Northwestern MSDS-458 Artificial Intelligence and Deep Learning

Assignment 3: Language Modeling with RNN

Andrew Stevens

February 20, 2022

(1) Abstract

This paper develops and tests a variety of deep learning frameworks for classification of news articles into four classes of topics. Methods of preparing the textual data will also be tested for measuring the effect on classification capability. The result of this study is that while all of the tested network types and basic data preparation perform fairly well, optimization of structure, type and hyperparameters can be performed to develop the optimal model.

(2) Introduction

The AG news topic classification dataset was assembled (Zhang, 2015) from the full AG set of over 1 million articles collected from over 2,000 news sources (Gulli). Textual analysis allows an algorithm to perform tasks against documents without a full understanding of the meaning of the words the way a human would. Natural language processing algorithms are designed to process the documents at different levels and by different means. Most of these technologies begin with tokenization to split a document (collection of text) into tokens made up of single words, several words, or parts of words. The tokens are then turned into vectors so they can be passed to a framework able to process numerical representations.

(3) Literature review

A thorough survey of Deep Learning Based Text Classification was developed by Minaee (2021). This article reviews more than 150 deep learning models, forty text classification datasets and the results of the two. Zhang (2015) developed a set of standard statistical methods and Convolutional Neural Network (CNN) structures across a set of data preparation tools including word2vec (Mikolov, 2013), lookup tables and thesaurus. The highest performing model against the AG news dataset was ngrams Term Frequency Inverse Document Frequency model (TFIDF), with an error rate of 7.64%. As shown by performance tracking on paperswithcode[1], in order to achieve higher levels of success in natural language, much more expensive and complex technologies must be made available such as BERT (Devlin, 2019).

(4) Methods

The ag_news_subset is available through tensorflow, preformatted for use by the package. Each of the 127,600 news articles is provided with a label for one of four categories of topics. Across the entire corpus, there are 95976 vocabulary words which will be preprocessed then down-selected. Since the neurons utilized by deep learning use algebraic expressions that are optimized through training, the text must be converted to a format usable by the NN. Tensorflow's TextVectorization function can take care of several pre-processing steps, facilitating the path to building a model. The function will standardize the text by lower-casing and removing punctuation, tokenize, index, and vectorize each of the documents.

Using the vectorization functions inputs, we can vary the size of the vocabulary to be used in vectorizing the article, padding any tokens that don't fall in that top number of most common tokens. We will then test a couple of methods for using masking/removal of consideration unrecognized words or common articles and other stopwords which would have (apparently) no meaningful weight towards any specific topic and are thus noise. A max_tokens value significantly below the number of actual words (we will be testing between 1000-3000, which is on the scale of 1% of the total) results in much of the documents being ignored (Figure 1).

Several of the models we will test will be of the Recurrent Neural Network type, which pass the outputs from one timestep to their input on the next timestep. This allows the network to consider

[1] https://paperswithcode.com/sota/text-classification-on-ag-news.

the order of tokens as it processes through the index. Adding a bidirectional wrapper on one of these RNNs duplicates the processing chain so it operates in both forward and reverse time (sentence) order. We will then test an LSTM, which is a specific implementation of an RNN that captures longer-term dependencies than a simple RNN. The next variation of an RNN will be a Gated recurrent units (GRU), which instead of using a memory unit as an LSTM, considers the full context (Chung, 2014). The last is a different topology, a Convolutional Neural Network where we can control the size and number of step filters that process over the tokens in the document. While RNNs are thought of as looking "across time" and CNNs "across space", both can extend to text as it is in series.

(5) Results

Considered the "Baseline" to compare other models against, a single recurrent layer LSTM without bidirectionality.

Table 1: Baseline LSTM Result

convs/ recurs	directionality	vocab size, experiment	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
1	uni	1000	656	0.36	0.8683	0.3798	0.8612	0.3977	0.8551

The RNN experimentation included varying the number of simpleRNN layers and whether one of those layers was wrapped with bidirectional, and none of these models performed better than the baseline.

Table 2: Simple RNN Results

convs/ recurs	directionality	vocab size, experiment	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
1	uni	1000	5324	0.3844	0.8641	0.4102	0.8548	0.4292	0.8472
2	uni	1000	7391	0.3797	0.8665	0.4513	0.8452	0.4623	0.8411
1	bi	1000	6002	0.3101	0.8883	0.3843	0.8578	0.4026	0.8529

The LSTM framework was then tested by varying the number of LSTM layers and wrapping them in bidirectionality. Usinig a second LSTM layer with bidirectionality was able to slightly improve performance.

Table 3: LSTM structure variation

convs/ recurs	directionality	vocab size, experiment	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
1	Bi	1000	608	0.3479	0.8722	0.382	0.8635	0.3986	0.8558
2	Bi	1000	361	0.3209	0.8813	0.3712	0.8667	0.3861	0.8613
1	uni	1000	656	0.36	0.8683	0.3798	0.8612	0.3977	0.8551
2	uni	1000	571	0.3871	0.8602	0.3948	0.8583	0.4079	0.8529

GRU networks were tested across the same design distinctions. Since the GRU, like the simpeRNN, are not GPU optimized, the training time was drastically longer. The GRU was the network design that achieved the highest accuracy on the test set by network design variation alone. They were tested on a 3000 word dictionary as that was found to have the highest performance in the experiment discussed below.

Table 4: GRU Results

convs/ recurs	directionality	vocab size, experiment	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
1	Bi	3000	16919	0.2758	0.9032	0.3176	0.8913	0.3225	0.887
2	Bi	3000	32301	0.2893	0.8989	0.3158	0.8895	0.3251	0.8884
1	uni	3000	6483	0.3301	0.8867	0.3577	0.8778	0.3652	0.8753
2	uni	3000	23993	0.2724	0.9048	0.324	0.8858	0.325	0.8876

The CNN design was tested with one or two convolution/pooling sets, and performed well, though not any better than the LSTM. I also tested adding regularization to the layer, which significantly reduced training time, but had a negative effect on performance. As CNN design can get vary complex, it’s likely that a structure exists that would perform significantly better (as seen in the literature review).

Table 5: CNN Results

convs/ recurs	regularization	vocab size, experiment	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
2	No	1000	220	0.2871	0.8966	0.4337	0.8575	0.4411	0.8554
1	No	1000	308	0.3179	0.8886	0.3992	0.8593	0.4169	0.8522
2	Yes	1000	99	0.3628	0.8738	0.4374	0.85	0.4574	0.8495

The preparation experimentation focused on how the vocabulary was revised before passing to an LSTM. A single bidirectional LSTM layer was used across all iterations. Increasing the vocabulary size alone improved performance, as this allowed a more significant percentage of meaningful vocabulary to be included. A reverse approach was then taken to test culling that vocabulary to remove less meaningful words. The most successful of these was masking.

Table 5: Vocabulary Experiment Results

Dictionary Size and Revision	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
2000	961	0.2639	0.9066	0.3398	0.8813	0.3418	0.8793
3000	552	0.2443	0.9143	0.3186	0.89	0.3223	0.8851
2934, deletion of 66 most common words	591	0.2839	0.9004	0.343	0.8783	0.3526	0.8751
2878, deletion of stop words	811	0.2503	0.9118	0.3199	0.8845	0.3221	0.8859
2000, masking=1	2553	0.2896	0.8969	0.3406	0.8767	0.3389	0.88114
3000,masking-[-1:2]	4115	0.2524	0.9134	0.3255	0.8867	0.3242	0.8871

The final experiment varied the output sequence length parameter, with both achieving higher accuracy than the baseline. This truncates the document text to the given length while being processed by the RNN (LSTM specifically)

Table 5: Output Sequence Length Results

Output sequence length	training time (s)	train loss	train acc	val loss	val acc	test loss	test acc
50	736	0.2762	0.9021	0.3379	0.8795	0.3425	0.8783
1000	1772	0.2727	0.9032	0.3355	0.8812	0.3405	0.8788

(6) Conclusions

The network design that achieved the highest accuracy was a GRU, though it was not worth the training time required. Taking the time to produce a GPU optimized model may be worth the time, plus experimenting with vocabulary variation (though compute time was not available to be performed for this study). A larger, though still relatively small to the corpus, vocabulary with “unimportant” words removed helps to improve model performance.

(7) References

1. Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015).

2. AG's corpus of news articles. (n.d.). Retrieved February 7, 2022, from http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

3. Minaee, Shervin, et al. “Deep Learning Based Text Classification: A Comprehensive Review.” ArXiv:2004.03705 [Cs, Stat], Jan. 2021. arXiv.org, http://arxiv.org/abs/2004.03705.

4. Mikolov, Tomas, et al. “Efficient Estimation of Word Representations in Vector Space.” ArXiv:1301.3781 [Cs], Sept. 2013. arXiv.org, http://arxiv.org/abs/1301.3781.

5. Papers with Code - AG News Benchmark (Text Classification). https://paperswithcode.com/sota/text-classification-on-ag-news. Accessed 21 Feb. 2022.

6. Devlin, Jacob, et al. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” ArXiv:1810.04805 [Cs], May 2019. arXiv.org, http://arxiv.org/abs/1810.04805.

7. Chung, Junyoung, et al. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” ArXiv:1412.3555 [Cs], Dec. 2014. arXiv.org, http://arxiv.org/abs/1412.3555.

(8) Figures

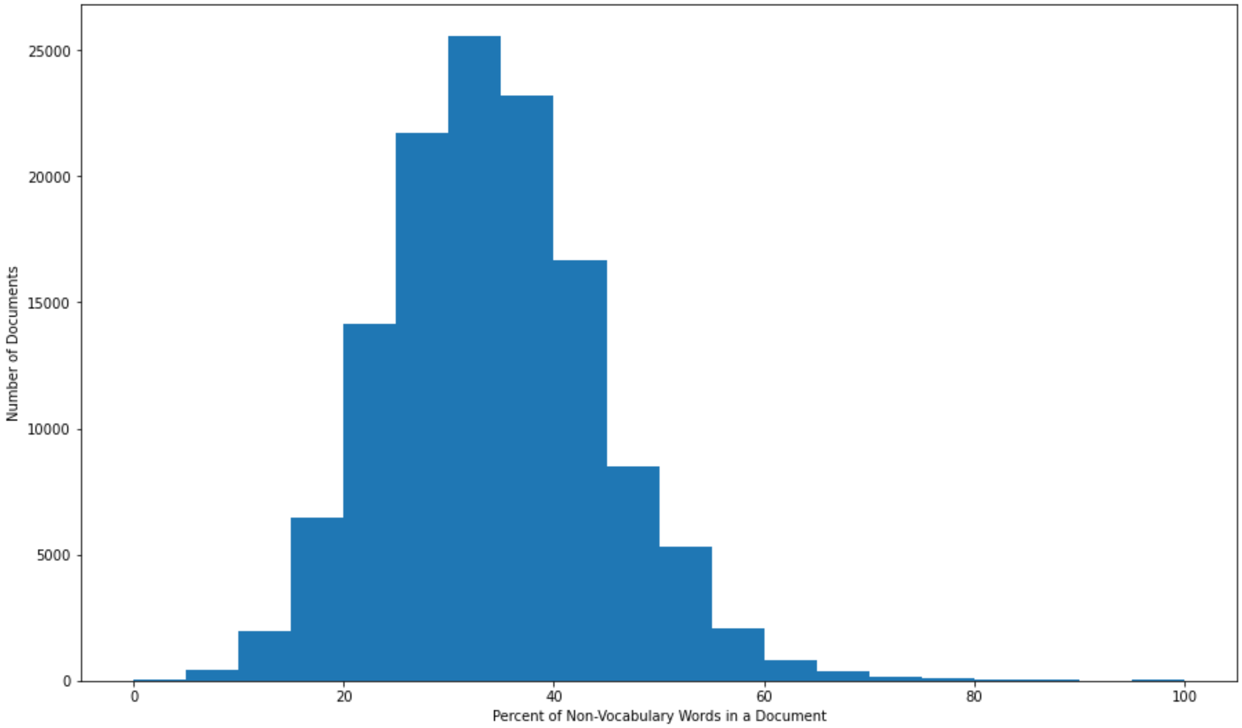

Figure 1: Histogram – Distribution of Articles by Percentage of Words not Found in 1000-word Vocabulary