Interpreting Deep Models for Text Analysis via Optimization and Regularization Methods

Hao Yuan
Washington State University
hao.yuan@wsu.edu

Yongjun Chen
Washington State University
yongjun.chen@wsu.edu

Xia Hu
Texas A&M University
hu@cse.tamu.edu

Shuiwang Ji
Texas A&M University
sji@tamu.edu

Abstract

Interpreting deep neural networks is of great importance to understand and verify deep models for natural language processing (NLP) tasks. However, most existing approaches only focus on improving the performance of models but ignore their interpretability. In this work, we propose an approach to investigate the meaning of hidden neurons of the convolutional neural network (CNN) models. We first employ saliency map and optimization techniques to approximate the detected information of hidden neurons from input sentences. Then we develop regularization terms and explore words in vocabulary to interpret such detected information. Experimental results demonstrate that our approach can identify meaningful and reasonable interpretations for hidden spatial locations. Additionally, we show that our approach can describe the decision procedure of deep NLP models.

Introduction

Deep neural networks have shown great success in many NLP tasks, such as sentence classification (Kim 2014; Zhang, Zhao, and LeCun 2015), natural language generation (Yu et al. 2017; Lin et al. 2017), machine translation (Vaswani et al. 2017; Gehring et al. 2016) and visual question answering (Wang and Ji 2018). Most existing approaches treat deep neural networks as black-boxes and only focus on the performance. Without understanding the working mechanisms of neural networks, deep models cannot be fully trusted, since we do not know how and why decisions are made. However, due to the complex structures of deep neural networks, it is challenging to interpret deep models and their behaviors, especially for NLP tasks that deal with discrete data.

Most existing approaches for interpreting NLP models only investigate the relationships between input sentences and output decisions to explore which input words are more important to make decisions (Lei, Barzilay, and Jaakkola 2016; Li et al. 2015). However, the inner workings of networks should also be studied to answer important questions regarding hidden layers, such as which hidden units are more important for a decision and why they are important. To the best of our knowledge, there are no related studies focusing on the interpretation of hidden neurons of NLP models.

In this paper, we propose an approach to interpret and understand deep NLP models. Specifically, we focus on convolutional neural networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012) for sentence classification tasks. Our approach employs gradient-based approaches (Simonyan, Vedaldi, and Zisserman 2013) and optimization techniques (Erhan et al. 2009) to select spatial locations with high contribution to the decision from hidden layers and study what is detected by these locations. We propose to approximately interpret the meaning of detected information using the nearest neighbors of the optimized representation based on the special property of word representations, which imply that words with semantically similar meanings are embedded to nearby points (Mikolov et al. 2013). Experimental results demonstrate that our approach can obtain reasonable and meaningful interpretation for hidden units. It is shown that our approach can explain the decision process in NLP models.

Background and Related Work

Most of the existing interpretation approaches are proposed to investigate deep models in computer vision rather than the NLP area. The saliency map techniques study which input pixels are more important to the final decision (Simonyan, Vedaldi, and Zisserman 2013; Du, Liu, and Hu 2018; Du et al. 2018). The importance of different pixels can be approximated by the gradient of output score with respect to the inputs (Zeiler and Fergus 2014; Springenberg et al. 2014; Mordvintsev, Olah, and Tyka 2015). The similar idea was applied to NLP models to study which input words contribute more to the prediction (Li et al. 2015). However, such techniques only provide word-level interpretation while different words are highly correlated to convey a meaning.

In addition, several approaches focus on feature visualization, which investigates what pattern the hidden neurons of a model try to detect (Olah, Mordvintsev, and Schubert 2017; Erhan et al. 2009; Nguyen et al. 2016; Mahendran and Vedaldi 2015; Nguyen, Yosinski, and Clune 2015). Optimization techniques are commonly used for such purposes. The key idea is to iteratively update a randomly initialized input to investigate a specific behavior in hidden lay-

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ers, such as maximizing the activation values of neurons or maximizing the score of a class. The optimized input can then be visualized as abstracted images to reflect the meaning. However, such a technique cannot be directly applied to NLP models since word representations are discrete and the meaning cannot be abstracted. Thus the optimized input is difficult to interpret. By combining the above two techniques, (Olah et al. 2018) investigate the meaning of hidden layers to interpret models for image classification tasks. However, as we mentioned above, the optimized input is a sequence of abstract vector representations and cannot be visualized as abstracted texts. We propose an approach to approximately interpret the high-level meaning of the optimized input by selecting the neighbors of these vector representations from the embedding space.

Methods

As discussed above, it is not enough to only build saliency maps on input sentences to visualize word-level interpretation, since different words may combine together to convey a meaning. In addition, without investigating the hidden layers, we still do not understand how the hidden neurons work, and neural networks remain a black box. To better understand deep NLP models, we propose an approach to focus on the contribution and meaning of hidden neurons, thereby allowing us to visually interpret the decision process.

Visual Interpretation of Hidden Units

In this work, we investigate the interpretation of CNN models for sentence classification tasks in NLP. The general structure of CNN models we study is shown in Figure 1. Given an input sentence, it first passes through an embedding layer and several convolutional layers. Then it is fed into a max-pooling layer and a fully-connected layer with softmax function to make predictions.

Intuitively, we wish to investigate the hidden units of a deep NLP model so that we can answer three questions; those are, which hidden spatial locations are more important to decisions? what is detected by these spatial locations from input sentences? and what is the meaning of the detected information? However, there are two main challenges for answering these questions; those are, how to explore what is detected by hidden units? and how to interpret the detected information? Existing approaches in computer vision cannot be directly applied since word representations are discrete from each other and cannot be abstracted as images.

We first combine the idea of saliency map and optimization to answer the question of what is detected by hidden units. Based on the property of word representations, we propose to approximately interpret the meaning of detected information using the nearest neighbors of the optimized representation. Then we develop regularization terms to help interpretation. Generally speaking, the interpretation procedure consists of three main steps. First, we employ gradient-based approaches to estimate the contributions of different spatial locations in a hidden layer. Based on the magnitude of contributions, the spatial locations are sorted, and those with high contribution are selected to be interpreted in the following steps. Second, to obtain what is detected by different spatial locations in hidden layers, we iteratively update a randomly initialized input via optimization. Finally, the optimized input is a sequence of numerical vectors but such abstract values are hard to interpret. Based on the property of word representations that words with semantically similar meanings are embedded to nearby points, we design regularization terms to encourage different vectors in the optimized input to be similar to each other. Then we explore the nearest neighbors (Altman 1992) in term of cosine similarity to approximately represent the meaning of the target spatial location. The general logic flow of our approach is illustrated in Figure 1.

Saliency Maps for Hidden Units

Since there are a large number of neurons in hidden layers, it is not possible to interpret each neuron. Hence we employ saliency map techniques to select spatial locations with high contributions for further interpretation. The saliency map acts like a heatmap, where saliency scores are estimated by the first order derivatives and reflects the contribution of different neurons. While most of existing approaches build saliency maps to explore the contribution of individual words in input sentences, we study the importance of different hidden spatial locations instead.

Formally, for an input sentence X, the model predicts that it belongs to class c and produces a class score S_c. Let a_{ij} represents the activation vector of the spatial location i of layer j, and its dimension is equal to the number of channels. Also let A_j denotes the activations of layer j, which is a matrix, where each column corresponds to one spatial location. The relationship between the score S_c and A_j is highly non-linear due to the non-linear functions in deep neural networks. Inspired by the strategy in recent work (Li et al. 2015; Simonyan, Vedaldi, and Zisserman 2013), we compute the first-order Taylor expansion as a linear function to approximate the relationship as

$$S_c \approx \mathrm{Tr}(w(A_j)^T A_j) + b, \tag{1}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix and $w(A_j)$ is the gradient of class score S_c with respect to the layer j. Such gradient can be obtained by using the first order derivative of S_c with respect to the layer A_j as

$$w(A_j) = \frac{\partial S_c}{\partial A_j}. \tag{2}$$

For the spatial location i in the layer j, the gradient of S_c with respect to this spatial location is the i^{th} column of $w(A_j)$, denoted as $w(A_j)_i$. Then the saliency score of this location $\mathrm{Score}_c(X)_{i,j}$ is calculated using linear approximation:

$$\mathrm{Score}_c(X)_{i,j} = w(A_j)_i \cdot a_{ij}, \tag{3}$$

where $\cdot$ refers to the dot product of vectors.

It is noteworthy that we do not directly use gradients as saliency scores. The reason is that gradients only reflect the sensitivity of the class score when there is a small change in the corresponding spatial location. The employed linear approximation incorporates the activation values to measure

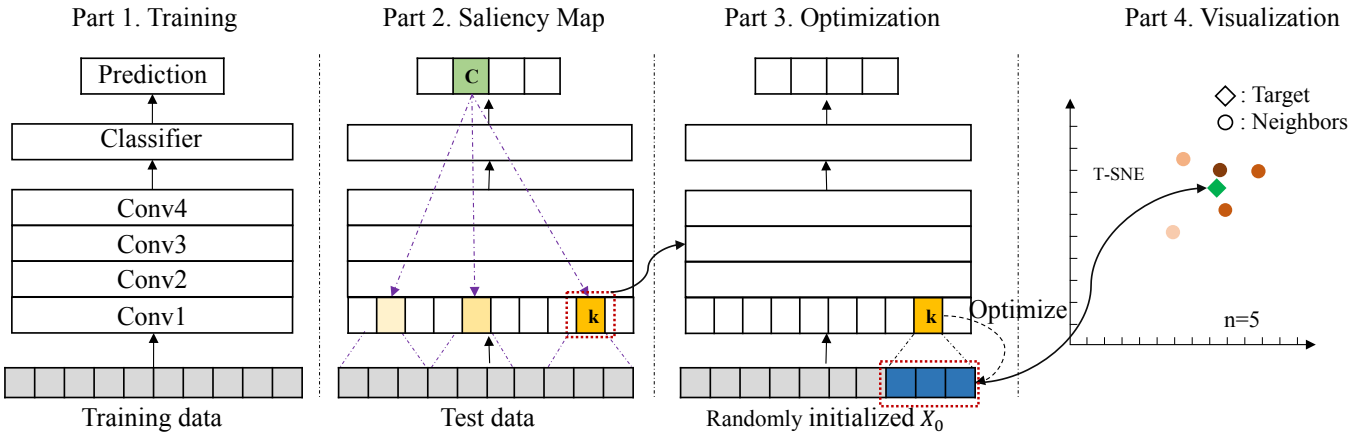

Figure 1: Illustration of the overall pipeline of our approach. Part 1 shows the general structure of the CNN model that we try to investigate. After training, we first build saliency maps for different hidden spatial locations, where saliency scores reflect contributions to the final decision. As the example shown in Part 2, the CNN model classifies the test sentence to class c (shown in green). For the conv1 layer, the saliency score is computed for each spatial location, and three spatial locations are selected (highlighted in yellow). Next, for each selected location, optimization technique is employed to determine what is detected from the test sentence. As shown in Part 3, for the spatial location k, a randomly initialized input X_0 is fed to the network and we iteratively update X_0 towards the objective function shown in Equation 6. Finally, based on the receptive field of location k (shown in blue with red bounding box), we obtain an overall representation for this receptive field. We search the vocabulary and select word representations with high similarity to the overall representation. Then, the t-SNE is employed to visualize these representations, as shown in Part 4.

how much one spatial location contributes to the final class score. In addition, after training, the weights and parameters in the model are fixed so that the gradient of S_c with respect to a specific spatial location is fixed and does not depend on the input. By using the linear approximation, the saliency score becomes input-dependent.

Input Generation via Optimization

By employing the saliency map technique, we can select spatial locations with high influence on the final decision. However, it is still not clear why they are important. In order to explore this direction, we propose to use optimization techniques to understand what is detected from the input sentence by these spatial locations. The key idea of optimization techniques for interpretation is to iteratively update a randomly initialized input towards an objective function. Such optimization procedure is similar to the training of deep neural networks. The main difference is that in such optimization techniques, the parameters of the networks are fixed but the input is optimized. When maximizing the activation value of a certain neuron, the optimized input reflects the pattern that this neuron tries to detect (Zeiler and Fergus 2014; Erhan et al. 2009). The activation value of each neuron shows the strength of the pattern detected from inputs. For the neuron k in the spatial location i of hidden layer j, we can obtain an optimized input $\overline{X}_{ijk}$ and the activation value a_{ijk}. When considering spatial locations as a whole, what is detected can be approximated using a weighted sum of $\overline{X}_{ijk}$ and a_{ijk} as

$$\overline{X}_{ij} = \sum_{k=1}^{n} a_{ijk} \overline{X}_{ijk}, \tag{4}$$

where n is the number of neurons in the spatial location i of layer j, which is equal to the number of channels.

Such approximations are not efficient since the number of channels can be large, and we need to obtain an optimized input for each neuron. Furthermore, it is challenging to add regularization since the optimized input is generated for each neuron separately. Hence, we propose to incorporate the activation vector of a spatial location and optimize the input for the whole spatial location. Formally, for a spatial location i of layer j, let a_{ij} represents its activation vector given the input sentence X. We randomly initialize another input X_0 and feed it to the network. For the same spatial location, we obtain another activation vector a'_{ij}. Then we iteratively update the input X_0 towards the following objective function:

$$\max a_{ij} \cdot a'_{ij}, \tag{5}$$

where $\cdot$ refers to the dot product of vectors.

Regularization

In Equation 5, there is no regularization term for optimization. However, without any regularization, the updating procedure will not converge since the input X_0 can be updated without any constraint, and the target $a_{ij} \cdot a'_{ij}$ keeps increasing. Hence, we add L_2 regularization to the objective function. In addition, in order to interpret the optimized input, we propose to add another regularization term, known as the similarity regularization, to make the optimized inputs readily interpretable.

Formally, let $\widehat{X_0}$ denotes the receptive field of the spatial location we try to investigate, and l and r are the leftmost and rightmost corresponding indices in $\widehat{X_0}$. Then we

have $\widehat{X_0} = [x_{0l}, \cdots, x_{0i}, \cdots, x_{0r}]$, where x_{0i} denotes the i^{th} column of X_0. By adding the regularization terms, the objective function becomes

$$\max a_{ij} \cdot a'_{ij} - \lambda_1 \left\| \widehat{X_0} \right\|_2^2 + \lambda_2 Sim(\widehat{X_0}), \quad (6)$$

where $\cdot$ denotes the dot product of vectors, $Sim(\cdot)$ is the similarity term, and λ_1, λ_2 are regularization parameters.

L_2 **Term**: By adding the $L2$ regularization, the optimization procedure converges much faster. Furthermore, the $L2$ term encourages features with high contributions to the target $a_{ij} \cdot a'_{ij}$ to increase more than others. This is beneficial, since features of high importance can better represent the meaning of hidden spatial locations.

Similarity Term: Intuitively, we try to assign each spatial location an estimated meaning to represent what is detected from the input sentence. After optimization, we obtain multiple vector representations. However, such representations may be very different from each other. In this case, it is challenging to find an overall representation for them. Based on the property of word representation that words with semantically similar meanings localize closer in the embedding space (Mikolov et al. 2013; Li et al. 2015), we propose the similarity regularization for optimization, which encourages different vector representations in optimized X_0 to be similar to each other. In this way, these vector representations are encouraged to have similar semantically meanings when mapping back to the word space. Formally, the similarity term is defined as

$$Sim(\widehat{X_0}) = \frac{1}{N} \sum_{\forall i,j} \frac{x_{0i}}{\|x_{0i}\|_2} \cdot \frac{x_{0j}}{\|x_{0j}\|_2}, \quad (7)$$

where $\cdot$ refers to dot product of vectors, $N = r - l + 1$ and $i, j \in [l, r]$.

Visualization of Optimized Inputs

By combining saliency maps and optimization, we know which spatial locations in hidden layers contribute most to the final decision. We also obtain an optimized input for each selected hidden spatial location to represent what is detected by this location. However, the optimized input consists of several numerical vectors and is still hard to interpret. It is challenging because words representations are discrete so that the optimized representations cannot be mapped to words directly. We propose to find representative words whose vector representations have high cosine similarity with the optimized input as an estimation of the meaning.

Given an optimized input X_0, based on the spatial location we can obtain its receptive field with respect to X_0, denoted as $\widehat{X_0} = [x_{0l}, \cdots, x_{0i}, \cdots, x_{0r}]$. Since we employ the similarity regularization term, different representations x_{0i} are encouraged to be similar. Additionally, in the case of word embedding, similar representations lead to similar semantic meanings. Hence, it is reasonable to take an average of these representations as an overall approximation as

$$x_{overall} = \frac{1}{N} \sum_{i=l}^{r} x_{0i}. \quad (8)$$

Dataset	c	N_{train}	N_{test}	$\|V\|$
MR	2	9596	1066	18160
AG's News	4	120000	7600	84252

Table 1: The summary statistics of the MR dataset and the AG's News dataset. In the table, c represents the number of classes, N_{train} denotes the number of training examples in the dataset, N_{test} is the number of test examples, and $|V|$ denotes the size of vocabulary.

It is impossible to find the exact meaning for $x_{overall}$. Instead, we study the neighbors of $x_{overall}$ in the embedding space. We believe the neighbors share similar high-level semantic meaning with $x_{overall}$. Specifically, we compare $x_{overall}$ with different word representations in the vocabulary using cosine similarity and obtain the top words and their corresponding representations. By studying the semantic meaning of these neighbors, we can understand the high-level meaning of the detected information by this spatial location. Finally, these representations can be visualized in the 2D space via dimension reduction techniques, such as t-SNE (Maaten and Hinton 2008) and principal component analysis (Wold, Esbensen, and Geladi 1987).

Experimental Studies

To demonstrate the effectiveness of our approach, we evaluate our methods both quantitatively and qualitatively. We first introduce two datasets we are using and the setup of the experiments in detail. Next, we report the interpretation results for several sentence examples. Finally, we present the quantitative evaluations of our methods.

Datasets

We conduct experiments to show the effectiveness of our approach based on two NLP datasets; namely the MR dataset and AG's News dataset. We report the summary statistics of these two datasets in Table 1.

MR Dataset: The MR dataset[1] contains movie review data for sentiment analysis. Each sample in the dataset is a one-sentence movie review and labeled with "positive" or "negative".

AG's News Dataset: The AG's News dataset[2] is constructed from AG's corpus of news articles. The dataset contains the largest 4 classes of news in the original AG's corpus, where only the title and description are used (Zhang, Zhao, and LeCun 2015). The label of each news example depends on the topic of the news, which can be "World", "Sports", "Business" or "Sci/Tech". Each class has 30,000 training examples and 1,900 testing examples.

Experimental Setup

In this section, we introduce the CNN model that we investigate in this work. We then discuss the interpretation setup

[1] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

	MR	AG's News
Length	56	195
Conv num	3	4
Kernel size	5	5
Conv channel	128, 64, 32	512,256,128,64
Activation	Relu	Relu
Embedding	300	300
Pre-train	Word2vec	Word2vec
Learning rate	2e-4	5e-4
Batch size	128	64

Table 2: The CNN models we used for the MR dataset and AG's News dataset. Different columns refer to the network settings for different dataset. *Length*: the length of input sentence; *Conv num*: the number of 1D convolutional layers in the model; *Conv channel*: the number of channels for convolutional layers; *Activation*: activation function in convolutional layers; *Embedding*: dimension of word embedding; *Pre-train*: the type of pre-trained word embedding employed.

in detail. Finally, we discuss the preprocessing procedure for text inputs.

CNN Model: We build CNN models for both datasets, and the overall structures are shown in Part 1 of Figure 1. The input sentence is padded to the same length and fed into the embedding layer, where the word2vec word embedding is employed (Mikolov et al. 2013). Then several 1D convolutional layers (LeCun et al. 1998) and a max-pooling layer are applied. Finally, a fully-connected layer with the softmax function produces the predictive decision. Detailed descriptions of models are given in Table 2.

Interpretation: After training, the parameters and vocabulary in CNN models are saved for interpretation. These trained parameters in CNN models are reused and fixed during the interpretation procedure. Given a test sentence, the saliency map technique returns the top m spatial locations for a hidden layer. We set m equal to 3 in our experiments and focus on the first hidden layer. The input in optimization is randomly initialized using the Xavier initialization method (Glorot and Bengio 2010). For the MR dataset, the regularization parameters are set as $\lambda_1 = 0.004$ and $\lambda_2 = 0.02$. For the AG's News dataset, we set $\lambda_1 = 0.002$ and $\lambda_2 = 0.01$. We implement our approach using TensorFlow and conduct our experiments on one Tesla K80 GPU. The learning rate in optimization procedure is set to $2 \times e^{-4}$ and we apply the Adam optimizer (Kingma and Ba 2014) with momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Preprocessing: The way to preprocess the text data is similar to the existing NLP application (Kim 2014). It is noteworthy that we do not convert words to lower case since the meaning of a word is case-sensitive.

Visual Interpretation Results

We first report the prediction accuracy of the CNN models that we try to interpret. The results are shown in Table 3. The CNN models we build can achieve competitive or even better performance compared with the baseline CNNs (Kim 2014; Zhang, Zhao, and LeCun 2015). The reason why we conduct such comparison is that we wish to show the CNNs we investigate are models with reasonable performance. Next, we present the visual interpretation results to demonstrate the effectiveness of our approach.

Dataset	MR	AG's News
Our CNN model	79.96%	92.05%
Baseline CNN model	81.50%	91.45%

Table 3: Comparison of prediction accuracy between the CNN models we build and the baseline CNNs.

MR Dataset: For the MR dataset, we show the visualization results for two testing examples; those are, *"As a good old fashioned adventure for kids spirit stallion of the cimarron is a winner"*; and *"Plays like one of those conversations that comic book guy on the simpsons has"*. Clearly, the first example is a positive review while the second one is a negative one. Both of them are correctly classified by the CNN models.

The visual interpretation result of the first example is shown in Figure 2. As demonstrated, three spatial locations (grids in red, blue and green) of the first convolutional layer are selected based on their saliency scores. The bounding boxes reflect the receptive fields of these spatial locations with respect to the input. The receptive fields contain words like "good", "fashioned", "adventure", and "spirit", which are commonly used in positive movie reviews. In addition, the top part of Figure 2 shows the visual interpretation for selected hidden spatial locations. Most of the neighbors identified by our approaches are positive adjectives, such as "unflinching", "ok", "smartly", and "gritty". We use these neighbors to represent the meanings of hidden spatial locations so that the locations should be interpreted as positive meaning. This is consistent with their receptive fields and the final positive prediction. Such interpretation helps us understand how the decision is made; that is, the information detected by these spatial locations is positive and these spatial locations have high contribution to the final decision so that the final prediction is positive.

In addition, we show the visualization result of the second MR example in Figure 3. Clearly, many words with negative meanings are selected to interpret the meaning of hidden spatial locations, such as "terribly", "awkward", "devoid", "unwatchable" and "brainless". Hence, these spatial locations can be interpreted as negative meaning, and it is consistent with the prediction. We also observe that most of neighbors are adjectives or adverbs.

AG's News Dataset: Similarly, we show the interpretation results for two examples from the AG's News dataset. Both of them are correctly classified by CNN models.

The first example with label "sports" is *"Looking at his ridiculously developed upper body, with huge biceps and hardly an ounce of fat, it's easy to see why Ty Law, arguably the best cornerback in football, chooses physical play over finesse. That's not to imply that he's lacking a finesse"*. As shown in Figure 4, several nouns are selected to interpret the hidden locations. Most of them are highly related to the topic "sports", for example, "Toni", "Elarton" and "Fahrenhorst"

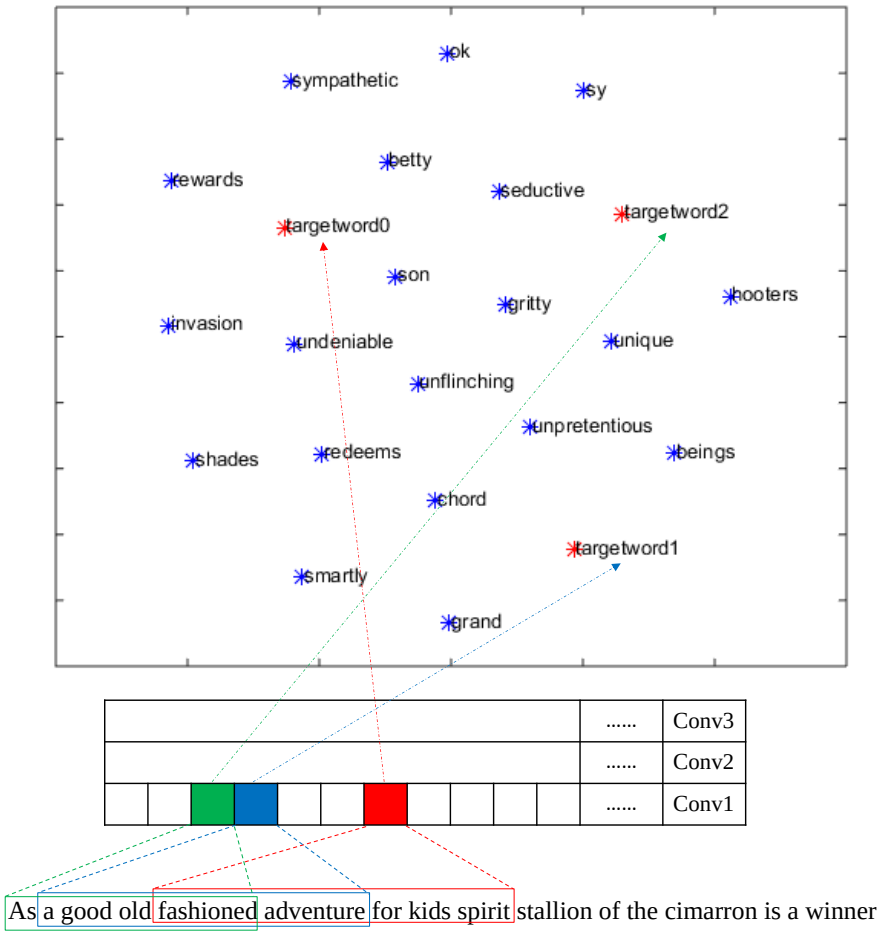

Figure 2: The visualization interpretation result for the first example for the MR dataset. The middle part of the figure shows the contribution of different spatial locations in hidden layers, where red color means highest contribution to the final decision; blue color refers to the second highest contribution; and green means the third highest contribution. The bounding boxes in different colors correspond to the receptive field of different spatial location. The top part shows the t-SNE visualization of the interpretation obtained by our approach. The interpretations of target spatial locations are marked as "targetword" and connected to the corresponding spatial locations by dash lines.

are names of famous players; "Toulouse " and "Newcastle" are names of famous sports teams. One may argue that such names can be used in many areas and are not limited in topic "sports". We claim that our interpretation results are based on the model and datasets where there are only four classes: "World", "Sports", "Business" and "Sci/Tech". When only considering these four types of news, these names are highly related to "sports". Hence, we believe the selected words are reasonable and consistent with the prediction.

The second example is *"Jet Propulsion Lab – Scientists have discovered irregular lumps beneath the icy surface of Jupiter's largest moon, Ganymede"*. Obviously, it belongs to topic "Sci/Tech". The interpretation result is shown in Figure 5. Similarly, the word selected by our approaches are highly related to "Sci/Tech" topic, such as "Solar", "protorosaur", "datacenter" and "Scientistcom".

In addition, it is interesting that for the MR dataset, the interpretation results are mostly adjectives and adverbs while the results of AG's News data contains more nouns. This is reasonable since in movie review, the positive or negative meaning is mostly expressed by adjectives and adverbs while the topic of a news is highly related to nouns. Such observation also demonstrates that our approach provides reasonable interpretation based on the model and dataset. In conclusion, the words selected by our approach to interpret the hidden locations are meaningful and reasonable. They interpret the information detected from the input sentence. In addition, such interpretations help explain how the decision is made and why the decision is made.

Evaluation of Interpretability

Intuitively, if the interpretations of hidden spatial locations are meaningful and reasonable, the hidden layers should convey similar high-level meaning compared with the original input sentence. In this section, we explore if the interpretations generated by our approach are reasonable. We first

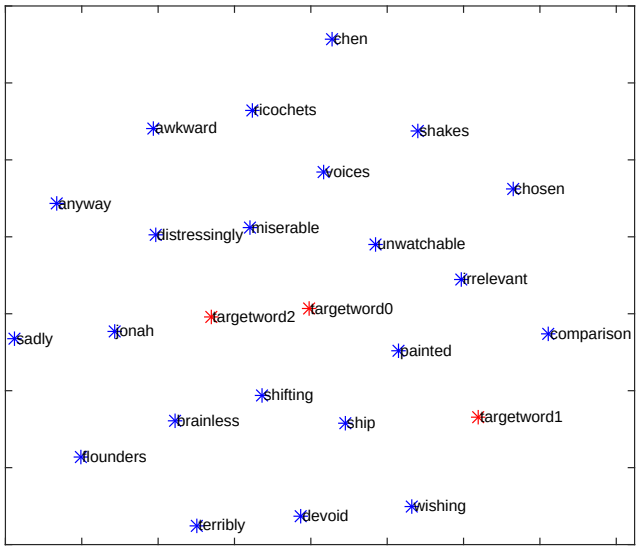

Figure 3: The visualization interpretation result for the second example for the MR dataset. Only the final result is presented due to space constraints.

Dataset	MR	AG's News
Matching rate	0.934	0.843

Table 4: The matching rates for the MR dataset and AG's News dataset.

introduce how we quantitatively evaluate the interpretation. Given an input sentence X, the model classifies it to class c. We obtain the interpretation of k locations with the highest contribution to the decision. For each location we use the m nearest neighbors to represent its meaning. In this way, for each input, we have a sentence with km words to interpret the hidden layer, denoted as X' . If we feed X' to the same network, we obtain another classification result c'. If c is equal to c', we call it a matching, and it means the interpretation of hidden locations shares similar high-level meaning with the input. Here, we focus on the first hidden layer and set $k = 3$ and $m = 5$, .

We conduct such evaluation for the two datasets and the results are reported in Table 4. It is obvious that for both datasets, our method provides reasonable interpretation for most examples. In addition, our approach has better performance on the MR dataset. We believe the reason is that the length of input examples in the AG's News is much greater than that of MR data. Then it is more challenging to use the interpretation of three locations to represent the meaning of whole input sentences.

Conclusion

Investigating hidden units in neural networks are of great importance to understand their working mechanisms. However, most current approaches focus on models for images tasks. It is challenging to understand the meaning of hidden units in NLP models, since word representations are discrete and

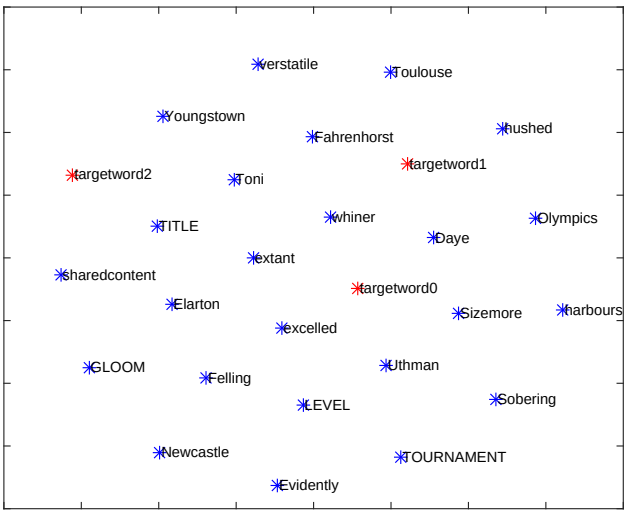

Figure 4: The visualization interpretation result of the first example for the AG's News dataset.

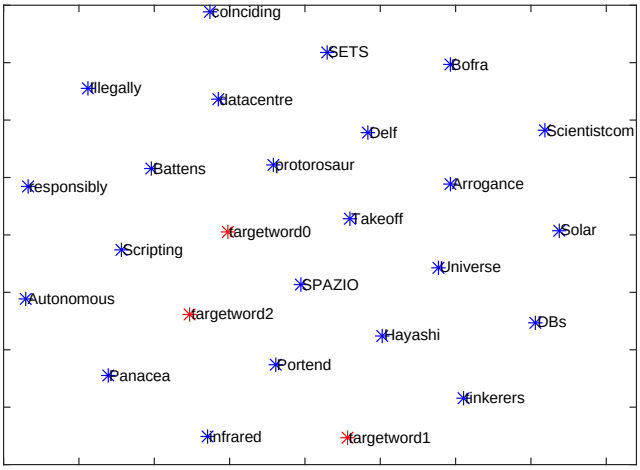

Figure 5: The visualization interpretation result of the second example for the AG's News dataset.

cannot be abstracted. In this work, we propose an approach to interpret deep NLP models. We first combine the saliency map and optimization techniques to explore the information detected by the most important hidden units in NLP models. Then we propose to approximately interpret the meaning of detected information using the nearest neighbors of the optimized representation based on the special property of word representations. Experimental results show that our approaches can identify reasonable interpretation for hidden locations, which shares similar high-level meaning with the input sentence. It is also shown that our method helps explain how the decision and why the decision is made.

Acknowledgement

We thank the anonymous reviewers for their feedback. This work was supported in part by National Science Founda-

tion grant [IIS-1615035] and Defense Advanced Research Projects Agency grant [N66001-17-2-4031].

References

Altman, N. S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* 46(3):175–185.

Du, M.; Liu, N.; Song, Q.; and Hu, X. 2018. Towards explanation of dnn-based prediction with guided feature inversion. *arXiv preprint arXiv:1804.00506*.

Du, M.; Liu, N.; and Hu, X. 2018. Techniques for interpretable machine learning. *arXiv preprint arXiv:1808.00033*.

Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network. *University of Montreal* 1341(3):1.

Gehring, J.; Auli, M.; Grangier, D.; and Dauphin, Y. N. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Li, J.; Chen, X.; Hovy, E.; and Jurafsky, D. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

Lin, K.; Li, D.; He, X.; Zhang, Z.; and Sun, M.-T. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, 3155–3165.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. *arXiv preprint arXiv:1412.0035*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June* 20(14):5.

Nguyen, A.; Yosinski, J.; Bengio, Y.; Dosovitskiy, A.; and Clune, J. 2016. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 427–436.

Olah, C.; Satyanarayan, A.; Johnson, I.; Carter, S.; Schubert, L.; Ye, K.; and Mordvintsev, A. 2018. The building blocks of interpretability. *Distill*. https://distill.pub/2018/building-blocks.

Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature visualization. *Distill*. https://distill.pub/2017/feature-visualization.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Wang, Z., and Ji, S. 2018. Learning convolutional text representations for visual question answering. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 594–602. SIAM.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2(1-3):37–52.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.