training times, So I will prefer using a pre-trained 3-layer AWD-LSTM model[1] developed by Salesforce's research that is already been trained on 100 million tokens from Wikipedia[2] articles.

Datasets and inputs

Dataset Amazon Reviews has been fetched from Xiang Zhang's Google Drive [3] dir. The file named amazon_review_full_csv.tar.gz will be used.

The dataset has 3 columns name Ratings, Summary and Review. It has 3000000, 650000 training reviews and testing reviews respectively.

	Ratings	Summary	Review
0	3	more like funchuck	Gave this to my dad for a gag gift after direc...
1	5	Inspiring	I hope a lot of people hear this cd. We need m...
2	5	The best soundtrack ever to anything.	I'm reading a lot of reviews saying that this ...
3	4	Chrono Cross OST	The music of Yasunori Misuda is without questi...
4	5	Too good to be true	Probably the greatest soundtrack in history! U...
5	5	There's a reason for the price	There's a reason this CD is so expensive, even...
6	1	Buyer beware	This is a self-published book, and if you want...
7	4	Errors, but great story	I was a dissapointed to see errors on the back...
8	1	The Worst!	A complete waste of time. Typographical errors...
9	1	Oh please	I guess you have to be a romance novel lover f...

Figure 1 Dataset

I will be dropping the column Summary from the dataset.

The ratings more than 3 will be marked 1(Positive) and rating smaller than 3 will be marked as 0(negative). All rows having ratings equal to 3 will be removed from the dataset. I am ignoring all the neutral reviews.

After manipulating the dataset training count will 2400000 and testing count will be 520000.

Solution statement

As mentioned above I will take advantage of transfer learning and use a pre-trained language model (AWD LSTM) trained on Wikitext-103 dataset[4]. So, my model will already have a good enough sense of English sentences.

Using language model, I will build an RNN classifier that will predict the polarity of the review. I believe using a pre-trained LM will enhance the score of prediction.

Benchmark model

For a benchmark model I will first use a count vectorizer[10] which will convert the collection of words into a matrix of tokens and then build a Logistic classifier[11] on top of that. Please note in this process the model will be trained from scratch.

Performance of this model will be compared with the model trained using the transfer leaning technique.

Evaluation metrics

The performance of each classification model is evaluated using three statistical measures, classification accuracy, sensitivity and specificity.

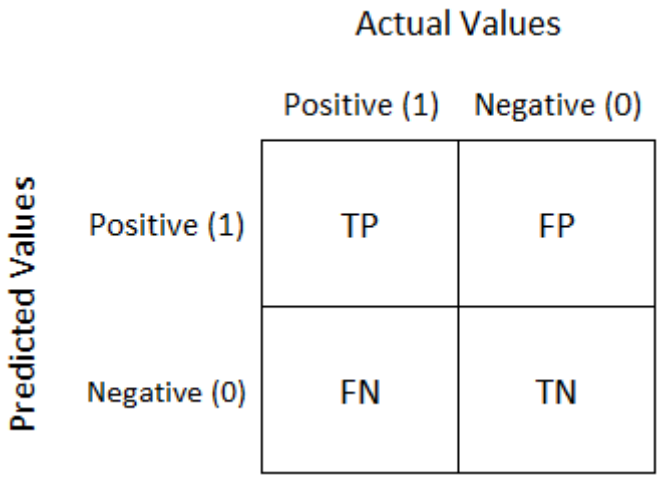

Figure 2 Confusion Matrix

In the above confusion matrix TP stands for True Positive (correctly predicted as positive), FP stands for False Positive (falsely predicted as positive), FN stands for False Negative (falsely predicted as negative) and TN stands for True Negative (correctly predicted as negative).

For any binary classification we have these 4 values calculated. Based on these values we can calculate accuracy, sensitivity and specificity.

Sensitivity (also called the true positive rate, the recall, or probability of detection in some fields) measures the proportion of actual positives that are correctly identified. It is calculated by dividing TP by total number of TP + TN.

Specificity (also called the true negative rate) measures the proportion of actual negatives that are correctly identified. It is calculated by dividing TN by total number of TN + TP.

Classification accuracy is defined as the ratio of the number of correctly classified cases and total number of cases. Mostly I will use accuracy as the evaluation metric. For further reading please see here[9].

During the training process the log loss will also tell the performance of model in real time.

Project design

AWD LSTM is a pre-trained language model. A Language model is a probability distribution over a sequence of words. Given such a sequence, say of length m, it assigns a probability to the whole sequence.

I am not preferring to use a word embedding approach for sentimental analysis. In language model the order of words matters but for a word embedding during training all it does is to predict the neighbouring words with a window regardless of the order.

I will first be training pre-trained Language model. It is a sequence to sequence to model having an encoder and a decoder part. After its training I will use to the encoder part of the language model to predict the class/sentiment of input review. The Classifier will most probably an RNN classifier.

Programming Languages and Libraries

- Python 3
- Scikit-learn[5]
- Pytorch[6]
- Fastai.text[7]

Workflow of project

- Explore the Dataset
    - Loading libraries and data
    - Peek at the training data
    - Statistical Summary
- Pre-Processing/Cleaning
    - Removing of neutral reviews
    - Remove any additional columns other than review and rating
    - Training and validation split
    - Tokenization
    - Numericalization of tokens
- Language Model training
    - Pass the numbers into the embedding layers
    - Load the Language model with existing weights
    - Replace the Encoder and decoder weights
    - Train the Language model
    - Save your Language model
- Classifier training
    - Load encoder of language model
    - Train the classifier
- Evaluation of results
- Final Conclusion

Citations

1. https://github.com/salesforce/awd-lstm-lm
2. https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/
3. https://drive.google.com/drive/folders/0Bz8a_Dbh9Qhbfll6bVpmNUtUcFdjYmF2SEpmZUZUcVNiMUw1TWN6RDV3a0JHT3kxLVhVR2M
4. https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/
5. https://scikit-learn.org/stable/
6. https://scikit-learn.org/stable/
7. https://docs.fast.ai/text.models.html#AWD_LSTM
8. Paper: https://arxiv.org/pdf/1708.02182.pdf
9. https://en.wikipedia.org/wiki/Sensitivity_and_specificity
10. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
11. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html