Predicting Virality of Online News Articles using Textual Content

Meredith Benson
University of Michigan / EECS 595
merbenso@umich.edu

1 Introduction

News holds a lot of power in today's society. It influences and impacts people in a multitude of ways. It can create hysteria, or, alternatively, can construct a false sense of security. It can influence the way people behave, the way they think, and the way they vote. Being able to predict the impact that a news article will have, before it is released, would be an incredibly useful tool for journalists everywhere. It would allow virtual media companies to consider the influence an article will have before they publish it, enabling them to think more carefully about the best way to present their piece. The aim of this research project is to predict the relative virality of an online news article, based solely on the textual data contained in the title and headline of the article. The title, and subsequently the headline, are the first pieces of the article a viewer interfaces with, and as such they hold the most power to hook the viewer and get them to continue reading. This, in turn, increases a viewer's likelihood of sharing the article and boosting the virality of the piece. A prediction in this manner would be useful to journalists and online news corporations, as it would also allow them to test out multiple potential titles to see which are likely to reach the most people.

2 Background

2.1 Motivation

This project was motivated by a few different factors. There has been a considerable amount of research into what exactly drives an article to go viral, with some promising findings. First, a study out of Columbia University estimated that 59% of links that are shared on Twitter are never clicked on by the person sharing it (Gabielkov et al., 2016). This means that the majority of the time, people share news articles on Twitter without ever reading them, but instead based solely on their title. This was a motivating statistic for this project, because it suggests that article titles contain some factor that directly correlates with an article's potential virality and may be used by a deep learning neural network to accurately predict virality. Second, several studies have shown that the sentiment of an article title contributes to the virality of the piece. A study by D. Molina et al. uses sentiment analysis to show that a negative sentiment enhances the virality of a news article (2011). Another study by Rameez et al. also shows that a tweet with a negative sentiment has a positive correlation with number of shares, while a tweet with a positive sentiment has a negative correlation with number of shares (2022). This gives additional validation that there is some information contained in the text content of the title that has an effect on the probabilistic virality of the news article. Finally, in the study of media virality, the different reasons behind sharing an article on social media are often cited as "information utility, opinion leadership, emotional impact, relevance, entertainment, and social cohesion" (D. Molina et al., 2021). These are well documented motivations behind sharing an article. Pairing this information with the fact that most people do not read an article before sharing it suggests that these 6 dimensions appear in some form in the title of the article, providing yet more evidence that virality information can be captured in the textual content of an article's title.

2.2 Definition of Virality

Virality is a nebulous concept, and can be defined in many ways. Oxford Dictionary defines virality as "the tendency of an image, video, or piece of information to be circulated rapidly and widely from one internet user to another." There are a number of ways such a concept could be defined quantitatively: by number of views, number of clicks, number of comments, number of likes, etc. For the purposes of this paper, virality will be defined in

terms of the number of shares an article receives. This appears to most closely match the definition given by Oxford Dictionary, and it is also the only metric that contributes directly to more people coming across the article, which further boosts the virality of the piece. Moreover, it is generally thought that shares hold more "weight" in social media algorithms than other post-specific metrics such as likes or comments. Due to the commercial nature of social media platforms, the specifics of how their algorithms actually work are kept hidden, but it appears that across most social media sites, shares are considered a more valuable indicator that an article should be shown to more people, as opposed to likes or comments. The decision to share something with your friends, family, coworkers, or even more broadly, your followers, takes more time and thought than other possible post interactions, and is therefore likely considered more heavily in a platform's algorithm when deciding if it should show a post to more people. For these reasons, it was determined that number of shares is the best way to define and compare virality.

2.3 Clickbait

Another factor to be considered in this project is the concept of clickbait. Clickbaiting occurs when a piece of media is given an extremely interesting, outrageous, or curiosity-inducing title that is misleading in some manner, which will prompt people to click on it to learn more but become disappointed when they see that the actual content does not live up to its title. Clickbait is viewed as having poor journalistic integrity, and for that reason, the model should avoid preferring 'clickbaity' titles if possible. Training the model based on number of shares rather than number of clicks or views helps to reduce the likelihood of the model learning to predict clickbait as more viral, as clickbait is likely to be clicked, but less likely to be shared if the person does click on the article and read it. Though, it should be noted that this is not a perfect solution, since, as mentioned previously, roughly 6 out of 10 people share articles without reading them, meaning some clickbait is likely to be shared. Ultimately, however, there is no good metric to use to entirely avoid counting clickbait articles as viral. Furthermore, recent research (D. Molina et al., 2021) has shown that though clickbait is relatively easy for a person to spot, it is still difficult to train artificial intelligence to be able to recognize it reliably. It is worth mentioning, though, that this same study determined that clickbait articles appear to be less engaging than non-clickbait articles, and for the 40% of people who do read articles before sharing, being non-clickbait will thus increase their likelihood of sharing it. They also determined that articles with clickbaity titles are viewed as less credible and invoke the reader's curiosity less than genuine, non-clickbait articles, further reducing their likelihood of sharing clickbait articles.

3 Related Work

There was a paper (Rameez et al., 2022) published earlier this year that proposed a classification model called ViralBERT to predict the virality class of a tweet on the social media platform Twitter. Their proposed model uses both natural language processing techniques on the textual content of the tweet as well as the tweet metadata available at time of publication, such as the time of day the tweet was posted and the number of followers the account has. In terms of natural language processing, they used a RoBERTa-based pre-trained model to perform sentiment analysis, which was included as one feature of the dataset. Additionally, they used a BERT-based pre-trained model called BERTweet, which was fine tuned on a corpus of 850 million tweets, to produce a pooled-output vector encoding of the tweet, which was also included as a feature in the dataset. They found that running ViralBERT using just the sentiment and BERT encoding of the tweet did not perform better than the baseline models, and concluded that the numerical data they factored in was necessary to achieve a decent performance. However, they did determine that the sentiment of the tweet was one of the most important factors that the model considers in the virality prediction, alongside follower count and number of hashtags used.

Another paper (L. Lopez et al., 2022) published this year attempted to solve the same problem as this project, predicting the relative virality of a news article based on the article's title. They used a couple different approaches, investigating both regression and classification models. The goal of the regression models was to estimate the difference in number of clicks between a pair of headlines, while the goal of the classification model was to predict which of the two headlines would ultimately receive more clicks. They found that their regression models were unable to meet the baseline, and were

therefore discarded. Their classification models, however, performed significantly better than the baseline. Their best classification model was a neural network that used BERT as the encoder for the titles, producing an ultimate accuracy of 64%.

4 Approach

4.1 Classification Task

There is no set definition for how many views, likes, or shares a piece of media has to receive before it is officially considered "viral." Instead, virality is most easily defined in terms of relation; it is unambiguous to determine which of two articles went more viral, but it is much more difficult to look at a number of shares received on a piece and decide if that number of shares constitutes virality. Though it may be more natural to think of virality prediction as a regression task, i.e. predicting the approximate number of shares an article will receive, recent work has shown that such regression results are often poor. There are multiple ways one could approach the problem of virality prediction, but for this project it will be attempted as a classification task, where the model will be given two titles and will choose the more viral of the two.

4.2 The Dataset

Due to the given timeline for this project, it was decided that a pre-existing dataset should be used rather than scraping and collecting new data for the purpose of this project. The dataset that was ultimately selected was the Newspop dataset from the machine learning community Huggingface (Moniz and Torgo, 2018). The Newspop dataset contains information surrounding news articles collected from well-known news aggregators such as Google News and Yahoo! News between November 2015 and July 2016. Each article surrounds one of four topics: President Obama, Palestine, the tech company Microsoft, and the economy. Each entry includes the title of the article as shared on social media, the headline of the article (or the lede of the article, in the case that a headline was not present), the topic (out of the four previously mentioned), the date the article was published, the source it came from, and finally the number of shares it received on each of the three social media sites, LinkedIn, Google+, and Facebook. This dataset was selected for a number of reasons: it is a large dataset, containing about 100,000 entries; it uses number of shares as the measure of popularity, which, as discussed previously, has been determined to be the best metric for measuring virality; and it contains data from not one, but three different social media platforms, which provides a more well-rounded view of the actual popularity of the piece.

4.3 Data Exploration

Because a third party dataset is being used for the project, it was important to get a feel for the shape and potential biases in the newspop dataset before using it to construct the dataset that will ultimately be fed to the model created for this project.

The first area that was investigated was the distribution of the four topics. A graph was created to get a quick, simplified view of the number of articles corresponding to each topic, as shown below.

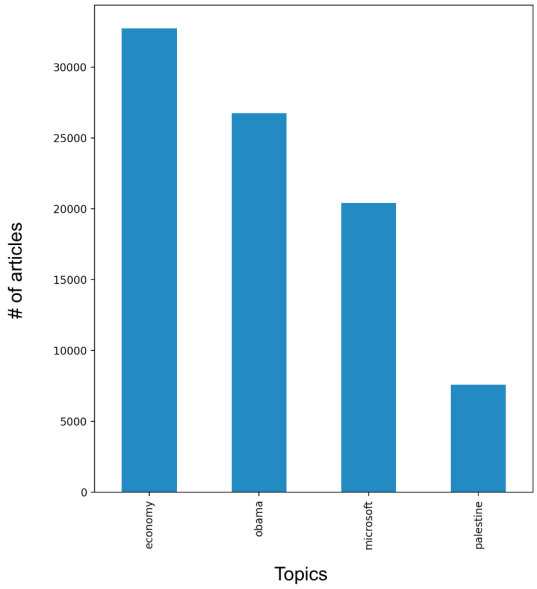

Figure 1: Each topic in the dataset plotted against the number of articles in the dataset that surround that topic.

It is clear that there is not an even distribution of articles pertaining to each of the four topics. Although this is not ideal, it was determined that this was acceptable, as it is a more natural model for conversation happening on social media platforms. In real-world scenarios, some topics will be mentioned much more frequently than others, and this distribution of data models that.

Additionally, it was important to check how the natural virality of each topic compares. The average number of shares per topic were plotted to get a feel for this.

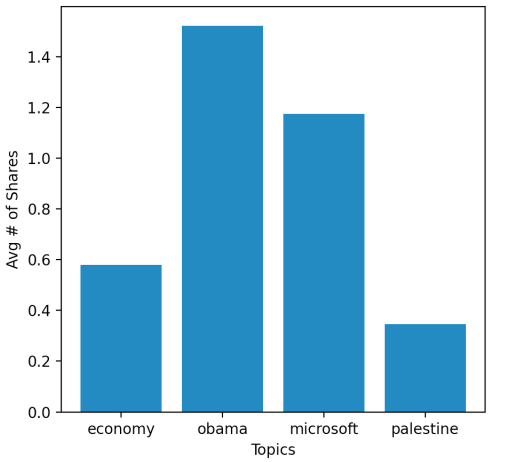

Figure 2: The average number of shares that each topic in the dataset received.

Based on the dataset, the topic of an article does appear to be an indicator of virality. As shown in the graph, articles on Obama and Microsoft tend to receive more shares than articles on the economy or Palestine. This is acceptable, because though there is some bias in the data, this bias stems from the humans behind the screen that are choosing to share certain articles more than others based on their topic. The time frame that this data was collected was 2015-2016, which was Obama's last year as president. It was also a particularly controversial election year, and it seems natural that articles surrounding the topic of the presidency would be more likely to go viral, as it was a subject that factored heavily into Americans' discourse at the time. For this reason, it is logical that the notion of topic should hold some indication for the potential virality of an article, and is an asset to the neural network in determining the likelihood of an article going viral.

4.4 Dataset Construction

As this project will be approached as a classification task, a new dataset had to be constructed to be used as input to the model, where each entry would contain two article titles and a class label. This project uses binary classification, where the class label 0 indicates that the first title is more viral than the second, and 1 indicates that the second title is more viral than the first. In order to pair the articles and produce such a label, the information in each entry had to be manipulated and condensed.

The first thing to look for were entries in the Newspop dataset that did not have data collected on the number of shares received. If a row was missing the number of shares on all three social media platforms, it would not contain any valuable information, and therefore such rows needed to be dropped. There were 5,744 entries that were dropped during this step. Entries that were missing share counts from only one or two rows were kept, as they still contained information that could be used by the classifier.

The next thing to be considered were articles for which the number of shares was collected, but which received 0 shares across all of the social media platforms. There were 14,875, or about 17%, of the entries that received no shares across all three of the platforms. This is useful information, as these articles show a strong example of what is not likely to go viral. For this reason, articles that received 0 shares on one or more social media site were kept.

After dropping the irrelevant data, the next task that needed to be done was to come up with one average share count for each article, based on the three different share counts given from each social media site. Before the share counts for each site could be averaged into one general number of shares, they each needed to be normalized. For example, if the average number of shares on Facebook is much higher than Google+ or LinkedIn, then averaging the number of LinkedIn and Google+ shares for an article that is missing the Facebook count would end up with an artificially lower score. To solve this issue, the average number of shares for each social media site were individually calculated, and then each share count was normalized by dividing it by the average number of shares for its respective social media site. The average number of shares for each site, as shown in table 1, differ drastically, proving the importance of this step.

Site	Avg # Shares
Facebook	129.36
Google+	4.21
LinkedIn	17.70

Table 1: Social media sites and an article's average number of shares.

Additionally, this normalization step allows us to take into consideration that articles may go viral within a specific platform. For example, the highlighted article in figure 3 went "super viral" on Google+, but received an average amount of shares

on Facebook, and no shares at all on LinkedIn. It is important that this concept is represented in the overall number of shares, which this normalization step preserves.

facebook	google+	linkedin	average # shares
8.309950	9.028273	0.169492	5.835905
4.993700	4.038964	0.000000	3.010888
0.146874	0.000000	4.632768	1.593214
0.131413	0.000000	0.000000	0.043804
0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000
0.919892	4.989309	0.000000	1.969734
0.525653	20.194821	0.000000	6.906824
0.123683	0.000000	0.000000	0.041228
0.672526	11.166548	0.000000	3.946358

Figure 3: An entry for an article in the dataset that went super viral on Google+, receiving 20x the average number of shares, but did not go viral on Facebook or LinkedIn.

The overall number of shares was then found by averaging the normalized number of shares each article received, excluding any time a share count was missing for a specific social media platform. This resulting number of shares was then visualized to get an idea of the shape of this newly condensed feature, as shown below.

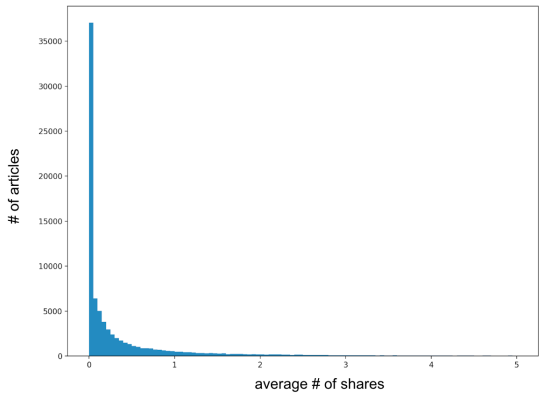

Figure 4: The number of articles in the dataset plotted against the average number of shares each article received.

As can be seen, the data is skewed towards receiving very few shares. In order to ensure that information is preserved in every entry in the target dataset, where each entry is two articles paired together, each article receiving 0 shares needs to be paired up with an article receiving more than 0 shares.

Since information, or entropy, will inevitably be lost by reducing the concrete number of shares received down to a simple class label, the goal in constructing the dataset was to retain as much of the information as possible. To do this, we wanted as much of a difference in the number of shares received between the two titles as possible, for every pair of titles in the new dataset. To achieve this, the dataset was sorted into ascending order based on number of shares, and then split in half. Each ith entry in the first half was then paired with the ith entry in the second half, effectively maintaining the largest possible gap in virality for every pair of articles. In order to differentiate class labels, a random coin was flipped for each pair to determine if the more viral article would be placed in the first position, corresponding to class 0, or the second position, corresponding to class 1. This completed the construction of the dataset needed for the purpose of this project.

4.5 Neural Network Model

For the classifier, a deep learning neural network model was constructed. The model first uses a BERT Tokenizer to transform each title into input that can be recognized by the BERT encoder. Then, the tokenized titles are individually given to BERT, which produces a unique pooled output vector that represents the entire title. These vector encodings of each title are then passed through a concatenation layer which appends the second title encoding to the first, and then a dropout layer to help prevent overfitting. It is then passed to the linear layer, and then finally the dense layer with an activation function which predicts a label for the combination of titles. A prediction of 0 indicates that the first title is more viral than the second, while 1 indicates that the second title is more viral than the first. A diagram of the model architecture can be seen in figure 5.

5 Results

5.1 Evaluation

As this is a classification project, the results of the classifier will be evaluated by looking at the accuracy of the model. The accuracy of the model represents the percentage of predictions that the classifier made and got correct. Since this is a binary classification task, there is a natural baseline of 50%, corresponding to a random guess. Additionally, as mentioned previously, the study by L. Lopez et al. with a similar setup was able to

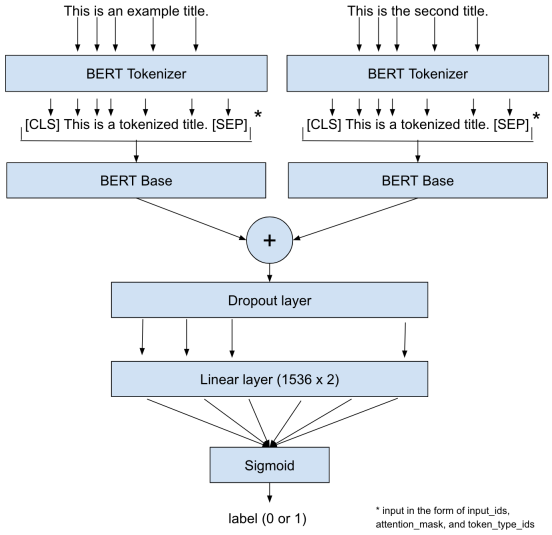

Figure 5: Architecture of the neural network classifier model.

achieve a 64% accuracy (2022), providing another baseline for comparison using current research in the field. When the model created for this project was running using article titles as input, it achieved a highest accuracy of 76.6%.

5.2 Headlines

As an additional experiment, it was hypothesized that including an article's headline in addition to its title would further improve the model, because it would include more potential information for the model to learn from. The dataset was modified to append each article's headline to its title, using the first sentences of the article if the headline was missing. It should be noted that BERT has a cap of 512 words as the maximum input length, so headlines that ran longer than this were cut off after 512 words. This new dataset was fed to the model, and after a bit of hyperparameter tuning, the model produced an output of 79.2%, a nearly 3% boost, confirming the hypothesis that this additional information would help the model improve its decision-making process.

Model	Test Accuracy
Baseline	50%
L. Lopez et al.	64%
Classifier w/ titles	76.6%
Classifier w/ titles + headlines	79.2%

Table 2: Results of Classification Models

6 Discussion

The ultimate accuracy achieved by the model was 79.2%, which was produced by the model using an article's title as well as its headline to predict virality. This is significantly higher than L. Lopez et al, who also approached the task by classifying which of two titles is more likely to go viral and achieved a highest accuracy of 64%. Even without use of headlines, the accuracy of the model in this study is still more than 10% higher than the previously mentioned study. It is suspected that this is occurring for a couple of reasons. First, the earlier study used a dataset of news articles collected from a site called Upworthy, which is a news site dedicated to sharing only positive news stories. As mentioned earlier, journal articles with a positive sentiment are less likely to go viral than those with a negative sentiment. Although it is not impossible for a happy news article to go viral, it is possible that their dataset does not contain many examples of viral articles and therefore their classifier would have less information from which to learn the differences between more viral and less viral titles. The dataset used for this project was collected from several different news outlets over a long period of time and therefore likely has a much better spread of both sentiment and virality. Second, the data used for the previous study used number of clicks and number of impressions as the target variables, defining the virality rate as the number of people who received an article versus the number of people who clicked on it. It is possible that number of shares, as used in this paper to define virality, is a better measure of actual human interest. As mentioned before, clickbait is not more likely to go viral, though it is likely to receive many clicks. Considering a clickbait article to be viral potentially washes out some of the information learned by the classifier on genuine, non-clickbait articles about what actually makes a piece viral. These are potential reasons that this project was able to achieve a higher accuracy than previous work.

7 Conclusion

A news story, especially a viral one, has the ability to impact society in a myriad of ways. The ability to predict the effect a news article may have, based solely on information available before the time of publishing, would be a valuable asset to virtual media companies. This paper investigated the prediction of virality of a news article, and cre-

ated a deep learning classifier that was able to pick the more viral of two titles with an 76.6% accuracy. Additionally, it was found that by also including the headlines of the articles, the accuracy could be further improved to 79.2%.

One of the biggest limitations of this study dataset is that all of the articles surrounded only 4 topics. In the future, it would be interesting to try recreating this experiment on a dataset that more naturally represents the modern digital newscape, including articles on a broader array of topics, as well as collecting information from more relevant and widely-used social media sites. Additionally, since both BERT and these neural networks operate as a sort-of 'blackbox', it would provide vital insight into the process of virality prediction if the reasoning behind the classifier's decision-making process was investigated in the future. Though it is suspected that both sentiment and topic are factors that are weighed in the process, there are likely other factors that remain yet undiscovered or unconfirmed.

References

Maria D. Molina, S. Shyam Sundar, Md Main Uddin Rony, Naeemul Hassan, Thai Le, and Dongwon Lee. 2021. Does clickbait actually attract more clicks? three clickbait studies you must read. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. 2016. Social clicks: What and who gets read on twitter? *SIGMETRICS Perform. Eval. Rev.*, 44(1):179–192.

Lars Kai Hansen, Adam Arvidsson, Finn Aarup Nielsen, Elanor Colleoni, and Michael Etter. 2011. Good friends, bad news - affect and virality in twitter. In *Future Information Technology*, pages 34–43, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yesid L. Lopez, Didier Grimaldi, Sebastian Garcia, Jonatan Ordoez, Carlos Carrasco-Farre, and Andres A. Aristizabal. 2022. Artificial intelligence model to predict the virality of press articles. In *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, ICMLC 2022, page 221–228, New York, NY, USA. Association for Computing Machinery.

N. Moniz and L. Torgo. 2018. Multi-source social feedback of online news feeds. *ArXiv*, abs/1801.07055.

Rikaz Rameez, Hossein A. Rahmani, and Emine Yilmaz. 2022. Viralbert: A user focused bert-based approach to virality prediction.

A Appendix

Included below is a description of the the hyperparameter tuning process.

As each hyperparameter was experimented with, all of the other potential parameters were kept at a baseline as shown in the table below.

Hyperparameter	Value
Epochs	3
Learning Rate	1e-5
Activation Function	Relu
Dropout Rate	0.5

Table 3: Default hyperparameter configuration.

A.1 Number of Epochs

Once the model had been created, all of the potential hyperparameters needed to be tested in order to determine the values that would lead to the highest accuracy of the model's predictions. The first hyperparameter that was experimented with was the number of epochs that were run during the training process. The number of epochs most directly correlates with the fit of the model, where too few epochs can underfit the model and too many epochs will overfit the model. The number of epochs and its corresponding testing accuracy can be seen in the table below.

Number of Epochs	Test Accuracy
3	74.2%
4	73.8%
5	71.9%
8	70.7%

Table 4: Experimentation with number of epochs.

The first value tested was 8 epochs, and it was noted that while during each epoch the training accuracy increased, the validation accuracy leveled out around epoch 4, and began to decrease with each subsequent epoch. This signifies that after about 4 epochs the model begins to overfit the data, and as a result the values of 3-5 epochs were tested, and it was discovered that 3 is the best number of epochs for this model.

A.2 Learning Rate

Then, the learning rate of the model had to be adjusted. Different values were tested, as shown in

the table below. It was found that 1e-5 was the best value for the learning rate.

Learning Rate	Test Accuracy
1e-4	49.2%
5e-5	49.1%
1e-5	74.2%
5e-6	71.7%
1e-6	73.4%

Table 5: Experimentation with the learning rate.

A.3 Activation Function

Next, different activation functions were tested in the final layer of the neural network. Sigmoid was found to be the best activation function, increasing the accuracy by about 2% compared to all other potential activation functions.

Activation Function	Test Accuracy
Relu	74.2%
Softmax	74.8%
Sigmoid	76.6%
Tanh	72.4%

Table 6: Experimentation with different activation functions.

A.4 Dropout Layer

A dropout layer is a simple but effective regularization tool to help prevent overfitting. Essentially, it simulates multiple possible architectures running in parallel, by probabilistically dropping nodes from the layer, forcing the model to learn in a broader way. In this case, the probability listed is the probability that any given node is kept. Generally, for hidden layers, this value is set to around 50% for best results. The model was run both with and without the dropout layer, and it was found that using the dropout layer improves model performance by about 3%.