Machine Learning Engineer Nanodegree

Capstone Proposal

Jeff Gerlach
July 13th, 2019

Proposal

Domain Background

A major problem in online interactions (such as on message boards, comment threads, and video game chat) is toxic behavior. In this context, toxicity is defined as 'anything rude, disrespectful, or otherwise likely to make someone leave a discussion' [1], which with the anonymity of the internet can be quite common on popular social media platforms, games, and websites. While this can be manually moderated on small scales - say a blog author monitoring a few comments posted regarding an article they wrote - today's large social platforms necessitate automated detection of such behaviors due to the large volume of user-generated content and the desire of platform owners to keep such hostility at bay.

At the close of 2017, a large (around 2 million) comment database was released by the Civil Comments platform [2] when it shut down in an effort to help researchers improve civility in online conversations. Jigsaw, an Alphabet company, adopted this goal and further annotated the data set for research purposes. They also sponsored a Kaggle competition using this data.

This capstone project will focus on the Kaggle competition 'Toxic Comment Classification Challenge', which aims to detect toxicity across a diverse range of conversations.

I personally found this competition interesting, as it involves natural language processing and has the potential to show how tried and true methods like logistic regression (if used with the correct features) can keep up with cutting edge neural networks in terms of classification accuracy. Finally, it provides a convenient input data format that can be adapted into a web interface where users can submit comments and classify their toxicity rating as a final deliverable for this project.

Problem Statement

The goal of this Kaggle competition (and thus this capstone) is to build a machine learning model that can classify toxicity types using machine learning methods. The model must predict the probability of each type of

toxicity for each comment in the test dataset.

Previous research has been performed developing machine learning models to classify toxic interactions [3], and the goal of this Kaggle compeition was to open this particular dataset to the community to see if more accurate models[4] were possible.

Datasets and Inputs

The training and test datasets are provided on the Kaggle competition page[5]. These datasets are provided as CSV files with the following column names:

- `id`
- `comment_text` - input data
- `toxic` - target value
- `severe_toxic` - target value
- `obscene` - target value
- `threat` - target value
- `insult` - target value
- `identity_hate` - target value

The comments themselves are from Wikipedia, and the target values were manually labeled by human raters for the type of toxic behavior they contained.

The training dataset contains 160,000 rows (with the above 8 columns) while the test set contains 153,000 rows, with just 2 columns (the `id` and `comment_text`). A CSV file containing the test set true value labels is also provided as the competition is over, which allows for model accuracy testing without the need to submit a kernel on Kaggle. This will lead to nearly a 1.05:1 training/test split. I plan on taking 10% of the training data to use as a validation set while experimenting with different modeling approaches.

One factor to consider is the distribution of toxicity classifications present in the training data. Brief investigation of the training set reveals that nearly 90% of the comments do not have a toxicity classification, and the percentages of toxic classifications are not even - 'toxic' ratings are much more prevalent (15,000 examples), with 'obscene' and 'insult' (around 8,000 each) at nearly half that amount and the rest at or below 1,500 examples each. Thus the effects of class imbalance will definitely need to be investigated.

Training set class distribution (159,571 total comments):

- No toxicity: 143,346
- Toxic: 15,294
- Obscene: 8,449
- Insult: 7,877

- Severe Toxic: 1,595
- Identity Hate: 1,405
- Threat: 478

Another thing to note is that of the comments with toxic classifications, around 60% of those have more than one toxicity class associated to them, so each comment will need to have multiple class probabilities associated with it.

Solution Statement

The output of the model will be a CSV file containing the comment `id` and the predicting probabilities of each of the 6 toxicity types (between 0 and 1). The intent is that this file will be suitable for submission on the Kaggle platform. The output will be compared to the provided `test_labels.csv` to determine the accuracy of the model classifications without having to submit the file on Kaggle to speed development times. The model will need to be able to perform multi-class classification, and both regression and neural network architectures using Tensorflow will be explored. A web app will be included in the final deliverable where users can input comments and receive toxicity type predictions live using a client-side Tensorflow model.

Benchmark Model

The model I will use as a benchmark will be one of the highest-rated Kaggle kernel submissions using logistic regression - (Logistic regression with words and char n-grams | Kaggle) as this uses methods we have covered in the Nanodegree course and is able to achieve scores very close to more complicated neural network architectures.

Evaluation Metrics

The evaluation for this project will be based off of the metric used by the Kaggle competition: the combined mean ROC AUC of each column (area under the curve of the receiver operating characteristic for each toxicity classification type) - this was introduced towards the end of the of original Kaggle challenge due to changes in the dataset [6].

Project Design

First I will investigate the training data. I plan on visualizing the distribution of toxicity classes in the training set to get an idea of any class imbalances that may be present and need to be addressed. As the input data for the model is a large amount of text data, investigating various required text preprocessing steps[7] will need to be done, most likely using the SciKit Learn Python library. Non-standard characters will need to be cleaned, and most likely the text will need to be vectorized (and possibly the characters as well) to feed into the machine learning models. A stop word list will need to be chosen, a range of n-grams, and the max number of words

and/or characters I would like to keep will need to be decided upon and most likely these will be experimented with while developing my models. The training data will also need to be split into a validation set to help tune model hyperparamters and prevent overfitting.

I plan on first developing a logistic regression model as a baseline to validate my text-preprocessing steps and experiment with how changing different setting affects classification performance. Once this is complete, I will explore a Naive Bayes Support Vector Machine (SVM) and then more complicated neural network architectures using Tensorflow to see if I can achieve improved performance.

My first approach for a deep learning model will be at getting an LSTM (long short-term memory) recurrent neural network running, as this architecture seems to have success running with text data. I also plan to explore pre-trained models and ensembles, where multiple models are run and their outputs are averaged together before making final classifications.

Once I have experimented with the above model types, I will pick the best performing model and export it into a client-side Javascript model (using Tensorflow.js) and create a web application where users can type a comment into an input and get toxicity predictions on the fly. A finalized Jupyter notebook and report will be provided as well.

---

1. "Toxic Comment Classification Challenge - Overview" https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview ↩
2. "Saying Goodbye to Civil Comments" Bogdanoff, Aja. https://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d ↩
3. Wulczyn, Ellery, Nithum Thain, and Lucas Dixon. "Ex machina: Personal attacks seen at scale." Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017. ↩
4. Georgakopoulos, Spiros V., et al. "Convolutional neural networks for toxic comment classification." Proceedings of the 10th Hellenic Conference on Artificial Intelligence. ACM, 2018. ↩
5. "Toxic Comment Classification Challenge - Data" https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data ↩
6. "Toxic Comment Classification Challenge - Overview | Evaluation" https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview/evaluation ↩
7. Mohammad, Fahim. "Is preprocessing of text really worth your time for online comment classification?." arXiv preprint arXiv:1806.02908 (2018). ↩