How Much Attention Do You Need?
A Granular Analysis of Neural Machine Translation Architectures

Steven Walton
University of Oregon

21 Feb 2019
Overview

Questions:
▶ If attention is all you need, then how much?
▶ Where is the attention important?
▶ What type of attention do we need? Self? LSTM? Transformers?
Overview

Questions:
▶ If attention is all you need, then how much?
▶ Where is the attention important?
▶ What type of attention do we need? Self? LSTM? Transformers?

Answers:
▶ Source attention on lower encoder layers brings no additional benefits.
▶ Multiple source attention and residual feed-forward layers are key.
▶ Self-attention is more important for the source than for the target side.
Flexible Neural Machine Translation Architecture Combination

- Neural Machine Translation (NMT)
- Architecture Definition Language (ADL)
- Layer Definitions
- Standard Architectures

- Related Work
- Experiments
- Conclusion
Neural Machine Translation (NMT)

- NMT is a sequence to sequence prediction task
  \[ X \mapsto Y \]
  
  \[ p(y_t | Y_{1:t-1}, X; \theta) = \text{softmax}(W_o z^L + b_o) \]

- \( W_o \) projects a model dependent hidden vector \( z^L \) of the \( L^{th} \) decoder layer to the dimension of the target vocabulary \( V_{trg} \)

- Training minimizes cross-entropy loss
Architecture Definition Language (ADL)

- ADL is a language used to describe network structures.
- This language lets us discuss a NMT in an easy to understand manner.
- E.G. pos → repeat(n, res(cnn(glu) → dropout))
  - Positional encoding → residual CNN with gated linear units → dropout
Architecture Definition Language (ADL)

Fixed Positional Embeddings

$$\text{pos}(h_t) = \text{dropout}(\sqrt{d}h_t + p_t)$$

Gated Linear Unit (GLU)

$$\text{glu}([h_A; h_B]) = h_A \otimes \sigma(h_B)$$

Dot Product Attention

$$\text{dot att}(Q, K, V, s) = \text{softmax}(\frac{QK^T}{\sqrt{s}})V$$

Layer Normalization

$$\text{nrom}(h_t) = \frac{g}{\sigma_t} \otimes (h_t - \mu_t) + b$$

$$\mu_t = \frac{1}{d}h_{t,j} \quad \sigma_t = \sqrt{\frac{1}{d}(h_{t,j} - \mu_j)^2}$$

Residual Layer

$$\text{res}(h_t, l) = h_t + l(h_t)$$
Architecture Definition Language (ADL)

Transformer
Encoder
\[ t_{enc} = res\_nd(mh\_dot\_self\_attn) \rightarrow res\_nd(ffl) \]

Decoder
\[ t_{dec} =
res\_nd(mh\_dot\_self\_attn) \rightarrow res\_nd(mh\_dot\_src\_att) \rightarrow res\_nd(ffl) \]

RNN
\[
\begin{align*}
\text{rnn}(h_t) &= f_{\text{rnn}_o}(h_t, s_{t-1}) \\
                    & \quad + b \\
s_t &= f_{\text{rnn}_h}(h_t, s_{t-1})
\end{align*}
\]

Convolution
\[
\begin{align*}
cnn(H, v, k) &= v(W[h_{i-\lfloor k/2 \rfloor}; \ldots; h_{i+\lfloor k/2 \rfloor}] + b)
\end{align*}
\]
Flexible Neural Machine Translation Architecture Combination

Related Work

Experiments

Conclusion
Related Work

- Britz et al. (2017) explored hyperparams of RNN NMT models with different attention mechanisms.
- Schrimpf et al (2018) defined language for exploring architectures for RNNs.
- Zoph and Le (2016), Negrinho and Gordan (2017), and Liu et al. (2017) explored architectures for image classification.
Flexible Neural Machine Translation Architecture Combination

Related Work

Experiments

Conclusion
Experiments: Setup

- Used adapted version of SOCKEYE (Heiber et al. 2017)
- Used WMT and IWSLT datasets for different sets of text, which are on the order of 5 million training sentences.
- English → German and Latvian → English translation problems.
- Ran each experiment 3 times with different random seeds. Values reported are the mean result and standard deviation of the BLEU and METEOR scores.
Experiments: What to Attend To?

- Best attention on the upper encoder block.
- No gains observed by attention on different encoder layers in source attention mechanism.

### BLEU Scores

<table>
<thead>
<tr>
<th>Encoder block</th>
<th>IWSLT</th>
<th>WMT’17</th>
</tr>
</thead>
<tbody>
<tr>
<td>upper</td>
<td>25.4 ± 0.2</td>
<td>27.6 ± 0.0</td>
</tr>
<tr>
<td>increasing</td>
<td>25.4 ± 0.1</td>
<td>27.3 ± 0.1</td>
</tr>
<tr>
<td>decreasing</td>
<td>25.3 ± 0.2</td>
<td>27.1 ± 0.1</td>
</tr>
</tbody>
</table>

`Encoder/Decoder Combinations`

- Baseline: 6 layer Transformer, 512 hidden units
- Data sets: IWSLT’16, WMT’17
- Metrics: BLEU (and METEOR in the paper)
- 3 runs, reporting mean and standard deviation
Experiments: Network Structure

RNN $\mapsto$ Transformer

- RNN includes multiple source attention layers, multi-headed attention, layer normalization, residual upscaling FF layers, and single headed MLP attention.
- Start with RNN, add multi-headed attention (mh), positional embedding (pos), layer normalization (norm), single headed attention, attention to residual blocks (multi-att), and residual feed-forward layers after attention blocks (ff).
- Gains from multi-headed attention and feed-forward residual layers.

Can see that as RNN $\mapsto$ Transformer the BLEU scores increase.
Experiments: Network Structure

CNN $\mapsto$ Transformer

- Neither Transformer nor CNN have dependency between decoder timesteps during training, use multiple source attention mechanisms, and use different residual structures.
- Largest gains from adding residual feed-forward layers.

Can see that as CNN $\mapsto$ Transformer the BLEU scores increase.
Experiments: Self-Attention Variations

- Self-attention has advantage because two positions directly connected and no dependencies between consecutive timesteps.
- Self-attention has disadvantage because positional information isn’t directly represented and need multiple heads.
- Experiments show attention is more important to decoder side.
- Attention on encoder shows little to no improvement.
# Experiments: Self-Attention Variations

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Decoder</th>
<th>IWSLT EN→DE BLEU</th>
<th>WMT’17 EN→DE BLEU</th>
<th>WMT’17 LV→EN BLEU</th>
<th>WMT’17 LV→EN METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>self-att</td>
<td>self-att</td>
<td>25.4 ± 0.2</td>
<td>27.6 ± 0.0</td>
<td>18.3 ± 0.0</td>
<td>51.1 ± 0.1</td>
</tr>
<tr>
<td>self-att</td>
<td>RNN</td>
<td>25.1 ± 0.1</td>
<td>27.4 ± 0.1</td>
<td>18.4 ± 0.2</td>
<td>51.1 ± 0.1</td>
</tr>
<tr>
<td>self-att</td>
<td>CNN</td>
<td>25.4 ± 0.4</td>
<td>27.6 ± 0.2</td>
<td>18.0 ± 0.3</td>
<td>50.3 ± 0.3</td>
</tr>
<tr>
<td>RNN</td>
<td>self-att</td>
<td>25.8 ± 0.1</td>
<td>27.2 ± 0.1</td>
<td>17.8 ± 0.1</td>
<td>50.6 ± 0.1</td>
</tr>
<tr>
<td>CNN</td>
<td>self-att</td>
<td>25.7 ± 0.1</td>
<td>26.6 ± 0.3</td>
<td>16.8 ± 0.4</td>
<td>49.4 ± 0.4</td>
</tr>
<tr>
<td>RNN</td>
<td>RNN</td>
<td>25.1 ± 0.1</td>
<td>26.7 ± 0.1</td>
<td>17.8 ± 0.1</td>
<td>50.5 ± 0.1</td>
</tr>
<tr>
<td>CNN</td>
<td>CNN</td>
<td>25.3 ± 0.3</td>
<td>26.9 ± 0.1</td>
<td>16.4 ± 0.2</td>
<td>47.9 ± 0.2</td>
</tr>
<tr>
<td>self-att</td>
<td>combined</td>
<td>25.1 ± 0.2</td>
<td>27.6 ± 0.2</td>
<td>18.3 ± 0.1</td>
<td>51.1 ± 0.1</td>
</tr>
<tr>
<td>self-att</td>
<td>none</td>
<td>23.7 ± 0.2</td>
<td>25.3 ± 0.2</td>
<td>15.9 ± 0.1</td>
<td>45.1 ± 0.2</td>
</tr>
</tbody>
</table>

Table 5: Different variations of the encoder and decoder self-attention layer.
Flexible Neural Machine Translation Architecture Combination

Related Work

Experiments

Conclusion
Conclusion

▶ Defined ADL for specifying NMT architectures on composable building blocks.
▶ Found RNN models benefit from multiple source attention mechanisms and residual feed-forward blocks.
▶ Found CNN benefits from layer normalization and feed-forward blocks.
▶ These features explain the effectiveness of transformers, as they make the respective models more transformer ”like”.
▶ RNN and CNN models with self-attention on the encoder side are competitive with transformers.
QUESTIONS?