A Decomposable Attention Model for Natural Language Inference

The paper "A Decomposable Attention Model for Natural Language Inference" by Ankur P. Parikh, Oscar Tackstrom, Dipanjan Das, and Jakob Uszkoreit published in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* in November 2016, proposed a simple neural architecture for natural language inference. They use attention to decompose the problem unto subproblems to to make it tricially parallelizable.

Currently, a large body of work based on neural networksfor text similarity tasks including NLI has the trend of building complex, deep text representation models, for example, with convolutional networks or long short-term memory networks with the goal of deeper sentence comprehension. While these approaches have yielded impressive results, they are often computationally very expensive, and result in models having millions of parameters (excluding embeddings). The authors argue that for natural language inference it can often suffice to simply align bits of local text substructure and then aggregate this information. Given two sentences, where each word is represented by an embedding vector, they first create a soft alignment matrix using neural attention. They then use the (soft) alignment to decompose the task into subproblems that are solved separately. Finally, the results of these subproblems are merged to produce the final classification. In addition, they optionally apply intrasentence attention to endow the model with a richer encoding of substructures prior to the alignment step.

Complexity of LSTMs: The complexity of an LSTM cell is $O(d^2)$, resulting in a complexity of $O(ld^2)$ to encode the sentence. Adding attention as in increases this complexity to $O(ld^2+l^2d)$. Complexity of decomposable attention model: Application of a feed-forward network requires $O(d^2)$ steps. Thus, the Compare and Aggregate steps have complexity $O(ld^2)$ and $O(d^2)$ respectively. For the Attend step, is evaluated $O(l)$ times, giving a complexity of $O(ld^2)$. Each attention weight requires one dot product, resulting in a complexity of $O(l^2d)$. Thus the total complexity of the model is $O(ld^2+l^2d)$, which is equal to that of an LSTM with attention. However, note that

with the assumption that $l < d$, this becomes $O(ld^2)$ which is the same complexity as a regular LSTM. Moreover, unlike the LSTM, our approach has the advantage of being parallelizable over l, which can be useful at test time.

Empirical results on the SNLI corpus show that our approach achieves state-of-the-art results, while using almost an order of magnitude fewer parameters compared to complex LSTM-based approaches. Adding intra-sentence attention gives a considerable improvement of 0.5 percentage points over the existing state of the art.