On Interpretability and Feature Representations: An Analysis of the Sentiment Neuron

Jonathan Donnelly and Adam Roegiest
{jonny.donnelly, adam.roegiest}@kirasystems.com

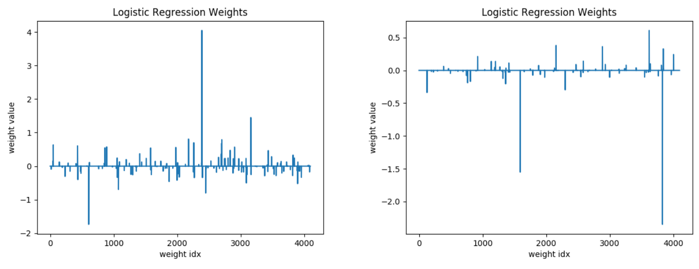
Kira Systems, Toronto, Canada

Abstract. We are concerned with investigating the apparent effectiveness of Radford et al.'s "Sentiment Neuron," [9] which they claim encapsulates sufficient knowledge to accurately predict sentiment in reviews. In our analysis of the Sentiment Neuron, we find that the removal of the neuron only marginally affects a classifier's ability to detect and label sentiment and may even improve performance. Moreover, the effectiveness of the Sentiment Neuron can be surpassed by simply using 100 random neurons as features to the same classifier. Using adversarial examples, we show that the generated representation containing the Sentiment Neuron (i.e., the final hidden cell state in a LSTM) is particularly sensitive to the end of a processed sequence. Accordingly, we find that caution needs to be applied when interpreting neuron-based feature representations and potential flaws should be addressed for real-world applicability.

1 Introduction

Several authors [13, 2, 9] have investigated the idea that single neurons or groups of neurons have easily interpretable behaviour. Recent work [6] has shown evidence that interpretable neurons do not necessarily correspond to improved neural network effectiveness and that reliance on interpretable neurons may be a sign of overfitting. To this end, we focus on Radford et al.'s [9] finding that after training a large, single layer LSTM [3] language model on $\sim 86M$ Amazon Reviews [5], a single neuron emerges as a strong predictor of sentiment, which they dub the "Sentiment Neuron" ("SN"). To examine the Sentiment Neuron's predictive capabilities, we perform an ablation analysis on the language model's features to test the impact that their removal has on classification accuracy across several datasets (Section 3). We find that the Sentiment Neuron is not necessary to achieve effective classification and that, in some cases, it can actually decrease effectiveness. Moreover, we find that randomly choosing 100 features (neurons) from the language model more often than not produces a classifier that outperforms one based on the Sentiment Neuron alone. This indicates that the Sentiment Neuron does not contain all or most of the knowledge needed for sentiment detection.

Furthermore, Radford et al.'s feature representation, where they use the final hidden cell state of the LSTM as a sequence's representation, is one of several possible valid representations. Following Howard et al. [1], we examine different methods of extracting features from an LSTM-based language model. We find that the presence of a neuron predictive of sentiment is an artefact of the network architecture regardless of how we generate features (e.g., mean-pool, final state) but its predictive power varies. In addition, a mean-pool representation appears to be more attuned to sequence length and its effects than the final state representation (Section 4). The main benefit to a mean-pool representation is that it is robust to adversarial examples (i.e., ending sequences with sentiment words); while the final state features are not (Section 5). We conclude that interpretable neurons do not guarantee success and that feature representations can and should be robust to potential adversarial cases.



(a) Radford et al. [9]. (b) Our reproduction.

Fig. 1: Plots showing linear classifier weights trained on SST with language model features from Radford et al. [9] and our reproduction.

2 Methodology

We follow the methodology of Radford et al. [9] which trains a logistic regression classifier on top of the language model features. In the default setting, these language model features are the final hidden cell state of the LSTM over a sequence. However, we vary this in the following sections to test the effectiveness of different representations. For each dataset, we shuffle and split them into three folds: 70% training set; 10% development; and, 20% test set. The development set is used to perform a light grid search of hyper-parameters for the classifier. While full cross-validation would likely yield a superior general purpose classifier, we are only concerned with examining the effect of representations and not finding the best classifier.

We evaluate this technique using four sentiment analysis datasets: Stanford Sentiment Treebank ("SST") [10]; IMDB Large Movie Review Dataset ("IMDB") [4], Rotten Tomatoes Short Movie Reviews ("MR") [8]; and Amazon Customer Review Dataset ("CR") [12]. Wang and Manning [11] provide a useful summary of these datasets. It is worth noting that the IMDB dataset consists of full length movie reviews and an order of magnitude more examples than the short reviews in the other datasets. Code to reproduce our results, as well as train a new language model from scratch, is made publicly available.[1]

[1] Source code available at github.com/kirasystems/science.

3 Neuron Ablation

We see that there is, indeed, a strongly predictive neuron when examining the weights of final state features in Figure 1. This holds for the released weights from Radford et al. [9] and our reproduction of their model. Similar to NVidia's reproduction [7], we find that the Sentiment Neuron exhibited opposite polarity from Radford et al., which indicates that such neurons are network artefacts and not just a one-off. Although, what this neuron looks like may vary between implementations.

Features	SST	MR	CR	IMDB
All features	91.76	87.52	91.38	92.28
SN Deleted	91.87	86.96	90.72	91.77
SN Only	88.52	84.52	88.33	91.46

Table 1: Accuracy on 4 datasets using all features, SN only, and SN deleted.

Using Radford et al.'s weights, we can alternatively isolate and ablate the SN from the features during training. From Table 1, isolation of the SN appears to yield decreases in effectiveness across all the datasets. When we ablate the SN, there is no substantial change from using all the features. This indicates that the inclusion of the SN does not appear to add crucial information to the classifier. Indeed, inclusion of the SN appears to hinder performance on the SST dataset. In essence, their remains enough information distributed among the other neurons to still effectively train a classifier.

Based upon the ablation results, we might wonder whether any other neurons would hinder classification accuracy. Accordingly, we ablate each individual neuron and train a new classifier in turn for all neurons on the SST and IMDB datasets. Figure 2 reports the results for SST as the IMDB results are similar. It does appear to be the case that some neurons do hinder performance. The neuron which when ablated gives the highest accuracy (92.15%), yields a classifier that is only slightly better than chance when used in isolation. Thus, it appears to be the case that some features do not contribute to sentiment detection.

We can also examine the performance of classifiers trained on each neuron in isolation for SST and IMDB. Figure 3 shows that there exist neurons that rival the performance of the SN in predicting sentiment. Moreover, there are many neurons that do not appear to be good predictors of sentiment. Interestingly, ablating multiple of the "hindrance" neurons (i.e., those that produce a better classifier when ablated) or the less predictive neurons does not yield substantial improvements in the resulting classifier. It would appear that the full feature classifier is able to "learn around" these features.

Based upon our findings thus far, we might wonder whether or not there is a necessary "critical mass" of neurons that are needed to produce an effective classifier. In Figure 4, we cumulatively ablate each neuron in order from highest to lowest corresponding logistic classifier weight [2] on the SST dataset. We see that

[2] These weights are generated from training the classifier on all available neurons.

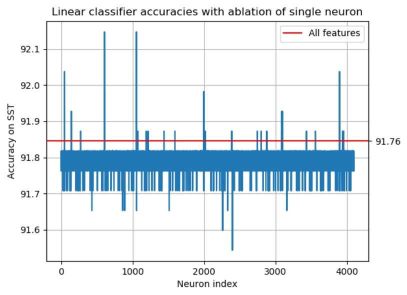


Fig. 2: Accuracy scores of classifier with each neuron ablated individually.

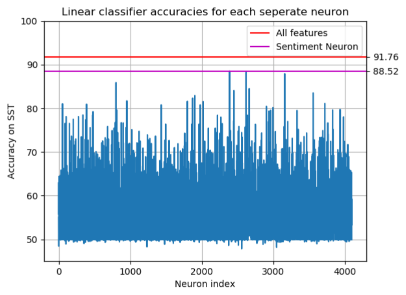


Fig. 3: Accuracy scores with linear classifier trained on each neuron individually.

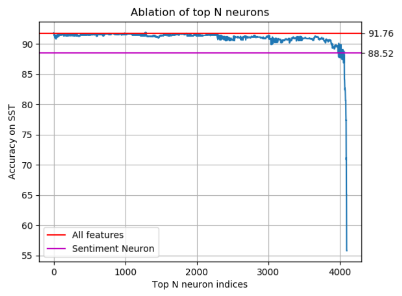


Fig. 4: Accuracy on SST as neurons are cumulatively ablated.

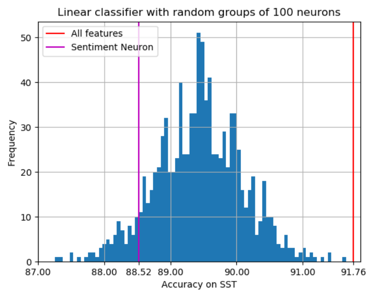


Fig. 5: Histogram showing the accuracy on 1000 random samples of 100 neurons.

with ∼100 neurons remaining we achieve parity with the SN before precipitously dropping. Accordingly, if the SN is our barometer for a "good" or "sufficient" classifier then we need at least 100 neurons to rival its effectiveness. We then proceed to train 1000 different classifiers using 100 randomly selected neurons. Figure 5 shows that the resulting accuracy scores form an approximately normal distribution with the SN's score coinciding with the lower tail. Accordingly, selecting a random group of neurons will more often than not yield a superior classifier to the SN. This leads us to conclude that, contrary to Radford et al. [9], there is a meaningful amount of sentiment information stored outside of the SN.

4 Features

Up until now, we have examined the feature representation using the final hidden cell state of the LSTM but this could allow the end of a sequence to have undue influence on the feature weights. While this may matter less for short pieces of text, longer pieces of text may not be ideally represented as there would be a skew towards the end of the sequence. Accordingly, we examine different ways of integrating information across time-steps including the mean-pool, min-pool, max-pool, and absolute-pool of values. For the sake of brevity, we report the mean-pool results only as they provide the most interesting counterpoint to the final state features. Similar to Howard et al. [1], we report the concatenation of final state and mean-pool representations as such a representation may offer the advantages of both representations.

Features	SST	MR	CR	IMDB
Final	91.76	87.52	91.38	92.28
Mean	88.03	82.97	89.52	92.82
concat(Final, Mean)	91.76	87.43	90.45	93.86

Table 2: Accuracy scores on 4 sentiment datasets using final, mean and concatenated state features.

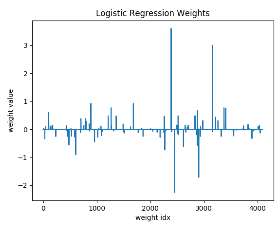


Fig. 6: Logistic regression weights for SST-model using mean-pool features.

Table 2 reports the accuracy scores across our four datasets for the different possible feature representations. Generally, the concatenated features generally perform as good as one of the constituent feature representations and can sometimes outperform both. Interestingly, we do not see as much degradation of the final state features in the longer IMDB text than we had expected but this may be due to repeated use of sentiment throughout a review. The generally inferior performance of the mean-pool features is not readily apparent but we posit that the shorter reviews did not allow the mean-pool features to stabilize on a good representation. Such a hypothesis warrants further investigation but the competitive performance of the mean-pool features on IMDB provides some evidence that this length may play a role. Further, if we compare the weights of the mean-pool neurons (Figure 6) to the final state ones (Figure 1), we see that the mean-pool features have a greater number of influential neurons (neurons with larger corresponding weights). This increase in influential neurons may then correspond to requiring longer sequences to make accurate classifications.

In spite of these differences, there are a class of examples that neither the final state nor mean-pool features are able to classify. Across the the four datasets, the overlap of incorrect examples ranges between 30-40% which appears to indicate that there are aspects of sentiment that neither of these feature representations capture. We note that manual examination of a selection of these incorrectly identified examples reveals that these express less obvious sentiment (e.g., *"What you would end up with if you took Orwell, Bradbury, Kafka, George Lucas and the Wachowski Brothers and threw them into a blender."*). While such examples are not unexpected, they highlight that Radford et al.'s network is unable to capture all nuances of sentiment.

5 Adversarial Examples

As suggested in the previous section, the final state features may be subject to undue influence of the ending of a sequence. We test this by adversarially adding a positive ("Wonderful") and negative ("Terrible") sentiment word to test examples and examine how this affects classification accuracy. As seen in Table 3, the effectiveness of the final state and concatenated features are substantially affected by the inclusion of a trailing sentiment word for small datasets; while the mean-pool features are robust to their inclusion. This lends credence to our hypothesis that final state features are unduly influenced by most recently processed input, whereas the mean-pool features are more capable of capturing a sequence's overall sentiment.

This trend does not hold for longer text since adding a single word does not appear to hamper the final state (or concatenated) features from achieving good scores on IMDB with only a single word added (i.e., the IMDB-1 results). However, the IMDB reviews are approximately 10 times longer than the smaller reviews, which means we may need to add a similar proportion of sentiment words to the end to achieve a similar adversarial effect. The results of the IMDB-10 setting bear this out and this indicates the choice of feature representation should take into account all possible use cases. It is not inconceivable that in the real-world such adversarial examples would occur, especially in short form communication (e.g., Twitter).

Sentiment	Features	SST	MR	CR	IMDB-1	IMDB-10
Positive	Final	52.17	57.22	77.06	91.16	50.30
	Mean	89.13	82.88	90.05	92.87	89.73
	concat(Final, Mean)	53.93	61.07	79.97	93.50	61.25
Negative	Final	72.16	59.52	48.01	90.69	53.17
	Mean	89.02	83.63	90.98	92.91	92.68
	concat(Final, Mean)	73.31	57.74	42.44	93.07	81.29

Table 3: Accuracy scores on 4 sentiment datasets when positive and negative words are appended to the test set examples. IMDB-1 denotes a single sentiment word added and IMDB-10 denotes 10 words added to the end of test examples.

6 Conclusion

Radford et al. [9] found that in a large LSTM-based language model, there is a single neuron that is predictive of sentiment. After conducting an ablation study, we find that this neuron is not necessary to achieve effective classification and that, in some cases, hinders effectiveness. Moreover, we find that the effectiveness of the Sentiment Neuron can be matched or exceeded, more often than not, by randomly selecting 100 neurons to train a classifier. This indicates that the reliance on a single "understandable" neuron may result in undue bias in the resulting classifier, which is in accord with the findings of Marcos et al. [6]. Additionally, we find that the feature representation is important and that relying on the final hidden cell state opens an end user up to exploitation by adversarial examples. Using the mean-pool of cell states across a sequence appears to be robust to this type of exploitation but is generally inferior to the final state features. Accordingly more work is necessary to find a robust but highly effective representation.

References

1. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2018)
2. Karpathy, A., Johnson, J., Li, F.: Visualizing and understanding recurrent networks. International Conference on Learning Representations (2016)
3. Krause, B., Lu, L., Murray, I., Renals, S.: Multiplicative LSTM for sequence modelling. International Conference on Learning Representations (2017)
4. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 142–150. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
5. McAuley, J.: Amazon product data. http://jmcauley.ucsd.edu/data/amazon/
6. Morcos, A.S., Barrett, D.G., Rabinowitz, N.C., Botvinick, M.: On the importance of single directions for generalization. In: International Conference on Learning Representations (2018)
7. Nvidia: Sentiment discovery. GitHub repository (2017)
8. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of ACL. pp. 115–124 (2005)
9. Radford, A., Józefowicz, R., Sutskever, I.: Learning to generate reviews and discovering sentiment (2017), http://arxiv.org/abs/1704.01444
10. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA (October 2013)
11. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2. pp. 90–94. ACL '12 (2012)
12. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation **1** (2005)
13. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. International Conference on Learning Representations (2014)