Interpretable Textual Neuron Representations for NLP

Nina Poerner, Benjamin Roth, Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
`poerner@cis.lmu.de`

Abstract

Input optimization methods, such as Google Deep Dream, create interpretable representations of neurons for computer vision DNNs. We propose and evaluate ways of transferring this technology to NLP. Our results suggest that gradient ascent with a gumbel softmax layer produces n-gram representations that outperform naive corpus search in terms of target neuron activation. The representations highlight differences in syntax awareness between the language and visual models of the Imaginet architecture.

1 Introduction

Deep Neural Networks (DNNs) have led to advances in Natural Language Processing, but they are hard to interpret. This is partly due to the fact that their smallest components, i.e., neurons, lack interpretable representations.

For computer vision problems, Simonyan et al. (2014) propose to use gradient ascent to find an input image that maximizes the activation of a neuron of interest. Using these image representations, one can for instance show that lower level neurons in vision CNNs specialize in patterns such as stripes (Mordvintsev et al., 2015).

Applying gradient ascent input optimization to NLP is not straightforward, as discrete symbols are not open to continuous manipulation. A common alternative approach is to search existing corpora for optimal documents or n-grams (e.g., Kádár et al. (2017), Aubakirova and Bansal (2016)). As this strategy only covers the space of existing inputs, we assume that it may lead to incorrect assumptions. For instance, the representation of a given neuron may suggest that syntax was learned, when in reality this is due to a lack of ungrammatical inputs in the corpus.

In the following, we propose and test methods for gradient ascent input optimization in NLP. Our

quantitative assessment suggests that one method, which is based on the gumbel softmax trick, produces inputs that are more highly activating than corpus search. By applying this method to the Imaginet architecture, we confirm that a language model pays attention to syntax to some degree, while a visual model looks for key content words and ignores function words.

2 Input optimization for NLP

In the following, we denote as $f(\mathbf{E})$ the activation of some neuron of interest when forward-feeding a sequence of embedding vectors $\mathbf{E} = [\mathbf{e}_1 \dots \mathbf{e}_T]$.

2.1 Embedding optimization

One straightforward approach to NLP input optimization is to treat $\mathbf{E}$ like Simonyan et al. (2014) treat images, i.e., to apply gradient ascent directly to the embedding vectors, while keeping other model parameters constant: $\operatorname{argmax}_{\mathbf{E}}\big[f(\mathbf{E})\big]$. However, there is no guarantee that the optimal vectors will correspond to the embedding vectors of real words, or even be close to them. In our experiments, the average cosine proximity to the closest real-word embedding is 0.24, suggesting that there is a divergence between the training goal (finding embedding vectors) and the real goal (finding a representation made up of real words).

2.2 Word optimization

Note that the embedding operation can be written as $\mathbf{E} = \mathbf{XM}$, where $\mathbf{X} \in \{0, 1\}^{T \times V}$ is a matrix of one-hot vectors and $\mathbf{M}$ is the embedding matrix for all V known words. If we relax the requirement that $\mathbf{X}$ be one-hot, we can perform gradient ascent directly on $\mathbf{X}$, while keeping $\mathbf{M}$ constant: $\operatorname{argmax}_{\mathbf{X}}\big[f(\mathbf{XM})\big]$. This approach has the undesirable effect that entries in $\mathbf{X}$ can become very large or negative, and therefore unlike the one-hot vectors seen in training.

To enforce positive vectors that sum to one, we can use the softmax function across the vocabulary axis: $\text{argmax}_{\mathbf{X}}\big[f(\mathbf{P}^{\text{smx}}\mathbf{M})\big]$, where $\mathbf{p}_t^{\text{smx}} = \text{softmax}(\mathbf{x}_t)$. However, this input can still be unlike the inputs seen during training, as the optimal distribution may be smooth.

To remedy this situation, we use the gumbel softmax trick (Jang et al. (2017), Maddison et al. (2017)): $\text{argmax}_{\mathbf{X}}\big[f(\mathbf{P}^{\text{gbl}}\mathbf{M})\big]$, where

$$\mathbf{p}_t^{\text{gbl}} = \text{softmax}\big[\tau^{-1}\big(\log(\mathbf{p}_t^{\text{smx}}) + \mathbf{g}_t\big)\big]$$

and $g_{t,v} \sim -\log(-\log(\mathcal{U}(0,1)))$. By slowly annealing τ, we are able to transition from a uniform probability distribution to a "spiky" one where probability mass is concentrated on few words.

3 Experiment

3.1 Model

We re-implement the Imaginet architecture from Kádár et al. (2017). It consists of a joint word embedding layer (embedding size 1024) and two separate unidirectional GRUs (hidden size 1024 each). One GRU serves as a language model, while the other predicts visual features of a scene described in the input sentence. The model is trained on 566435 MSCOCO captions with visual features taken from Chrupała et al. (2017)[1].
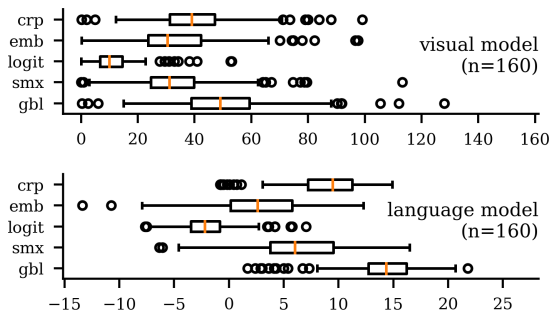

Figure 1: Activation after input optimization for randomly selected projection layer neurons. crp: corpus search; emb: embedding opt.; logit: word opt. w/o softmax; smx: word opt. w/ softmax; gbl: word opt. w/ gumbel softmax.

3.2 Quantitative evaluation

We evaluate the input optimization methods by the activation that they achieve in 160 randomly chosen neurons of the language and visual model projection layers. For embedding optimization, representations are derived by finding the nearest real-word neighbor of the optimized embeddings in the

embedding space. For word optimization, we take the argmax over the vocabulary dimension of $\mathbf{X}$.

We find that while representations from embedding, logit and softmax optimization are not competitive, the gumbel softmax trick outperforms the corpus search strategy in terms of target neuron activation.

3.3 Qualitative observations

Table 1 shows optimal 5-grams for some neurons.

We observe that, contrary to what corpus search suggests, optimal inputs for the visual model rarely contain function words, i.e., the model seems to ignore them. Optimal inputs for the language model sometimes display grammatically correct structures with function words directly before the predicted word (e.g., "stare to their [left]", "under an [umbrella]", see Table 1). This suggests that the language model pays attention to function words and has indeed learned some syntax, as suggested by Kádár et al. (2017).

method	optimal 5-gram	target neuron activation
crp	pizza a sandwich and appetizers	48.44
gbl	fangs calzone raspberries sandwich pizzas	64.46
	315th neuron in visual projection layer	
crp	fighter jet flying in formation	31.25
gbl	propelleor phrases jetliners treetops flight	37.82
	657th neuron in visual projection layer	
crp	a woman sitting under an	13.28
gbl	campbell lawn raincoat under an	17.54
	314th neuron ("umbrella") in language model projection layer	
crp	the view through a car	10.42
gbl	logging jeep watch through cracked	14.87
	957th neuron ("windshield") in language model projection layer	
crp	a giraffe looks to its	10.64
gbl	fest stares stares to their	13.22
	973th neuron ("left") in language model projection layer	

Table 1: Examples of optimal 5-grams via corpus search and via gradient ascent with gumbel softmax. Spelling errors stem from the Imaginet dictionary.

4 Conclusion

The gumbel softmax trick makes it possible to extend the input optimization method to NLP, and to find interpretable textual neuron representations via gradient ascent. Our experimental results suggest that this technique exceeds naive search on a large in-domain corpus in terms of target neuron activation. The representations also show interesting differences in syntax awareness based on target modality in Imaginet.

---

[1]https://zenodo.org/record/804392/files/data.tgz

References

Malika Aubakirova and Mohit Bansal. 2016. Interpreting neural networks to improve politeness comprehension. In *Empirical Methods in Natural Language Processing*, pages 2035–2041, Austin, USA.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Representations of language in a model of visually grounded speech signal. In *Annual Meeting of the Association for Computational Linguistics*, pages 613–622, Vancouver, Canada.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, Toulon, France.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, Toulon, France.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. 2015. DeepDream – a code example for visualizing neural networks. *Google Research*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations*, Banff, Canada.