Fake News Detection Model

Zach Holcomb

My project aims to create a neural network that will determine if a news article contains real or fake news

Data Preprocessing

The data is from Kaggle. It gives us 2 datasets for real and fake news, each with over 20,000 articles. The text in our data contains a description of the article followed by a dash and then a short snippet of text from the article, most likely the title or sub-headings.

To prepare the data, we need to ensure each article is formatted the same. We do this by first forcing all letters to be lowercase. We then remove the article descriptors as well as any non-alphanumeric characters:

Once our text is formatted correctly, we then tokenize it to prepare it for training.

Selection, training, and use of a machine learning model

For our task, we want to use a neural network for maximum performance. This could be done with a simpler model such as random forest, but that isn't as exciting. Since we are working with text, a recurrent neural network would work best. We will specifically be working with long short-term memory to ensure the model uses the full context of the text.

To train our model, we use 20% of our data for validation and use the other 80% for training. We use early stopping based on validation loss to avoid overfitting. For our loss function, we use a weighted loss to prioritize accuracy on fake news articles.

Interpretation of results

Once the model is trained, we can look at some performance metrics:

Test Accuracy: 97.49%

Test Recall: 98.21%

Test Precision: 97.25%

Overall, the model performs very well on both fake and real articles. We successfully prioritized recall to avoid type 2 errors such as labelling fake news as real.

The biggest limitation here is the data it was trained on. The model might not work as well on articles that aren't from the same publishers as the ones on our data. Our data is also at least 6 years old, so newer articles might contain words that our model is unfamiliar with.

Communication of results

Now that our model is successfully trained, we can use it on any article (after applying our preprocessing) and have our model label it as fake or real news with around 97.5% accuracy.

For example, we can use the model on the title of this article on the recent assassination of the UnitedHealthcare CEO to determine if it is real.

Our model returned a value of 0.9994198. Our model outputs a value between 0 and 1 where values closer to 1 indicate a higher confidence that the article contains real news. So, our model is very confident that this article is real.

Ethical implications

The biggest concern with this model is bias towards specific news outlets. If our model wasn't trained on news from a specific outlet, it may be less accurate and incorrectly label articles from that outlet as fake.

Another issue that would need to be thoroughly addressed is personal data. Some articles may contain personal data like names and addresses. Proper measures would need to be taken to avoid conflict with some data privacy laws.

One way that this model could be misused is by using it to learn how to make more convincing fake news articles. While this may not be illegal, it would make it harder for the average person to find and identify real news.

Grading Category: I believe I earned a 1 on this project. Not only did I train an accurate model, I went beyond the requirements by using new techniques like a custom loss function, an unfamiliar loss function (leaky ReLU), and creating a text based model with a categorical output. Notebook, ChatGPT chat