Is attention really all you need?

Kyle Roth
Brigham Young University
kylrth@gmail.com

Contents

Abstract

This is a wonderful abstract.

1 Introduction

The attention mechanism is an exciting development in the artificial intelligence community. The concept is simple, and rooted in our understanding of attention in biological intelligence [Larochelle and Hinton, 2010; Hassabis *et al.*, 2017]. Attention mechanisms also lend themselves to more intuitive interpretation, an attractive feature as concerns about model interpretability gain traction.

Ironically, attention mechanisms have received lots of attention since their conception in 2015. Initially formulated as a word alignment mechanism for text translation [Bahdanau *et al.*, 2015], the concept was expanded

Input	Output
images	the objects the image contains
audio recordings	a text transcription
English text	a French translation

Table 1: The caption of the table

2 Theoretical generalization

2.1 Self-attention

2.2 Attention between pairs

2.3 Attention on arbitrary graphs

2.4 Attention as a lookup

2.5 Active memory

Neural memory resources aren't the same thing [Graves *et al.*, 2014]?

3 Application

4 Interpretation

5 Conclusion

A An overview of neural machine learning

Machine learning is a branch of artificial intelligence that deals with training a model to perform tasks by providing the model with data. There are various types of machine learning algorithms, applicable to different tasks. One of the most popular algorithms is the neural network. A neural network is a system of connections between simple functions (e.g. multiplication, summation, ReLU [Nair and Hinton, 2010]), which together approximate a larger, more complex function. It has been proven that sufficiently large neural networks can approximate arbitrary continuous functions [Cybenko, 1989]. Theoretically, this means that for every reasonable mapping there exists a neural network that replicates it. Some examples are listed in Table 1.

Most commonly, neural networks are trained in a supervised way, meaning that they are shown the correct output for each training example. Neural networks generally use backpropagation and gradient descent to train weights that are present in the simple functions of the network. Backpropagation uses the difference between the correct example and

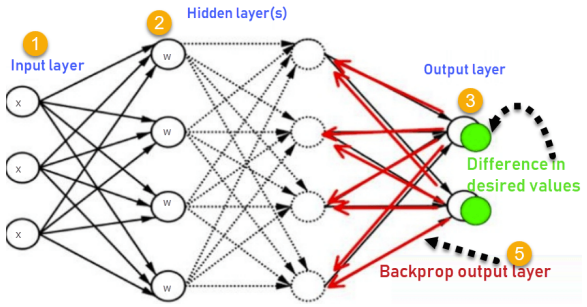

Figure 1: A visualization showing that backpropagation uses the difference between the desired and actual output to train the weights of the neural network. Image source [Guru99, 2019].

the network's output to determine how the weights in the network need to change. Figure 1 provides a visualization of backpropagation.

While the theoretical limitations of neural networks are quite permissive, in practice it is difficult to train them. Backpropagation is sensitive to "noisy" or incorrect data, meaning that data selection is at least as important as architecture selection. The training process is also relatively expensive in terms of computation time and memory usage; neural networks are commonly trained for days or even months [Vaswani *et al.*, 2017].

Another real issue is that backpropagation is unlikely to find the "global optimum", the absolutely optimal set of weights that approximate the function. This is due to the fact that the algorithm only adjusts the weights incrementally, and can get stuck in a "local optimum", a condition where no slight change would improve performance but performance is still not as good as at the global optimum.

These difficulties with training neural networks are the reason that so much effort is given to developing better network architectures. Certain architectures and data representations lend themselves to a smoother search space for the weights, increasing the likelihood that backpropagation reaches a better optimum.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[Cybenko, 1989] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[Graves *et al.*, 2014] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines. *CoRR*, abs/1410.5401, 2014.

[Guru99, 2019] Guru99. Back propagation neural network: Explained with simple example. 2019.

[Hassabis *et al.*, 2017] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95:245–258, 2017.

[Larochelle and Hinton, 2010] Hugo Larochelle and Geoffrey Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 1243–1251, USA, 2010. Curran Associates Inc.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.