A Review of the Recursive Neural Tensor Network

October 23, 2022

This article summarises Richard Socher et al.'s paper on sentiment treebank and recursive neural tensor network [1]. The review will start by discussing design ideas and motivations behind datasets and model architecture. Finally, the advantages and potential problems of the entire process will be described.

Stanford Sentiment Treebank (SST) To better train the model for sentiment detection, the Stanford team applied artificial fine-grained sentiment labels to 215,154 phrases in parse trees for 11,855 sentences. Delicate manual labelling work is one aspect, and the biggest highlight is the tree-like structure based on the Stanford parser [2]. A complex problem in natural language processing (NLP) is understanding the element position logic. However, the Stanford team uses a tree-like structure in the sentiment detection task to binarise the position relationship. The parser can convert long sentences into tree-like structures. The critical point is that such a conversion is based on linguistic logic; that is, the sentiment essence of the sentence is the superposition of sub sentiment-phrases. Today, some models pursue complexity and deep architecture while ignoring the intrinsic connection between NLP tasks and linguistics, which is undoubtedly shortsighted. Last, each node of the tree structure is labelled with a sentiment score, significantly improving the model interpretability, and the details are described below.

Recursive Neural Tensor Network (RNTN) When element positions are binarised, the focus of recursive models becomes to efficiently extract representations from two input elements and output a formally recursive result. As stated in the paper, a basic recursive architecture is $p = f\left(W\left[x_1, x_2\right]\right)$, where p refers to a parent node, x_i refers to child nodes, and $f() = \tanh$. When matrix $W \in \mathbb{R}^{d\times 2d}$, there is only a linear transformation between the output before the activation layer and input elements. Thus, the basic architecture is insufficient to model the complexity of the problem. As Socher mentioned: inputs can only interact implicitly via $f() = \tanh$. Therefore, a tensor-based architecture is proposed in Socher's paper.

$$p = f\left(\left[\begin{array}{c} x_1, x_2 \end{array}\right]^T V^{[1:d]} \left[\begin{array}{c} x_1, x_2 \end{array}\right] + W \left[\begin{array}{c} x_1, x_2 \end{array}\right]\right).$$

This review is not intended to repeat the paper, but the analysis of the motivation is necessary. Socher mentioned that he wanted each tensor slice $V^{[i]}$ in the d-dimension to capture a specific feature. Such an idea is reminiscent of the multihead self-attention mechanism in the transformer [3]. Many studies have demonstrated that each head of the transformer has a different division of labour. In addition, is it possible to replace the tensor V with a multilayer neural network?

$$p = f\left(W_1 f\left(W_2\left[x_1, x_2\right]\right)\right).$$

That year, Socher tried the same, but the model was difficult to train. This disadvantage of being difficult to train will continue to occur in the RNTN. Each node can be considered a subtask from representation extraction to sentiment classification. However, the tree-like structure places higher demands on the model, namely global error minimisation. The model must exhibit a global vision, which intuitively means assigning different importance coefficients to various elements. Although the model does not need to consider positional information, it must still model the importance of elements and representational extraction. Thus, the model architecture requires a certain level of complexity, but excessive complexity can make the global error challenging to minimise locally.

Advantages and Potential Problems Interpretability is an excellent advantage of the RNTN. By observing the sentiment output of each node, researchers can understand the inner logic of model judgement. For example, in negation sentence judgement, a positive word, such as "definitely," combined with a negative word, is often classified as negative. Such interpretable logic aligns with linguistic intuition.

The RNTN demonstrates that the Stanford team has a deep understanding of linguistics, and the idea of weakening positional information learning through artificial architecture is inspiring. In future research, the complexity of language model architecture is essential. However, the most critical direction is combining linguistics with deep learning, machine learning, and statistical algorithms.

The performance of the RNTN in the SST sentiment analysis task is excellent, but some drawbacks also limit its development. First, the success of RNTN relies heavily on the SST-based tree structure, and creating an SST-type dataset is challenging to replicate. Second, the architecture of RNTN is insufficiently flexible. As mentioned, the RNTN predicts the sentiment score for each node on the tree structure, so training complex architectures is problematic. Finally, the RNTN has strict requirements on the language structure, and the difficulty of training primarily depends on the self-consistency of the sentiment score for each node. For example, some languages are challenging to convert into tree-like structures, so the RNTN is difficult to apply.

In the sample code for the RNTN, the training process is based on each instance, and during backpropagation, the gradient is calculated on each node. Such a training strategy cannot fully employ the performance of the current graphics processing unit and causes hidden problems regarding the training time. Avati and Chen [4], also from Stanford, have explored this issue and written a paper on it.

In conclusion, the creation of SST and RNTN is instructive. Although the RNTN architecture has not been widely used, its ideas and inspiration will undoubtedly have a profound influence on the field of NLP.

References

[1] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[2] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430, 2003.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] Anand Avati and Nai-Chia Chen. Fast tree-structured recursive neural tensor networks.