Critical Thinking 6 - Feature Engineering and Hyperparameter Tuning

Karl Estes

Colorado State University Global

CSC 525: Principles of Machine Learning

Dr. Pubali Banerjee

January 30th, 2022

Critical Thinking 6 - Feature Engineering and Hyperparameter Tuning

1. Introduction

Feature engineering and hyperparameter tuning are two crucial steps in crafting and training a machine learning (ML) model. While there are numerous ways to address these two items, the list of potentially viable methods is narrowed down significantly by the nature of the problem being addressed and the chosen learning model. With that in mind, this paper discusses a theoretical feature engineering and hyperparameter tuning approach for the following scenario: **A text classification model that can identify fake news and/or abusive content**.

The following discussion is purely theoretical. The rationale behind the chosen model and potential feature selection and hyperparameter optimization strategies are explained in each relevant section.

2. Problem Breakdown

Fake news and targeted derogatory and hateful language have been given a chance to propagate with the growth of the internet. Nowadays, it is relatively easy to publish a website, create a blog, and share any type of information via the numerous online social networking (OSN) platforms. While the global web's interconnectivity and ease of information dissemination are beneficial, a digital space has been created where fake news can be readily shared and peddled as authentic information. The American public became very aware of this during the 2016 presidential election, and since then, there has been an increased effort to identify and combat fake news across media and OSN platforms.

When it comes to fake news generation and detection, the development of AI has helped and hindered each side in recent years. Consider the GROVER model produced by Zellers et al.

(2020)as a research venture. GROVER, which stands for **G**enerating a**R**ticles by **O**nly **V**iewing m**E**tadata **R**ecords, allows individuals to controllably and efficiently generate an entire fake news article. Zellers et al.'s approach differed from previous models by providing GROVER control of the generation of an entire web page. The model can create content in the style of numerous other news organizations and tailor language to sound like specific reporters. While GROVER has been shown to produce disinformation content that people rated as more trustworthy than similar content generated by humans, it has also been shown to be highly effective at detecting fake-news articles (Zellers et al., 2020). As Coldewey (2019) of *TechCrunch* noted, while GROVER is best at detecting its own fake articles, it can also detect generations of other well-known models such as OpenAI's GPT2 with relatively high accuracy.

The actual process for classifying content as information or misinformation commonly considers the problem as a version of unconditional language modeling (Bengio et al., 2001). GROVER built on this approach with metadata inclusion, and similar approaches are valuable in identifying cyberbullying on OSNs as well; Including network data alongside content-based features increased the accuracy of ML models classifying various levels of harassment in tweets (Salawu et al., 2020).

While those approaches added additional features to their training set, the core problem is still text classification, "the task of automatically placing pre-defined labels on previously unseen documents" (Scott & Matwin, 1999, p. 379). In the context of the following discussion, documents simply refer to any example passed to an ML model for classification purposes. With text classification, it is vital to consider the high-dimensionality of the data, the potential ambiguity and contextual underpinnings of language, and the discrepancy of rare positive class

examples that occur in some text datasets (Scott & Matwin, 1999). Given these considerations, the Long Short-Term Memory (LSTM) network appears to be a valid theoretical model for addressing the previously mentioned problem.

3. Theoretical Model

Before discussing the reasons for choosing an LSTM model, it is essential to note that LSTM or other recurrent neural network (RNN) based models are not the only viable solution for text classification. Scott & Matwin (1999) discuss RIPPER, a rule-based learner built for learning new rules to classify text in various ways. Rodríguez & Iglesias (2019) tested an encoder-decoder-based transformer network which successfully learned various textual classifications. Talpur & O'Sullivan (2020) demonstrated that decision trees and random forest models helped classify potential cyberbullying tweets in a multi-class environment. Convolutional neural networks (CNNs) have also shown promise in named entity recognition (NER), part of speech (POS) tagging, and natural language understanding (NLU) (Young et al., 2018)

An LSTM was chosen as the theoretical model, though, since it is an RNN based model which is designed to learn relationships in data sequences (Sherstinsky, 2020). Natural linguistic sequences arise when one considers a given language's lexical and grammatical structure. Grammatical rules indicate in which order words should appear and how the context and placement of words modify one another. Since modification can apply to preceding and succeeding words, Rodríguez & Iglesias (2019) noted that a bidirectional LSTM may be a promising model. Bidirectional LSTMs feed data both forward and backward, increasing the likelihood of learning bidirectional sequences. However, further discussion will focus on a

general LSTM since specific feature representation schemes (discussed in the following section) may remove contextual information.

4. Feature Engineering and Selection

With a chosen model, the issue now stands on how to adequately select and engineer features to maximize model learning and efficiency. While it is possible to feed text data directly into an LSTM, other approaches attempt to disambiguate word modifications and relationships and remove potentially noninformative information. Three of these representation schemes are Bag of Words (BoW), phrase-based representation (PBR), and hypernym-based representation (HBR).

The **Bag of Words** approach is a representation scheme that dominated text classification in the late 1990s and is still used today (Scott & Matwin, 1999). In the BoW approach, each feature corresponds to a single word in the training set. In order to reduce complexity, both infrequent and highly-frequent appearing words are commonly filtered out, and a stemming algorithm is commonly used to remove prefixes and suffixes.

Phrase-based representation attempts to preserve information lost in the BoW scheme by using whole phrases as features. BoW typically renders text incomprehensible to humans while making it machine-readable. PBR maintains a greater amount of human comprehensibility; however, “the main problem [with a] bag of phrases is the huge potential increase in the number of features” (Scott & Matwin, 1999, p.380). As such, a selection strategy is commonly employed to keep the feature set bounded and filter out non-meaningful sequences. Lewis (1992) also noted that PBR introduces several issues if Bayesian classification is employed.

A **hypernym representation** scheme falls between BoW and PBR in the general size of the produced features. Hypernyms denotes an "is a" relationship, and HBR attempts to map words with "low information" to potential hypernyms to yield more significant information gain. HBR typically relies on WordNet, a lexical database for the English language, containing information on synonymy and hypernymy relations (Miller et al., 1990). One potential downside to this approach is the loss of contextual understanding. WordNet may have difficulty choosing the correct sense for a word if it has different contextual meanings.

While it is not clear which of the representation schemes would produce the most accurate results with an LSTM model, if possible, preliminary testing and comparison with each representation scheme would most likely be the best approach. However, if a single approach had to be chosen, intuition might dictate PBR due to the sequence-based nature of phrases and the LSTM's recurrent structure being designed to handle such data.

5. Hyperparameter Optimization

Although proper feature selection and engineering are crucial, failure to optimize a model's hyperparameters can result in decreased performance regardless of proper data representation. Hyperparameter optimization (HPO) has not always been a straightforward process, and history has shown that difficulties in the investigatory procedures can turn the entire activity into more of an art than a science (Bergstra et al., 2011). The development of more rigid HPO methods has paved the way for more reproducible, scientific results, though, and the application of higher quality HPO methods has also been shown to increase LSTM performance specifically, even when compared to more recently designed ML models (Melis et al., 2017).

While many HPO methods exist, an LSTM for text classification might benefit significantly from a genetic algorithm (GA) based HPO. GAs are a type of population-based algorithm that are relatively simple to implement, can handle multiple different data types (continuous, categorical, and conditional variables), and can be deployed in highly parallelized training environments (Feurer & Hutter, 2019). Ease of implementation and parallelized training are already two positives, but the key to a GA approach is handling different hyperparameter data types. If the LSTM model were employed in a deep neural network configuration, then it would possess a number of important hyperparameters, including the number of layers, number of nodes per layer, learning rate, regularization rate, loss function, activation functions, optimization methods, and evaluation methods (Kumar et al., 2021). The data type (discrete/ continuous) and the conditional relationships among these variables are not consistent, a discrepancy that is easily handled by a GA-based approach.

Hyperparameter strings are also easily modeled as *chromosomes* which are crucial to the foundational operations of GAs (Mitchell, 1997). Since multiple variables can be encoded into a single chromosome, GAs can simultaneously tune multiple hyperparameters. The simultaneous tuning of variables was crucial to Kumar et al.'s (2021) demonstration that LSTM HPO via GA could perform better than other HPO methods. Given this data and previous discussion of Melis et al.'s (2017) findings, a GA-based HPO approach for an LSTM text classifier appears to be a theoretically viable approach.

6. Conclusion

Considering the problem scope of classifying a body of text as misinformation versus information, or abusive versus non-abusive content, an LSTM model appears to be a viable

theoretical approach. For feature engineering to ensure maximal efficiency and performance, some form of phrase-based representation of the text could be utilized. Given other fake news and cyberbullying models' use of metadata and network traffic, respectively, engineering other features surrounding the platform of type of text being analyzed may also be helpful. HPO via GA also appears to be a viable approach given other research on LSTM classification models.

Nonetheless, the listed approach is only one possible configuration to tackle a problem of this nature. As mentioned in section 3, numerous other models have shown promising results for text classification. An LSTM was chosen for this theoretical discussion because of the sequences nature of language; however, another model may better represent relationships between text and non-text data. If a significant number of metadata-like features were included, it might be prudent to consider a different ML approach.

References

Bengio, Y., Ducharme, R., & Vincent, P. (2001). A neural probabilistic language model. In *Advances in Neural Information Processing Systems* (pp. 932–938).

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, *24*. https://papers.nips.cc/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

Coldewey, D. (2019, June 10). To detect fake news, this AI first learned to write it. *TechCrunch*. https://social.techcrunch.com/2019/06/10/to-detect-fake-news-this-ai-first-learned-to-write-it/

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer.

Kumar, P., Batra, S., & Raman, B. (2021). Deep neural network hyper-parameter tuning through twofold genetic approach. *Soft Computing*, *25*(13), 8747–8771. https://doi.org/10.1007/s00500-021-05770-w

Lewis, D. D. (1992). *Representation and learning in information retrieval*. University of Massachusetts.

Melis, G., Dyer, C., & Blunsom, P. (2017). *On the state of the art of evaluation in neural language models*. https://arxiv.org/abs/1707.05589v2

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, *3*(4), 235–244. https://doi.org/10.1093/ijl/3.4.235

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Rodríguez, Á. I., & Iglesias, L. L. (2019). *Fake news detection using Deep Learning*. https://arxiv.org/abs/1910.03496v2

Salawu, S., He, Y., & Lumsden, J. (2020). Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing*, *11*(1), 3–24. https://doi.org/10.1109/TAFFC.2017.2761757

Scott, S., & Matwin, S. (1999). Feature engineering for text classification. *Proceedings of ICML-99, 16th International Conference on Machine Learning*, 379–388.

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Talpur, B. A., & O'Sullivan, D. (2020). Multi-Class imbalance in text classification: A feature engineering approach to detect cyberbullying in Twitter. *Informatics*, *7*(4), 22. https://doi.org/10.3390/informatics7040052

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2020). Defending against neural fake news. *ArXiv:1905.12616 [Cs]*. http://arxiv.org/abs/1905.12616