Feature Engineering and Hyperparameter Tuning

M. Danish

Computer Science, CSU Global

CSC525, Principles of Machine Learning

Dr. Joseph Issa

02/26/2023

In this paper, I will discuss hyperparameter selection, hyperparameter tuning, feature selection, and feature engineering for a theoretical machine learning model. Because my current professional interest is the design of smart chatbots that help connect my co-workers with our employer's organizational knowledge assets, the use case I chose to explore is that of a text classification model that detects "fake news." "Fake news" has been defined as everything from obvious satire to intentional—and malicious—attempts to deceive the public, but the common thread among all definitions is that it includes false information that is presented so confidently that a layperson in the subject has trouble distinguishing it from fact. Despite the fact that chatbots do not have a motive—at least, not *their own* motive—to deceive the user, I think such a model could be applied to a generative chatbot in order to help ensure the accuracy of the information it provides in answer to questions. In the next section, I will explain the use case in more detail.

Use Case: Text Classification in Automated Fact Checking

A problem I have noted with existing generative chatbots is that the information they relay back to users is not always accurate and does not credit its source(s)—i.e., they need some built-in way to check their facts and to ensure that the source information they're learning answers from is reputable. An example of this phenomenon is ChatGPT, which was banned in December from the software developer forum StackOverflow.com for frequently synthesizing incorrect code in response to programming questions (Rose, 2022). The problem with ChatGPT appears to have been corrected since the ban, but in my opinion, bad information should not ever be presented to users in the first place. Theoretically, a fact check could be implemented via a text classification machine learning model similar to the type used to help moderate social media sites and other internet forums.

Hyperparameter Tuning Methods

Machine Learning Model

Type

The paper *New explainability method for BERT-based model in fake news detection* (Szczepański et. al, 2021) describes a BERT-based machine learning model that has been used to identify "fake news," which its authors defined as purposely fabricated false information that is presented as true for the purposes of misleading the public. Szczepański et. al (2021) note that the exBAKE BERT-based model they reviewed had a better F1 score than other new machine learning models designed for the same purpose. Other researchers have suggested improving the BERT-based model by layering it with a Convolutional Neural Network, and testing with such a model resulted in a 98.90% score on test data (Szczepański, 2021).

Methods

exBAKE uses supervised learning in its text classification method. Researcher Shashank Gupta (2018), however, highlights a problem with supervised text classification models that would certainly apply to their use with a chatbot—Labeling the training text is a time-consuming endeavor. His team proposed a "Train Once, Test Anywhere" model that allows an algorithm to learn between sentences and categories on a large, noisy dataset, and then generalize what it learns to new categories and—ideally—new datasets. Their results were in the range of 72-74% accuracy (Gupta, 2018). Although this accuracy level was significantly lower than that achieved by comparable state-of-the-art models of the time, the results were still encouraging and suggested a potential for improvement (Pushpankar & Srivastava, 2017).

Hyperparameters

Hyperparameters for a fact-checking text-classification neural network that can be applied to a generative chatbot would include learning rate, architecture (number of layers), batch size, and dropout. Because we're focusing on accuracy, we might want the learning rate and batch size to be low. Dropout prevents overfitting—which we would be carefully avoiding in this model—so we would start with a small value (Radhakrishnan, 2017). The architecture could consist of one or two layers.

Extracting Features to Boost Model Classification

Some evidence suggests that there might be some words that occur in "fake news" at unusual frequencies. Sentiment, too, can be extracted to help determine whether information is truthful, and other models have already used sentiment analysis effectively for that purpose. Syntax is another feature that has been used successfully to help identify false information via syntax analysis.

Reputation—more specifically, of the information's source—could also be an extremely useful feature to capture. For example, we could apply reputation analysis to determine whether information scraped by a bot comes from a peer-reviewed source. Additionally, we could combine it with other analysis methods to capture features like peer review status and outcome.

Enhancing Efficiency with Feature Engineering

Missing Data

Missing data may be *the* most important issue to resolve before running any kind of data analysis, whether a machine learning model is used or not. When you do not account for missing data, you cannot be sure your analysis really means what you think it means, and you may spend unnecessary time chasing down answers that are later proven to be completely incorrect,

especially when it comes to questions of cause and effect. Therefore, missing data can—and should—be addressed during the feature engineering stage of working with machine learning models.

One way to deal with missing data is to simply discard other data that are associated with it—for example, if we're trying to find a correlation between preferences and age, and we find that we're missing age data for some of our sample, then we simply don't include the subjects whose ages are missing in our analysis. Another technique is imputation, in which we replace the missing data with the median value as calculated from the other samples or with 0 (Rençberoğlu, 2019).

Standard Deviation

Another issue that has to be addressed with feature engineering is data that falls outside of the standard deviation—i.e., the variability—as calculated from the dataset's distribution. Outlier values can be dropped or capped (Rençberoğlu, 2019). For the use case discussed in this paper, it makes sense to drop outlier values rather than allow them to affect the dataset's distribution and, therefore, the outcome of the analysis. If we want answers and other information provided by generative chatbots to be seen as equal to, say, reading an encyclopedia, then we certainly don't want to allow them to appear to be engaging in speculation—or, worse, lending credence to fringe hypotheses.

Feature Splitting and Grouping

There are a handful of techniques that can be used to group features. First, it is often helpful to use splitting to extract features—for example, from strings that have not yet been split into columns—so that they *can* be grouped. Grouping then increases efficiency by allowing us to calculate counts, sums, frequencies, and ratios that can be used to speed up analyses. Grouping

methods also help prevent overfitting, but there can be tradeoffs with performance (Rençberoğlu, 2019).

Conclusion

A text classification model could be applied to a generative chatbot that incorporates machine learning techniques to synthesize answers to users' questions. In the same way text classification NNs can be used to identify "fake news," they could be applied to the information learned by generative chatbots and used to detect incorrect answers along with those that are intentionally fake. Hyperparameters for a fact-checking model would include learning rate, batch size, dropout, and architecture. Effectiveness of the model could be enhanced by extracting word frequency, sentiment, and syntax. Efficiency could be enhanced by engineering around missing data and outliers in data. Additionally, feature splitting and grouping could be used to help prevent overfitting.

References

Gupta, S. (2018, January 11). *Automated text classification using machine learning*. Towards Data Science. https://towardsdatascience.com/automated-text-classification-using-machine-learning-3df4f4f9570b

Pushpankar, K. & Srivastava, M. (2017). Train once, test anywhere: zero-shot learning for text classification. *Parallel Dots, Inc.* https://www.researchgate.net/publication/321902262_Train_Once_Test_Anywhere_Zero-Shot_Learning_for_Text_Classification/fulltext/5a388709458515919e71fdd3/Train-Once-Test-Anywhere-Zero-Shot-Learning-for-Text-Classification.pdf

Radhakrishnan, P. (2017, August 9). *What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?* Towards Data Science. https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

Rençberoğlu, E. (2019, April 1). *Fundamental techniques of feature engineering for machine learning*. Towards Data Science. https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114

Rose, J. (2022, December 5). *Stack Overflow bans ChatGPT for constantly giving wrong answers*. VICE. https://www.vice.com/en/article/wxnaem/stack-overflow-bans-chatgpt-for-constantly-giving-wrong-answers

Szczepański, M., Pawlicki, M., Kozik, R., & Choraś, M. (2021). New explainability method for BERT-based model in fake news detection. *Scientific Reports*, *11*(1), 23705. https://doi.org/10.1038/s41598-021-03100-6