RNN

• Advantages:
  
  • State-of-the-art for variable-length representations such as sequences
  
  • RNN are considered core of Seq2Seq (with attention)

• Problems:
  
  • Sequential process prohibits parallelization. Long range dependencies
  
  • Sequences-aligned states: hard to model hierarchical-alike domains ex. languages
• Better than RNN (Linear): path length between positions can be logarithmic when using dilated convolutions

• Drawback: require a lot of layers to catch long-term dependencies
Attention and Self-Attention

- **Attention:** \( \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \)
  - Removes bottleneck of Encoder-Decoder model
  - Focus on important parts

- **Self-Attention:**
  - all the variables (queries, keys and values) come from the same sequence
# Why Self Attention

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Complexity per Layer</th>
<th>Sequential Operations</th>
<th>Maximum Path Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Attention</td>
<td>$O(n^2 \cdot d)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Recurrent</td>
<td>$O(n \cdot d^2)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Convolutional</td>
<td>$O(k \cdot n \cdot d^2)$</td>
<td>$O(1)$</td>
<td>$O(\log_k(n))$</td>
</tr>
<tr>
<td>Self-Attention (restricted)</td>
<td>$O(r \cdot n \cdot d)$</td>
<td>$O(1)$</td>
<td>$O(n/r)$</td>
</tr>
</tbody>
</table>
Transformer Architecture

- Encoder: 6 layers of self-attention + feed-forward network
- Decoder: 6 layers of masked self-attention and output of encoder + feed-forward
Encoder

- $N = 6$
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward
Positional Encoding

- Positional encoding provides relative or absolute position of given token

\[
PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i}/d_{\text{model}}}\right)
\]

\[
PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i}/d_{\text{model}}}\right)
\]

- where pos is the position and i is the dimension
Encoder

- \( N = 6 \)
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward
Scaled Dot Product and Multi-Head Attention

Scaled Dot-Product Attention

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

Multi-Head Attention

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O
\]

where \(\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\)
Encoder

- $N = 6$
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward
Residual Connection

- \text{LayerNorm}(x + \text{Sublayer}(x))
Encoder

- $N = 6$
- All layer output size 512
- Embedding
- Positional Encoding
- Multi-head Attention
- Residual Connection
- Position wise feed forward
Position Wise Feed Forward

- two linear transformation with a ReLU activation in between

\[ FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \]
Decoder

- \( N = 6 \)
- All layer output size 512
- Embedding
- Positional Encoding
- Residual Connection: \( \text{LayerNorm}(x + \text{Sublayer}(x)) \)
- Multi-head Attention
- Position wise feed forward

\[
\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}
\]
• Queries (Q) come from previous decoder layer, and the memory keys (K) and values (V) come from the output of the encoder.

• all three come from previous layer (Hidden State)

Figure 1: The Transformer - model architecture.
Training

- Data sets:
  - WMT 2014 English-German:
    - 4.5 million sentences pairs with 37K tokens.
  - WMT 2014 English-French:
    - 36M sentences, 32K tokens.

- Hardware:
  - 8 Nvidia P100 GPUs (Base model 12 hours, big model 3.5 days)
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>Training Cost (FLOPs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EN-DE</td>
<td>EN-FR</td>
</tr>
<tr>
<td>ByteNet [15]</td>
<td>23.75</td>
<td></td>
</tr>
<tr>
<td>Deep-Att + PosUnk [32]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GNMT + RL [31]</td>
<td>24.6</td>
<td>2.3 \cdot 10^{19}</td>
</tr>
<tr>
<td>ConvS2S [8]</td>
<td>25.16</td>
<td>9.6 \cdot 10^{18}</td>
</tr>
<tr>
<td>MoE [26]</td>
<td>26.03</td>
<td>2.0 \cdot 10^{19}</td>
</tr>
<tr>
<td>Deep-Att + PosUnk Ensemble [32]</td>
<td>40.4</td>
<td>8.0 \cdot 10^{20}</td>
</tr>
<tr>
<td>GNMT + RL Ensemble [31]</td>
<td>26.30</td>
<td>1.8 \cdot 10^{20}</td>
</tr>
<tr>
<td>ConvS2S Ensemble [8]</td>
<td>26.36</td>
<td>7.7 \cdot 10^{19}</td>
</tr>
<tr>
<td>Transformer (base model)</td>
<td>27.3</td>
<td>3.3 \cdot 10^{18}</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>28.4</td>
<td>2.3 \cdot 10^{19}</td>
</tr>
</tbody>
</table>
## More Results

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>(d_{\text{model}})</th>
<th>(d_{\text{ff}})</th>
<th>(h)</th>
<th>(d_k)</th>
<th>(d_v)</th>
<th>(P_{\text{drop}})</th>
<th>(\epsilon_{ls})</th>
<th>train steps</th>
<th>PPL (dev)</th>
<th>BLEU (dev)</th>
<th>params (\times 10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>6</td>
<td>512</td>
<td>2048</td>
<td>8</td>
<td>64</td>
<td>64</td>
<td>0.1</td>
<td>0.1</td>
<td>100K</td>
<td>4.92</td>
<td>25.8</td>
<td>65</td>
</tr>
<tr>
<td>(A)</td>
<td></td>
<td></td>
<td>1</td>
<td>512</td>
<td>512</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.29</td>
<td>24.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>128</td>
<td>128</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.00</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>32</td>
<td>32</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>4.91</td>
<td>25.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td>16</td>
<td>16</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.01</td>
<td>25.4</td>
<td></td>
</tr>
<tr>
<td>(B)</td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.16</td>
<td>25.1</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.01</td>
<td>25.4</td>
<td>60</td>
</tr>
<tr>
<td>(C)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>6.11</td>
<td>23.7</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.19</td>
<td>25.3</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>4.88</td>
<td>25.5</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.75</td>
<td>24.5</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>4.66</td>
<td>26.0</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>5.12</td>
<td>25.4</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>4096</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>4.75</td>
<td>26.2</td>
<td>90</td>
</tr>
<tr>
<td>(D)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>5.77</td>
<td>24.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>4.95</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>4.67</td>
<td>25.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>5.47</td>
<td>25.7</td>
<td></td>
</tr>
<tr>
<td>(E)</td>
<td></td>
<td>positional embedding instead of sinusoids</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>4.92</td>
<td>25.7</td>
<td></td>
</tr>
<tr>
<td>big</td>
<td>6</td>
<td>1024</td>
<td>4096</td>
<td>16</td>
<td></td>
<td></td>
<td>0.3</td>
<td></td>
<td>300K</td>
<td>4.33</td>
<td>26.4</td>
<td>213</td>
</tr>
</tbody>
</table>
Summary

- Introduces a new model, named Transformer
- In particular, introduces the concept of multi-head attention mechanism.
- It follows a classical encoder + decoder structure.
- It is an autoregressive model
- Achieves new state-of-the-art results in NMT