What is One Grain of Sand in the Desert? *Analyzing Individual Neurons in Deep NLP Models* paper reading report

Sicheng MAO
maosicheng98@gmail.com

Abstract

This is a paper reading assignment for the course **DS-Télécom-20 Natural Language Processing methods for sentiment analysis**. I have picked the paper titled *What is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models*[3]. In this paper, I'm going to 1. analyze the structure and the content of the paper, 2. highlight the main contributions of the paper and analyze its advantages and issues, 3. give a some general comments on it and its related works, 4. make a small demo to reproduce its work and try my attempts for improvement.

1 Analysis of the paper

1.1 Structure

A paper in machine learning field is generally composed of six parts: title, abstract, introduction, method, experiment and conclusion. For this paper, its title quotes a poetic phrase to attract readers. Its abstract briefly introduces the background and their motivation (interpretability of deep neural network). Comparing to previous work, what they feel necessary and remains to improve. What they contribute to the task and what particular methods they have proposed and done. Last but not least, they share their codes to welcome peer review and reproduction. Here, we only analyse the abstract as an example and other parts will be covered from the content perspective.

1.2 Motivation

In contrast to the state-of-the-art performance in AI, neural network lacks interpretability. One of the motivations to study interpretability is due to ethical concerns. The paper falls in the topic of interpretability of deep neural network in NLP task.

In NLP field, previous work on interpretability focus on analysing embeddings and hidden states of models trained on a downstream task. There are two main shortcomings in this approach. First it focus on the whole vector representation but neglecting the role of an individual neuron. Comparing to another major field Computer Vision in AI, there are many works have investigated the behaviour at neuron level. In response, this paper proposes a methodology to identify linguistically-meaningful neurons in deep NLP models inspired by analogous work in CV. Second, it fails to test if the captured features really matter for the end task, i.e. the analysis stays on text explanation on representation level, but not inspect further the influence to the end task. In response, this paper proposes a quantitative evaluation method to the retrieved neurons.

Specifically, the interpretability in this paper is demonstrated by the so-called post-hoc decomposability. According to [7], the post-hoc interpretability only provides analysis on trained results, not on mechanism of algorithm(here the training of neural network). On the other hand, decomposability aims at analysing model behaviour at the level of individual components.

1.3 Method

Here we give an example to briefly explain how to retrieve the important neurons. Suppose we have train a model on a NMT task and we want to retrieve salient neurons from its encoder layer (i.e. the neurons activating hidden states, denoting $z \in R^D$) [1]. The paper

[1] Here, we use D as the dimension of encoding/the number of hidden states/(inspecting) neurons, L as the label numbers and T as the sample numbers. The original paper doesn't treat the notation perfectly, for example when it refers to z_i, I need to ponder a little while to tell whether it refers to the i-th hidden state or the i-th word encoding

proposes two methods to retrieve neurons. The first is to launch a supervised task, that is, given labeled examples $\{x_i, l_i\}_{i=1}^{T}$, do a logistic regression with encoding $\{z_i\}_{i=1}^{T}$. And the learned weights $\theta \in R^D$ directly provides a measure of importance. This is called Linguistic Correlation Analysis, which is not new as supervised task has been performed across various analyzing works. On top of that, this paper proposes another unsupervised method: train several model instances on a certain task with different input data and initialization, compute[2]

$$score(M_{ij}) = \max_{1 \le i' \le N, 1 \le j' \le D, i \ne i'} \rho(M_{ij}, M_{i'j'})$$

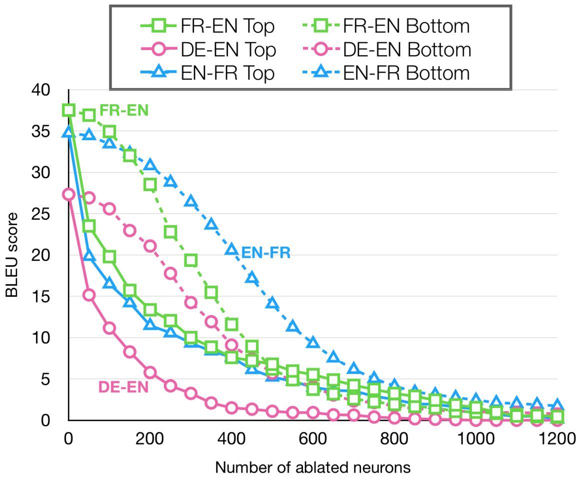
This unsupervised method is one of the paper's contributions called Cross-model Correlation Analysis. I would like to spend more time exploring the ideas behind. The underlying hypothesis is that, if a neuron is salient for the model, then its function should be recognized over all model instances, i.e the model can be always trained to obtain such a neuron to be responsible for the very function(note the relative position of the neuron in the network may vary), whatever the input data and initialization are. To compare the function, the paper uses an evaluation set of T words to activate the neurons. The correlation $\rho(M_{ij}, M_{i'j'})$ measures the similarity between activations of j in i and those of j' in i'. Thus high correlation means the function of neuron j in model i is similar to the one of j' in model i'. And the score of M_{ij} does two things: 1. it finds out the neuron which has the most similar function to M_{ij} in another model instance(max over j'). 2. it scores M_{ij}(actually its function) by the maximum correlation over all the instances.

However, simply taking maximum over all the instances may be biased. Suppose in such an extreme case, we have somehow two identical instances k, k' different from the others, then any neurons $M_{kj}, M_{k'j'}$ in these two model will have score 1 thus rank first. However the function of these neurons may not be captured in other instances, hence it's salient for the instance but not at the model level. To fix this, we may propose another score's formulation:

$$s_{i'}(M_{ij}) = \max_{1 \le j' \le D} \rho(M_{ij}, M_{i'j'})$$

$$score(M_{ij}) = \sum_{i \ne i'} s_{i'}(M_{ij})/(N-1)$$

We call $s_{i'}(M_{ij})$ the presence of function (of neuron j in i) in model i', on which we average to alleviate the bias from instance level.



There still remains one problem: the ranking is performed on a particular model instance i, hence the evaluation of retrieved neurons will also performed on this instance. How can we know which instance to choose in order that the neuron ranking on this instance can represent the neuron ranking for the model, in other words, what if neuron ranking based on different instances differ dramatically? Intuitively, our improvement on score formulation should alleviate this, but this still need to verify.

1.4 Experiment

The paper proposes two ways to ablate neuron. One is masking neuron, i.e. zeroing out the activation in the classifier during test. The other one is removing neuron and retraining the classifier. Masking out neuron causes a drop on test accuracy while retraining can regain some of the accuracy. It suggests that some of the missing information contained in the ablated neuron can be recovered from the other neurons. Hence, although masking can evaluate the neuron ranking, removing and retraining is a more natural way that reveals substantially the information contained in the neuron. Since the former only illustrates how much the information is missing by ablating the neuron while the latter illustrates that how much information is missing and cannot be recovered by the other neurons, i.e. the essential information.

Besides, I find the BLEU score curve dropping with respect to number of ablated neurons contains some interesting pattern. We shall notice that what the author argues as a validation of their neuron ranking is that at the beginning of the curves, the one with ablating top neurons drops faster than the one with ablating bottom neurons. However in the middle stage (between 200 and 600 ablated neurons), the latter also experiences a major score dropping. Hence both ways of ablation

[2] Here $M_{ij} \in R^T$, refers to the activations of neuron j in model i over T words from evaluation set and ρ refers to the pearson correlation coefficient

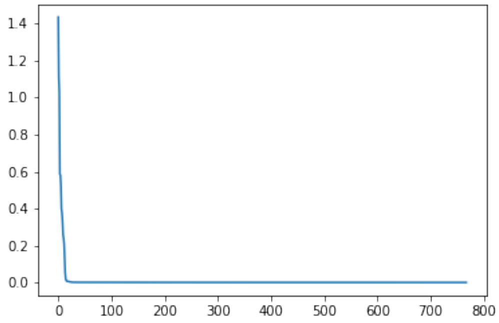


Figure 1: weights distribution of label ,

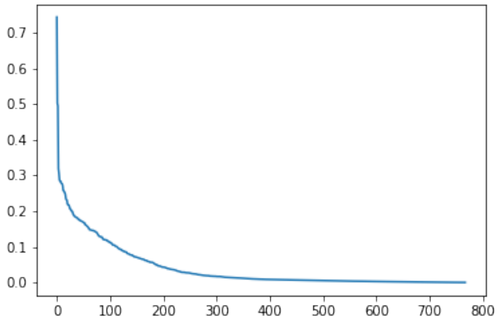


Figure 2: weights distribution of label NN

have rather long tails. In other words, even with high ranking neurons existing, for example at 800 ablated neurons, at least 400 top neurons persisting, the performance is close to have only 400 bottom neurons(within 5 scores difference). The shape of the curves may indicates that the importance of the scale of the neuron representation, i.e. the reason for major score drop at the middle stage may be the structural collapse of the neuron representation and less directly dependent to the absolute ranking order of individual neurons.

Meanwhile, it is beneficial to plot the actual score used for neuron ranking, so that we may have a quantitative impression of their importance instead of an plain order.

1.5 related works

[1] explores the possibility of controlling salient neurons based on the ranking method in this paper. [6] continues to analyze neurons in more complex architecture as XLNet and BERT. [5] applies the neuron ranking to analyse redundancy in large pretrained model as BERT and performs model compression which maintains 97% performance within 10% neurons in original model. [4] integrates the analysing tools in this paper into a toolkit for interpretability research.

2 Reproduction and Experiments

The reproduction depends on [2]. The dataset I choose is CoNLL-2000 [9] which is a POS-tagging dataset with nearly 9k training data labeled with 44 classes. The model we choose is the bert-base-uncased encoder. The notebook is in attachment, here are some results.

We train the POS-tagging classifier on the output embedding of bert encoder(of 768 neurons). We notice that the weight distribution is more concentrate for punctuation(closed class) than the one for noun(open class) which fits to the fact that open class has more variability thus contains more information to be encoded.

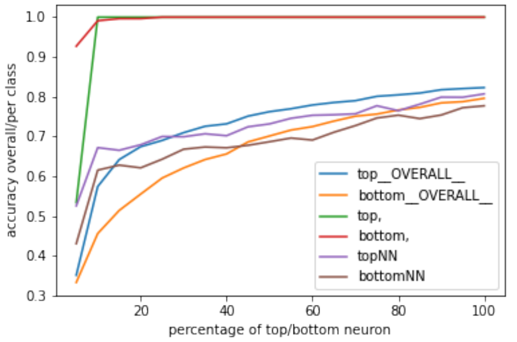


Figure 3: accuracy change with respect to existing neurons

We progressively ablate top/bottom neurons and retrain the classifier and plot the accuracy change.

We notice that the accuracy of the classifier trained with same amount of top neurons is always higher than the one trained with bottom neurons. Besides, the accuracy of open class is more influenced than the one of close class.

3 General comments

This paper is one of the publications from NeuroX project[8]. Its main contribution is a general methodology and a unsupervised method to inspect a single neuron in the representation level. Apart from what we mentioned before, in general we expect future analysis on the intermediate neurons(i.e on architectures) and on the training (i.e. on the dynamics) since they are the state and space factor to the representations. The understanding on them may help us get rid of post-hoc, instance-specific analysis to more efficiently design networks. For example, we may design a more economic model before training, rather than train it first and perform model compression. However, due to the complexity of architecture and the randomness in training process, these tasks are much more difficult.

References

[1] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, J. R. Glass, Identifying and controlling important neurons in neural machine translation, CoRR abs/1811.01157.
URL `http://arxiv.org/abs/1811.01157`

[2] F. Dalvi, Neurox, `https://github.com/fdalvi/NeuroX`.

[3] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, J. R. Glass, What is one grain of sand in the desert? analyzing individual neurons in deep NLP models, CoRR abs/1812.09355.
URL `http://arxiv.org/abs/1812.09355`

[4] F. Dalvi, A. Nortonsmith, A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, J. R. Glass, Neurox: A toolkit for analyzing individual neurons in neural networks, CoRR abs/1812.09359.
URL `http://arxiv.org/abs/1812.09359`

[5] F. Dalvi, H. Sajjad, N. Durrani, Y. Belinkov, Analyzing redundancy in pretrained transformer models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 4908–4926.
URL `https://aclanthology.org/2020.emnlp-main.398`

[6] N. Durrani, H. Sajjad, F. Dalvi, Y. Belinkov, Analyzing individual neurons in pre-trained language models, CoRR abs/2010.02695.
URL `https://arxiv.org/abs/2010.02695`

[7] Z. C. Lipton, The mythos of model interpretability, CoRR abs/1606.03490.
URL `http://arxiv.org/abs/1606.03490`

[8] H. Sajjad, N. Durrani, F. Dalvi, Neurox, `https://neurox.qcri.org/`.

[9] E. F. T. K. Sang, S. Buchholz, Introduction to the conll-2000 shared task: Chunking, CoRR cs.CL/0009008.
URL `https://arxiv.org/abs/cs/0009008`