CIS 607: Deep Learning for NLP
How Much Attention Do You Need?

Steven Walton

March 13, 2020

1 Introduction

This paper is a response to Google Brain/Research's *Attention Is All You Need*. If attention is all that you need, then the obvious question is "how much?" This paper introduces the Architecture Definition Language, also known as ADL, and then uses it to determine where attention is useful, how much is useful, and how it compares to other network architectures. Specifically this paper compares transformer networks to Convolutional Neural Networks, CNNs, and Recurrent Neural Networks, RNNs. This work differentiates itself from other works by examining specifically transformers instead of self-attention. The work found that source attention on lower encoder layers isn't beneficial, multiple attention and residual feed-forward layers are most beneficial, and self-attention is most beneficial on the encoder side of the network.

2 Architecture Definition Language

To easily experiment with different architectures the authors introduce the Neural Machine Translation Architecture Definition Language, or NMT ADL for short. NMT is used to perform sequence to sequence prediction tasks using machine learning. Specifically, a source sentence X is translated auto-regressively into a target sentence Y, token-wise.

$$p(y_t|Y_{1:t-1}, X, \boldsymbol{\theta}) = softmax(\boldsymbol{W}_o \boldsymbol{z}^L + \boldsymbol{b}_o)$$

where $\boldsymbol{W}_o$ projects a model dependent hidden vector $\boldsymbol{z}^L$ and $\boldsymbol{b}_o$ is the basis vector.

2.1 ADL

ADL is used to define standard NMT architectures. ADL is composed of many layers, l. Every layer has a definition based on the hidden states of the previous layer. Layers are chained together with the notation $l_1 \rightarrow l_2 \rightarrow \cdots \rightarrow l_L$ or $(l_L(...(l_2(l_1(\boldsymbol{H}^o)))))$ if there are no attention layers, where $\boldsymbol{H}^o$ is the matrix of hidden states. This notation is also shortened by the convention $repeat(n, l) = l_1 \rightarrow \cdots \rightarrow l_L$. ADL includes: dropout, fixed positional embedding, linear layers, feed-forward layers, convolutions, identity, concatenations, recurrent neural networks, attention, dot product attention, MLP attention, source attention, self-attention, layer normalization, and residual layers.

A set of standard architectures are also used. These include: RNMT, ConvS2S, and Transformers. RNMT is defined as follows. The encoding layer is represented by

$$\boldsymbol{U}^{L_s} = dropout \rightarrow birnn \rightarrow repeat(n-1, res\_d(rnn))$$

The decoder is represented by

$$\boldsymbol{Z}^{L} = dropout \rightarrow repeat(n, res\_d(rnn)) \rightarrow concat(id, mlp\_att) \rightarrow ff$$

ConvS2s has an encoding layer defined as

$$\boldsymbol{U}^{L_s} = pos \rightarrow repeat(n, res(cnn(glu) \rightarrow dropout))$$

and a decoding layer

$$\boldsymbol{Z}^{L} = pos \rightarrow res(dropout \rightarrow cnn(glu) \rightarrow dropout \rightarrow (dot\_src\_att(s=1)))$$

The transformer has an encoding block

$$t_{enc} = res\_nd(mh\_dot\_self\_att) \rightarrow res\_nd(ffl)$$

and a decoding block

$$t_{dec} = res\_nd(mh\_dot\_self\_att) \rightarrow res\_nd(mh\_dot\_src\_att) \rightarrow res\_nd(ffl)$$

The transformed encoder is

$$\boldsymbol{U}^{L_s} = pos \rightarrow repeat(n, t_{enc}) \rightarrow norm$$

and the decoder

$$\boldsymbol{Z}^{L} = pos \rightarrow repeat(n, t_{dec}) \rightarrow norm$$