Interpretable recurrent neural networks

Maksim Gaponov, Oleg Bahteev, Konstantin Yakovlev, Vadim Strijov
Moscow Institute of Physics and Technology
{gaponov.me, bakhteev, strijov}@phystech.edu

Abstract

Consider the problem of interpretation of recurrent neural networks. We understood the interpretability of the model as the ability to obtain a linear dependence of the output data on the input data. We propose to generalize the OpenBox method designed for the interpretation of neural networks with piecewise linear activation functions. In this method, the neural network is represented as an ensemble of interpretable linear classifiers, each of which is defined on a convex polyhedron. Therefore, the interpretations of close objects are consistent. To test the operability of the proposed method, we conducted an experiment on a sample of the IMDB Review dataset.

Keywords: Interpretation · Recurrent neural network · Piecewise linear neural network

1 Introduction

This paper considers a method for interpreting recurrent deep learning models. Such models are used to process sequences of data [1]. However, due to the large number of layers in the neural network, it works like a "black box". That is, there is no explicit dependence of the output data on the input data. The work is devoted to the construction and selection of interpretable models for recurrent neural networks with piecewise linear activation functions.

A recurrent neural network is a sequence of layers of neurons: input layer, hidden layers and output layer [2]. What distinguishes recurrent neural networks from neural networks are cyclic connections between neurons. Such connections allow to memorize states for processing data sequences.

The interpretation must be exact and consistent [3]. An interpretation is called exact if the predictions of the interpretable model are similar to those of the original model. An interpretation is called consistent if the interpretations of close objects are similar, that is, the significance of the features are close.

Analysis of the dependence of model prediction on input data in recurrent neural networks is an open problem. The methods considered in papers [4, 5, 6, 7, 8, 9] are either inexact, that is, the constructed interpretable models do not correspond to the behavior of the original model, or inconsistent, that is, the significance of features for close objects differ significantly.

Papers [4, 5] have considered methods based on interpretation of features learned by hidden layers. This approach reflects the behavior of the hidden layers, but does not show the behavior of the whole network. Papers [6, 7] propose methods for mimicking the model. In this approach, a model with similar behavior to the original model is constructed. The constructed model has a simpler structure, so it is easier to interpret. However, because of the insufficiently complex structure, the constructed model does not approximate the original model well. Finally, papers [8, 9] consider local interpretation methods, which analyze the behavior of the model in the neighborhood of the original object. In this approach an exact interpretation of a single object is obtained, but the interpretations of close objects can differ significantly. Thus, this approach does not provide a consistent interpretation.

The approach proposed in this paper is a generalization of the OpenBox method designed to interpret neural networks with piecewise linear activation functions [3]. The neural network is represented as an ensemble of linear classifiers. Each of them classifies objects within a convex polytop in the feature space, so the interpretations are consistent.

A computational experiment was conducted on the IMDB Movie Review Dataset [10] to analyze the quality of the method and test the interpretations for exactness and consistency.

2 The problem of selecting interpretable models

Consider a recurrent neural network $\mathbf{f}_{\mathbf{w}}(\mathbf{x})$ with piecewise linear activation functions. We denote the input data by $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbf{X}$, where $\mathbf{X} \subset \mathbb{R}^d$ — feature space. We denote the output data by $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n \in \mathbf{Y}$, where $\mathbf{Y} \subset \mathbb{R}^k$ — prediction space.

The model is a function $\mathbf{f}_{\mathbf{w}} : \mathbf{X} \to \mathbf{Y}$. Because of the large number of parameters $\mathbf{w}$ it is impossible to obtain an explicit dependence of the value of the function $\mathbf{f}_{\mathbf{w}}$ on the argument $\mathbf{x}$.

We need to build a model corresponding to the function $\mathbf{g}$, which has the following properties.

First, the model must be interpretable. That is, you can build a linear dependence of the output data on the input data.

Also, the model must be exact, that is, minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{f}_{\mathbf{w}}(\mathbf{x}_i) - \mathbf{g}(\mathbf{x}_i)\right)^2 \to \min_{\mathbf{g}} \tag{1}$$

The model must be consistent, i.e.

$$\exists \varepsilon > 0 \, \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X} \; ||\mathbf{x}_1 - \mathbf{x}_2|| < \varepsilon \implies \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}_1) = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\mathbf{x}_2) \tag{2}$$

3 Interpretation of recurrent neural networks

The following describes methods for interpreting recurrent neural networks.

3.1 OpenBox method

Consider a recurrent neural network $\mathbf{f}(\mathbf{x})$. For sequences of fixed length, it can be represented as a neural network if you replace the recurrent layers with a sequence of default neuron layers. Let a recurrent neural network consisting of n neurons with a piecewise linear activation function f_i. Denote the number of recurrent runs of a given layer by k. Let's replace this layer in the neural network architecture with k layers consisting of n neurons with activation functions f_i. Thus, the neural network is represented as an default network.

Let us introduce the following notations for the i-th layer

- $\mathbf{a}_i$ – layer input
- $\mathbf{h}_i$ – hidden state
- f_i – activation function
- $\mathbf{W}_i$ – matrix of linear transformation
- $\mathbf{b}_i$ – bias vector

The recurrent neural network is defined by the relation (3)

$$[\mathbf{a}_{i+1}, \mathbf{h}_{i+1}] = f_i\left(\mathbf{W}_i[\mathbf{a}_i, \mathbf{h}_i] + \mathbf{b}_i\right) \tag{3}$$

We consider piecewise linear activation functions, i.e.

$$f_i(x) = \begin{cases} r_1 x + t_1 & \text{if } x \in I_1 \\ r_2 x + t_2 & \text{if } x \in I_2 \\ \dots & \dots \\ r_u x + t_u & \text{if } x \in I_u \end{cases}$$

Where $I_1, \dots, I_u$ – partitioning of $\mathbb{R}$.

For fixed input data $\mathbf{x}$ each piecewise linear function f_i is linear in some neighborhood of its argument.

Also the condition of belonging of the argument x to the interval I_j – is an inequality of the kind $p \leq x \leq q$.

Theorem 1. Let a neural network be given $\mathbf{f}(\mathbf{x})$ with piecewise linear activation functions f_i. Then it is represented as a set of linear functions $F_1, \dots, F_n$, each of which is defined on a convex polyhedron $P_1, \dots, P_n \subset \mathbf{X}$.

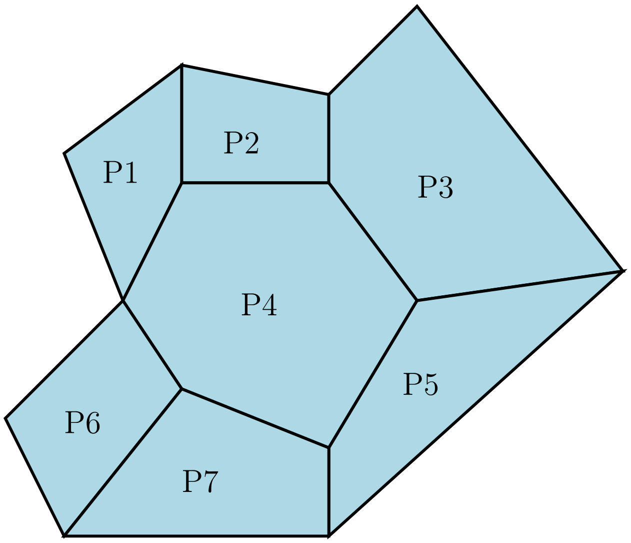

Figure 1: R^2 is splitted into convex polygons $P_1, P_2, P_3 \dots$

Proof. Consider an arbitrary $\mathbf{x}_0 \in \mathbf{X}$. Piecewise linear activation functions f_i are linear in some neighborhood of their argument. The condition that the argument of the function f_i belongs to the interval I_j is the intersection of two linear inequalities, each of which defines a half-space in the space $\mathbf{X}$. The intersection of such set of half-spaces is a convex polyhedron. In addition, the composition of linear functions is a linear function.

Thus a convex polyhedron P_h is found (the intersection of the set of half-spaces) and the linear function F_h (the composition of linear functions) such that the following is fulfilled

$$\forall \mathbf{x} \in P_h \, \mathbf{f}(\mathbf{x}) = F_h(\mathbf{x})$$

Since $\mathbf{x}_0$ is chosen arbitrarily then the space $\mathbf{X}$ is splitted into convex polyhedrons P_h, on which there are linear functions F_h.

The neural network $\mathbf{f}$ has a finite number of neurons, so, there are a finite number of inequalities, defining half-spaces in space $\mathbf{X}$. And a finite number of inequalities defines only a finite number of convex polyhedrons. □

For solving the problem of interpreting recurrent neural networks with piecewise linear activation functions the following algorithm is proposed.

1. Calculate the coefficients of the linear classifier F_h, (h such that $\mathbf{x}_0 \in P_h$) using the formula $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_0)$

2. Obtain linear inequalities on $\mathbf{x}_0$, related to the convex polyhedron P_h.

3. The interpretation of the model on the input object $\mathbf{x}_0$ is the set of coefficients of the linear classifier F_h, corresponding features, and a set of conditions specifying P_h.

The constructed interpretable model is mathematically equivalent to the original model, i.e. the property of exactness is fulfilled (1).

The model is consistent (2), as close objects belong to the same convex polyhedron P_h, and therefore are classified by the same linear classifier F_h.

3.2 LIME method

Consider LIME method [11] for selecting interpretable models.

Algorithm for constructing an interpretable model $\mathbf{g}(\mathbf{x})$

1. Fix the constants n and K.
2. Generate samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x_n}$ in the neighborhood of $\mathbf{x}$.
3. Calculate the predictions $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$ of the model $\mathbf{f}$ on samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$. That is $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$.
4. Build a linear model $\mathbf{g}(\mathbf{x})$ with K parameters using the method of least squares on samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ and predictions $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$.

Thus, the behavior of the model is approximated by a linear model in the neighborhood of the original object.

4 Computational experiment

The experiment deals with the task of classifying reviews on IMDB. We use IMDB Movie Review Dataset [10] consisting of 50,000 reviews, as well as types of reviews: positive, negative. The task is to determine whether the review belongs to the positive class or negative.

To solve the task we used RNN $\mathbf{f}$ (3), whose argument was the sequences of pre-trained vectors of word embeddings. The model has 2,592,105 parameters, it is not interpretable.

The sample is divided into two parts: training and test, in the ratio 3:1. The model is trained on the training sample. After training, the quality of the model is measured on the test sample. The accuracy of the classification was obtained (correct answer rate) 65.55%.

The prediction obtained by the LIME method is as follows.

I went and saw this movie last night after being coaxed to by a few friends of mine. I'll admit that I was reluctant to see it because from what I knew of Ashton Kutcher he was only able to do comedy. I was wrong. Kutcher played the character of Jake Fischer very well, and Kevin Costner played Ben Randall with such professionalism. The sign of a good movie is that it can toy with our emotions. This one did exactly that. The entire theater (which was sold out) was overcome by laughter during the first half of the movie, and were moved to tears during the second half. While exiting the theater I not only saw many women in tears, but many full grown men as well, trying desperately not to let anyone see them crying. This movie was great, and I suggest that you go see it before you judge.

The highlighted words have the strongest effect on the result. The model predicted the review to be in the positive class based on the presence of words and, able, men, last. Obviously, these words do not affect the type of the review. Thus, even without considering accuracy, it is clear that the model is useless because it makes predictions based on irrelevant features.

I went and saw this movie last night after being coaxed to by a few friends of mine. I'll admit that I was reluctant to see it because from what I knew of Ashton Kutcher he was only able to do comedy. I was wrong. Kutcher played the character of Jake Fischer very well, and Kevin Costner played Ben Randall with such professionalism. The sign of a good movie is that it can toy with our emotions. This one did exactly that. The entire theater (which was sold out) was overcome by laughter during the first half of the movie, and were moved to tears during the second half. While exiting the theater I not only saw many women in tears, but many full grown men as well, trying desperately not to let anyone see them crying. This movie was great, and I suggest that you go see it before you judge.

The second example shows that the interpretations are inconsistent. The important words are different. This effect can arise due to the high complexity of the model.

We will try to avoid the above disadvantages in our new approach.

To evaluate the quality of the interpretation, let us make the following graphs:

- Predictions of the interpretable model and the original model.

- The cosine similarity between the predictions of the original model and the constructed interpretable model.

$$\frac{(p_{\text{source}}, p_{\text{interpretable}})}{||p_{\text{source}}|| \cdot ||p_{\text{interpretable}}||}$$

5 Error analysis

Fix the sample size 1,000. Generate a similar object for each object by replacing some words with synonyms. Construct an interpretation on the current object and predict the probability of belonging to a positive class for a similar object. The less the resulting probability differs from the probability predicted by the original model, the better.

For the LIME method, we will build predictions of probabilities not on a similar object, but on the original object.

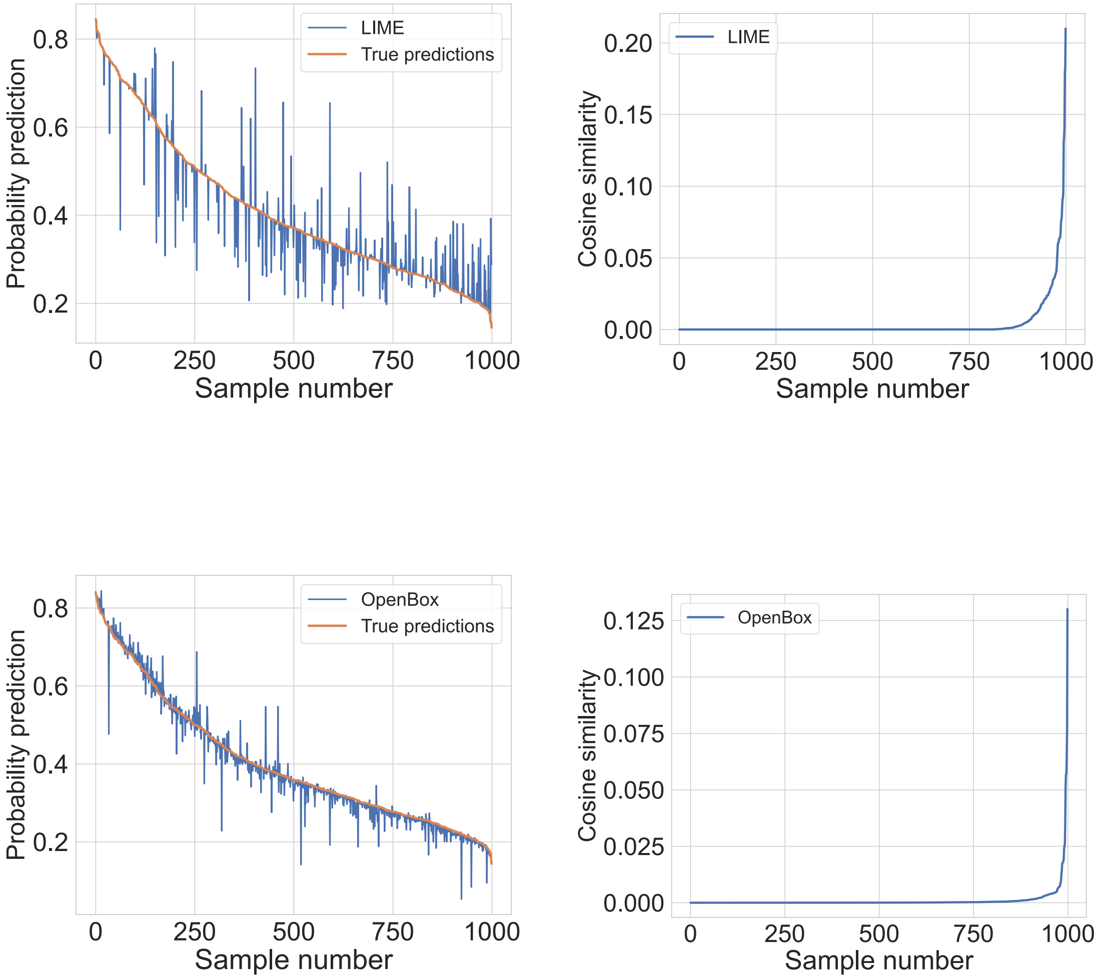

Figure 2: Accuracy of predictions obtained by LIME and OpenBox methods. On the left are graphs of true predictions and predictions of the method, on the right are the cosine similarities between true predictions and method predictions.

The LIME method is not accurate enough in predicting probabilities. The graph is noisy, the predicted probabilities differ significantly from the true values.

The predictions obtained by the LIME method differs (in terms of cosine similarity) on a significant number of objects.

Consider now the OpenBox method.

As we can see from the graph, there is an improvement over the LIME method. The predicted probabilities are less different from the original probabilities.

The cosine similarity graph also shows that the interpretations obtained by the OpenBox method are consistent. There is only a small number of objects where there is a significant difference between the true prediction and the method prediction.

Calculate quality metrics.

Method	RMSE	MAE	MAPE
OpenBox	0.01541	0.00849	0.01923
LIME	0.04085	0.01479	0.03923

The errors of the OpenBox method are smaller, even though the measurements for this method were performed not on the original objects, but on similar ones.

6 Conclusion

The paper proposed a method for selecting an interpretable recurrent model that has the properties of exactness and consistency of interpretations.

Our approach was based on the OpenBox method for interpreting neural networks with piecewise linear activation functions.

We compared the proposed method with the LIME method. The accuracy and consistency of the proposed method were higher.

As an extension of this topic, it can be considered nonlinear activation functions (such as hyperbolic tangent, logistic sigmoid, etc.). If we generalize the method to popular nonlinear activation functions, then there would be a way to obtain interpretations for many well-known recurrent neural networks.

References

[1] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning, 2015.

[2] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, Mar 2020.

[3] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution, 2019.

[4] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks, 2016.

[5] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection, 2018.

[6] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014.

[7] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction, 2019.

[8] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.

[9] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[10] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[11] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.