Dellarontay Readus
Michael Kaplan
dreadus@stanford.edu
m2kaplan@stanford.edu

Classifying Native Advertisements and Sponsored Content

Abstract

Dynamic native advertising insertion[3] provides companies and institutions methods of producing native advertisements centered around content a user is already viewing. Casual users browsing the web can be bombarded by such native advertisements until the content seems ubiquitous to the content they are looking for. Convolutional 1-Dimensional neural networks are efficient at producing text classifiers for a number of classes in a number of domains and can provide a structure to classify native advertisements. This paper seeks to generate a general Keras neural network model to enable binary classification on sets of parsed html files from a social media aggregator and discuss the general method.

Introduction

Native advertisements, ads produced by publications or corporations that attempt to mimic an “organic” piece of content such as an unbiased consumer review, and which do not use the language and imagery associated with advertising, are often difficult for average consumers to differentiate from real content. User response to this has been mixed. Some users appreciate seeing advertisements that blend in well with a given website and do not disrupt the user experience[4], while other users believe that inserting advertisements that are not clearly marked as such is deceptive. Various text classification methods, including deep neural networks, have been used to both classify and produce sponsored content.[5] The input to our algorithm are sequences of texts that are propagated to output binary classification over the sponsored and non-sponsored content.

Purpose

As the number of legitimate and illegitimate online publications rises, new streams of sensationalized content come pouring into the minds of the public. Recent advances in AI, including deep neural net methods, have even allowed the automated production of sponsored content which closely mimics the existing content of a website.[3] Now more than ever, it is important for consumers to be able to easily identify the content they are consuming. Currently under European guidance, the GDPR has become an avid example of protecting consumer rights owned by the purchasers of technology. Privacy policies that are at least transparent have become the norm for any corporation seeking to do business in Europe. Already moral complications force many serious corporations to identify their advertisements plainly. In the future where consumer rights will continue to be protected, the liability companies have to inform users of sponsored native ads may be a legal one.

Data

Our final model used a sample of 10,000 html files from a Kaggle competition dataset of html files served by StumbleUpon, a content aggregator that served users with randomly selected posts based on their preferences and friend recommendations. This dataset contains roughly 300,000 files, each containing the full html content of a web page (text content, style formatting, and links to images or video). The articles varied considerably in page formatting.

Method and Related Work

We approached this text classification task with the hypothesis that it was closely related to the sentiment analysis and to fraud/”fake news” detection, since we hypothesized that the differences between sponsored and non-sponsored text would be based on sentiment. We began by testing a Convolutional Neural Network (CNN) architecture based on a model used to classify news articles of similar length and context (shared through social media) as the Kaggle data. Wang (2017) and Yang et. al (2018) use a CNN architecture to detect misleading news, mostly in shorter length articles of a political nature. Although is a great deal of prior research on both fake news/fraud detection and general sentiment analysis, we found few sources specifically attempting to classify unannounced native advertisements.

Every model that we created included Glove word embeddings in the first layer. This greatly reduced dimensionality compared with the one-hot encoding approach and took advantage of large text corpuses (such as Wikipedia), as we did not have enough “full sentence” data to capture syntax and grammar. Using Keras’ embedding layer and Stanford Glove’s pretrained weights from their co-occurrence matrix representation enabled our models to deal with relationships between words without having to train the layer ourselves. Setting the embedding layer to be non-trainable also greatly reduced the number of trainable parameters our model would calculate, thereby providing performance benefits and reducing runtime.

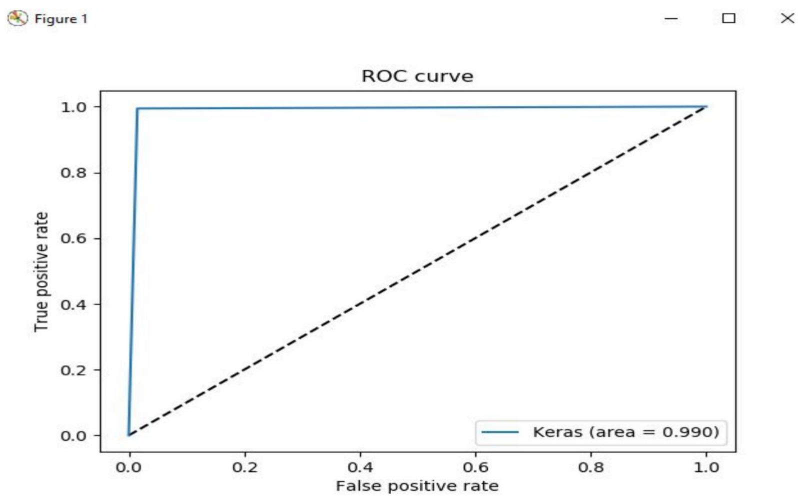
As previously mentioned, recurrent neural networks and convolutional 1-dimensional neural networks can be used to generate classifiers for text classification. Our original model included 3 GRU units to form an rnn with inputs n over the total number of features. We observed minimal gains in minimizing the loss of the validation set, so we moved on to an nn structure with 3 convolutional 1D layers built with additional max pooling layers, dropout, and batch normalization. This model seemed to produce the same results as the rnn example. To generate a model that would minimize the loss correctly required reexamining the dataset and performing random oversampling over the dataset to fix the class imbalance of 9(non-sponsored) to 1(sponsored) located there. We used the RandomOversampling method from the Imb_learn package to randomly repeat rows of the minority class. Sklearn provided a function to produce class weights that would scale the loss function appropriately for class imbalances; however, this method did not work to resolve the class imbalance. Instead it provided a way to check that random oversampling did in fact resample the training dataset correctly.

The titles and body text of a small subset of the dataset were used as input features to train the final model to accurately classify texts of sponsored or non-sponsored content. Using 2 features in the model only produced slightly better results ~2% over just using the title as the only feature, whereas using the body text as the only feature provided significantly worse results ~ -10%. Keras and its binary cross entropy loss function were implemented to solve this binary classification problem. We split our data into 80% training, 10% validation, and 10% test set.

Using a convnet model with 3 convolutional layers and 3 pooling layers, (similar to the model used in the Keras pre-trained word embeddings example for classifying news) we achieved a training set accuracy of ~.98 and a validation set accuracy of ~.96. Another metric for testing the performance of our model is the Area under Receiver Operator Characteristic Curve that was used to judge the Kaggle Dato-Native competition. Our final model received a validation set ROC score of 0.9896 which would give us the 3rd leaderboard position out of 274 competitors well within the Gold score. These accuracy numbers fluctuated for a small number of epochs. Changing optimization algorithms also did not change the ending accuracy. However, using an Rmsprop optimizer lead to much higher starting error but faster convergence to the same final accuracy.

Hyperparameters

We experimented with number of layers in the model, and number of convolutional units per layer. 128 units per layer was found to perform slightly (~1%) better than 32 or 64, though doubling the layer of hidden units significantly increased training time. We also found that adding more dense layers beyond 2 did not increase performance. We varied the learning rate on the log scale {0.001, 0.01, 0.05, 0.1} and found that a learning rate of 0.01 produced the best results. The mini-batch size was varied between 1 (Stochastic Gradient Descent) and 512 based on the the size of the training dataset. We found that any batch size below 32 produced wildly oscillating errors during training, while the difference in validation ROC and convergence speed was not noticable for batch sizes between 64 and 512. Through cross-validation, we found that a dropout of 0.2 brought the validation set error very close to the training set.



Results

Our main result was that a model using only article titles, or the first 200-300 characters of an article body, performed within 0.02 ROC of models which concatenated titles with features from the article body. The body text alone was not as informative as the titles, despite the fact that the body of the articles was much longer. This suggests that the title was a much more important feature than the body, link urls, or numeric features.

A relatively simple CNN model, built to filter phrase patterns of the length normally seen in a news article title (~3-6), performed well when compared with the baseline of non-neural network models used in the original Kaggle competition^todo put a link to kaggle leaderboard here. Our model did not require manual feature extraction or engineering beyond parsing html tags and counting characters. In contrast, most of the other high-performing models that were publicly shared in this Kaggle competition used some form of model ensemble or random forest.

The performance of the model was very sensitive to class imbalance in the training data. Training the initial model architecture on the highly imbalanced Kaggle dataset produced very poor ROC scores (~0.56-0.58). Introducing random oversampling of the minority class raised the ROC to ~0.9 without any other model architecture or hyperparameter adjustments. Our model code : https://github.com/Dellarontay/SponsoredContent.git

Next Steps

The GLove pre-trained word embeddings were chosen based off of similar text classification projects which also looked at social media content. With more data from a particular content source, we wish to investigate whether a word embedding trained specifically on data from that content source (i.e. forum comments vs news articles) would outperform a general corpus. We would also like to see if our model would generalize well to sponsored content delivered by other social media platforms/content aggregators, as all of the data in our study came from StumbleUpon.

Contributions

Dellarontay and Michael both worked to theorize, build, and update the Keras model. Dellarontay worked on providing visuals from running the model with various hyperparameters on his local machine with a GTX 970 NVIDIA gpu. Michael worked through numerous environment errors and examined many academic papers to ascertain the best possible model. Batch normalization was used before dropout thanks to his research. Dellarontay found through empirical testing parametric relu converged faster than a regular relu activation function.

References:

Alan S. Abrahams , Eloise Coupey , Eva X. Zhong , Reza Barkhi , Pete S. Manasantivongs, Audience targeting by B-to-B advertisement classification: A neural network approach, Expert Systems with Applications: An International Journal, v.40 n.8, p.2777-2791, June, 2013https://www.tdcommons.org/dpubs_series/1119

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). ACM, New York, NY, USA, 797-806. DOI: https://doi.org/10.1145/3132847.3132877

Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S Yu. 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. arXiv preprint arXiv:1806.00749 (2018).

S. Bajaj, ”’The Pope Has a New Baby!’ Fake News Detection Using Deep Learning”, Stanford University, 2017.

Ågren, A., & Ågren, C. (2018). Combating Fake News with Stance Detection using Recurrent Neural Networks.

Agarwal B., Mittal N. (2014) Text Classification Using Machine Learning Methods-A Survey. In: Babu B. et al. (eds) Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012. Advances in Intelligent Systems and Computing, vol 236. Springer, New Delhi

Bianca Harms, Tammo H.A. Bijmolt & Janny C. Hoekstra (2017) Digital Native Advertising: Practitioner Perspectives and a Research Agenda, Journal of Interactive Advertising, 17:2, 80-91, DOI: 10.1080/15252019.2017.1357513

Colin Campbell and Pamela E. Grimm The Challenges Native Advertising Poses: Exploring Potential FTC Responses And Identifying Research Needs. Journal of Public Policy & Marketing In-Press.

Kolari, P., Java, A., Finin, T., Oates, T., Joshi, A. Detecting Spam Blogs: A Machine Learning Approach. AAAI. 2006.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

Lin, Bo and Fu, Bin, "Automatic quality assessment of native advertisements", Technical Disclosure Commons, (March 30, 2018)

Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2018. Fake News Identification on Twitter with Hybrid CNN and RNN Models. In Proceedings of the 9th International Conference on Social Media and Society (SMSociety '18). ACM, New York, NY, USA, 226-230. DOI: https://doi.org/10.1145/3217804.3217917
Retrieved from: https://arxiv.org/pdf/1806.11316.pdf

https://github.com/keras-team/keras/blob/master/examples/pretrained_word_embeddings.py

Footnotes:

[1] Abrahams et al, 2013

[2] Ajao et al., 2018

[3] https://patents.google.com/patent/WO2015084457A1

[4] Harms et al. 2017
[5] Lin et al. 2018