165B
Machine Learning Transformer

Lei Li (leili@cs)
UCSB
Encoder-Decoder Paradigm

I like singing and dancing

我 喜欢 唱歌 和 跳舞
Seq2Seq

- Machine translation as directly learning a function mapping from source sequence to target sequence

Encoder: LSTM
Decoder: LSTM

Source: 天 气 很 好

Target: The weather is nice

$P(Y|X) = \prod P(y_t|y_{<t}, x)$

Training loss: Cross-Entropy

$l = - \sum_n \sum_t \log f_\theta(x_n, y_{n,1}, \ldots, y_{n,t-1})$

Teacher-forcing during training.
(pretend to know groundtruth for prefix)

Sutskever et al. Sequence to Sequence Learning with Neural Networks. 2014
Limitation of RNN/LSTM

- No full context (only one-side)
  - Bidirectional LSTM encoder could alleviate
- But still no long context
- Sequential computation in nature (encoder)
  - not possible to parallelize the computation
- Vanishing gradient
Transformer

• Only use Attention in both encoder and decoder
• no recurrent

Source: 我喜欢唱歌和跳舞。

target: I like singing and dancing.
Transformer

Encoder

Decoder

I like singing and dancing.

我 喜 欢 唱 歌 和 跳 舞。

Vaswani et al. Attention is All You Need. 2017
Transformer Multi-head Attention

- C layers of encoder (=6)
- D layers of decoder (=6)
Scaled Dot-Product Attention

\[
\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V
\]
Multi-head Attention

- Instead of one vector for each token
- break into multiple heads
- each head perform attention

$$\text{Head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_h)W^o$$
Multi-head Attention

\[ X \times W^Q = Q \]

\[ X \times W^K = K \]

\[ X \times W^V = V \]

\[ \text{softmax} \left( \frac{Q \times K^T}{\sqrt{d_k}} \right) \]

\[ Z \]

\[ Z \times V \]

\[ \text{sent len x sent len} \]

\[ \text{sent len x dim} \]

Alammar, The Illustrated Transformer
Self-Attention for Decoder

- Maskout right side before softmax (-inf)
Feedforward Net

- \( FFN(x) = \max(0, x \cdot W_1 + b_1) \cdot W_2 + b_2 \)
- internal dimension size = 2048 (in Vaswani 2017)
Residual Connection and Layer Normalization

- Residual Connection
- Make it zero mean and unit variance within layer
- Post-norm
- Pre-norm
Embedding

- Token Embedding: 512 (base), 1024 (large)
  - Shared (tied) input and output embedding
- Positional Embedding:
  - to distinguish words in different position, Map position labels to embedding, dimension is same as Tok Emb

\[ PE_{pos,2i} = \sin\left(\frac{pos}{1000^{2i/d}}\right) \]
\[ PE_{pos,2i+1} = \cos\left(\frac{pos}{1000^{2i/d}}\right) \]
Transformer

Encoder

Decoder

I like singing and dancing.

我 喜 欢 唱 歌 和 跳 舞。
Training Loss

\[ P(Y \mid X) = \prod P(y_t \mid y_{<t}, x) \]

Training loss: Cross-Entropy

\[ l = - \sum_n \sum_t \log f_\theta(x_n, y_{n,1}, \ldots, y_{n,t-1}) \]

Teacher-forcing during training.

(pretend to know groundtruth for prefix)

target: I like singing and dancing.

Source: 我喜欢唱歌和跳舞。
Training

• Dropout
  – Applied to before residual
  – and to embedding, pos emb.
  – $p=0.1 \sim 0.3$

• Label smoothing
  – 0.1 probability assigned to non-truth

• Vocabulary:
  – En-De: 37K using BPE
  – En-Fr: 32k word-piece (similar to BPE)
Label Smoothing

- Assume \( y \in \mathbb{R}^n \) is the one-hot encoding of label
  \[
y_i = \begin{cases} 
1 & \text{if belongs to class } i \\
0 & \text{otherwise}
\end{cases}
\]

- Approximating 0/1 values with softmax is hard

- The smoothed version
  \[
y_i = \begin{cases} 
1 - \epsilon & \text{if belongs to class } i \\
\epsilon/(n - 1) & \text{otherwise}
\end{cases}
\]

  - Commonly use \( \epsilon = 0.1 \)
Training

- **Batch**
  - group by approximate sentence length
  - still need shuffling
- **Hardware**
  - one machine with 8 GPUs (in 2017 paper)
  - base model: 100k steps (12 hours)
  - large model: 300k steps (3.5 days)
- **Adam Optimizer**
  - increase learning rate during warmup, then decrease

\[
\eta = \frac{1}{\sqrt{d}} \min\left( \frac{1}{\sqrt{t}}, \frac{t}{\sqrt{t_0^3}} \right)
\]
\[ m_{t+1} = \beta_1 m_t - (1 - \beta_1) \nabla \ell(x_t) \]
\[ v_{t+1} = \beta_2 v_t + (1 - \beta_2)(\nabla \ell(x_t))^2 \]
\[ \hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}} \]
\[ \hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^{t+1}} \]
\[ x_{t+1} = x_t - \frac{\eta}{\sqrt{\hat{v}_{t+1} + \epsilon}} \hat{m}_{t+1} \]
Model Average

• A single model obtained by averaging the last 5 checkpoints, which were written at 10-minute interval (base)
• decoding length: within source length + 50
Evaluation for Machine Translation
SpaceX launched a mission Wednesday night to put four amateurs with no space experience into orbit.

SpaceX conducted a launch mission on Wednesday night, sending four amateurs with no aerospace experience into space orbit.

SpaceX conducted a launch mission Wednesday night that sent four amateurs with no spaceflight experience into orbit.

SpaceX carried out a launch mission on Wednesday night to put four amateurs without Aerospace experience into orbit.
• Measuring the precision of n-grams
  – Precision of n-gram: percentage of tokens in output sentences
    \[ p_n = \frac{\text{num. of correct token ngram}}{\text{total output ngram}} \]

• Penalize for brevity
  – if output is too short
    \[ bp = \min(1, e^{1-r/c}) \]

• BLEU = \( bp \cdot \left( \prod p_i \right)^{\frac{1}{4}} \)

• Notice BLEU is computed over the whole corpus, not on one sentence
Ref: A SpaceX rocket was launched into a space orbit Wednesday evening.

System A: SpaceX launched a mission Wednesday evening into a space orbit.

System B: A rocket sent SpaceX into orbit Wednesday.
Ref: A SpaceX rocket was launched into a space orbit Wednesday evening.

System A: SpaceX launched a mission Wednesday evening into a space orbit.

<table>
<thead>
<tr>
<th>n-gram</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram</td>
<td>9/11</td>
</tr>
<tr>
<td>Bigram</td>
<td>4/10</td>
</tr>
<tr>
<td>Trigram</td>
<td>2/9</td>
</tr>
<tr>
<td>Four-gram</td>
<td>1/8</td>
</tr>
</tbody>
</table>

\[ bp = e^{1-12/11} = 0.91 \]
\[ \text{BLEU} = 0.91 \times (9/11 \times 4/10 \times 2/9 \times 1/8)^{1/4} = 28.1\% \]
Machine Translation using Seq2seq and Transformer
LSTM Seq2Seq w/ Attention

Jean et al. On Using Very Large Target Vocabulary for Neural Machine Translation. 2015
Performance with Model Ensemble

Luong et al. Effective Approaches to Attention-based Neural Machine Translation. 2015
**Results on WMT14**

<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>39.2</td>
</tr>
<tr>
<td>GNMT + RL [31]</td>
<td>24.6</td>
<td>39.92</td>
</tr>
<tr>
<td>ConvS2S [8]</td>
<td>25.16</td>
<td>40.46</td>
</tr>
<tr>
<td>MoE [26]</td>
<td>26.03</td>
<td>40.56</td>
</tr>
<tr>
<td>Deep-Att + PosUnk Ensemble [32]</td>
<td></td>
<td>40.4</td>
</tr>
<tr>
<td>GNMT + RL Ensemble [31]</td>
<td>26.30</td>
<td>41.16</td>
</tr>
<tr>
<td>ConvS2S Ensemble [8]</td>
<td>26.36</td>
<td>41.29</td>
</tr>
<tr>
<td>Transformer (base model)</td>
<td>27.3</td>
<td>38.1</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>28.4</td>
<td>41.0</td>
</tr>
</tbody>
</table>
Effectiveness of Choices

- num. heads
- dim of key
- num layers
- hid dim
- ffn dim
- dropout
- pos emb

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$d_{model}$</th>
<th>$d_{ff}$</th>
<th>$h$</th>
<th>$d_{k}$</th>
<th>$d_{v}$</th>
<th>$P_{drop}$</th>
<th>$c_l$s</th>
<th>train steps</th>
<th>PPL (dev)</th>
<th>BLEU (dev)</th>
<th>params $\times 10^{9}$</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>1</td>
<td>512</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.29</td>
<td>24.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>128</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.00</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>32</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.91</td>
<td>25.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.01</td>
<td>25.4</td>
<td></td>
</tr>
<tr>
<td>(B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.16</td>
<td>25.1</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.01</td>
<td>25.4</td>
<td>60</td>
</tr>
<tr>
<td>(C)</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.11</td>
<td>23.7</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.19</td>
<td>25.3</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.88</td>
<td>25.5</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.75</td>
<td>24.5</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.66</td>
<td>26.0</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1024</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.12</td>
<td>25.4</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4096</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.75</td>
<td>26.2</td>
<td>90</td>
</tr>
<tr>
<td>(D)</td>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.77</td>
<td>24.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.95</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.67</td>
<td>25.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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></td>
<td></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>
Deep Transformer

- 30 ~ 60 encoder
- 12 decoder
- dynamic linear combination of layers (DLCL)
  - or. deeply supervised
  - combine output from all layers

<table>
<thead>
<tr>
<th>Model</th>
<th>Param.</th>
<th>Batch (×4096)</th>
<th>Updates (×100k)</th>
<th>ōTimes</th>
<th>BLEU</th>
<th>Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017) (Base)</td>
<td>65M</td>
<td>1</td>
<td>1</td>
<td>reference</td>
<td>27.3</td>
<td>-</td>
</tr>
<tr>
<td>Bapna et al. (2018)-deep (Base, 16L)</td>
<td>137M</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>28.0</td>
<td>-</td>
</tr>
<tr>
<td>Vaswani et al. (2017) (Big)</td>
<td>213M</td>
<td>1</td>
<td>3</td>
<td>3x</td>
<td>28.4</td>
<td>-</td>
</tr>
<tr>
<td>Chen et al. (2018a) (Big)</td>
<td>379M</td>
<td>16</td>
<td>0.075</td>
<td>1.2x</td>
<td>28.5</td>
<td>-</td>
</tr>
<tr>
<td>He et al. (2018) (Big)</td>
<td>210M</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>29.0</td>
<td>-</td>
</tr>
<tr>
<td>Shaw et al. (2018) (Big)</td>
<td>210M</td>
<td>1</td>
<td>3</td>
<td>3x</td>
<td>29.2</td>
<td>-</td>
</tr>
<tr>
<td>Dou et al. (2018) (Big)</td>
<td>356M</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>29.2</td>
<td>-</td>
</tr>
<tr>
<td>Ott et al. (2018) (Big)</td>
<td>210M</td>
<td>14</td>
<td>0.25</td>
<td>3.5x</td>
<td>29.3</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>post-norm</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Base)</td>
<td>62M</td>
<td>1</td>
<td>1</td>
<td>1x</td>
<td>27.5</td>
<td>reference</td>
</tr>
<tr>
<td>Transformer (Big)</td>
<td>211M</td>
<td>1</td>
<td>3</td>
<td>3x</td>
<td>28.8</td>
<td>+1.3</td>
</tr>
<tr>
<td>Transformer-deep (Base, 20L)</td>
<td>106M</td>
<td>2</td>
<td>0.5</td>
<td>1x</td>
<td>failed</td>
<td>failed</td>
</tr>
<tr>
<td>DLCL (Base)</td>
<td>62M</td>
<td>1</td>
<td>1</td>
<td>1x</td>
<td>27.6</td>
<td>+0.1</td>
</tr>
<tr>
<td>DLCL-deep (Base, 25L)</td>
<td>121M</td>
<td>2</td>
<td>0.5</td>
<td>1x</td>
<td>29.2</td>
<td>+1.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pre-norm</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Base)</td>
<td>62M</td>
<td>1</td>
<td>1</td>
<td>1x</td>
<td>27.1</td>
<td>reference</td>
</tr>
<tr>
<td>Transformer (Big)</td>
<td>211M</td>
<td>1</td>
<td>3</td>
<td>3x</td>
<td>28.7</td>
<td>+1.6</td>
</tr>
<tr>
<td>Transformer-deep (Base, 20L)</td>
<td>106M</td>
<td>2</td>
<td>0.5</td>
<td>1x</td>
<td>28.9</td>
<td>+1.8</td>
</tr>
<tr>
<td>DLCL (Base)</td>
<td>62M</td>
<td>1</td>
<td>1</td>
<td>1x</td>
<td>27.3</td>
<td>+0.2</td>
</tr>
<tr>
<td>DLCL-deep (Base, 30L)</td>
<td>137M</td>
<td>2</td>
<td>0.5</td>
<td>1x</td>
<td>29.3</td>
<td>+2.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Param.</th>
<th>newstest17</th>
<th>newstest18</th>
<th>$\Delta_{avg.}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. (2018a) (post-norm, Base)</td>
<td>102.1M</td>
<td>25.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pre-norm Transformer (Base)</td>
<td>102.1M</td>
<td>25.8</td>
<td>25.9</td>
<td>reference</td>
</tr>
<tr>
<td>pre-norm Transformer (Big)</td>
<td>292.4M</td>
<td>26.4</td>
<td>27.0</td>
<td>+0.9</td>
</tr>
<tr>
<td>pre-norm DLCL-deep (Base, 25L)</td>
<td>161.5M</td>
<td>26.7</td>
<td>27.1</td>
<td>+1.0</td>
</tr>
<tr>
<td>pre-norm DLCL-deep (Base, 30L)</td>
<td>177.2M</td>
<td>26.9</td>
<td>27.4</td>
<td>+1.3</td>
</tr>
</tbody>
</table>

Table 4: BLEU scores [%] on WMT’18 Chinese-English translation.
Summary

• Key components in Transformer
  – Positional Embedding (to distinguish tokens at different pos)
  – Multihead attention
  – Residual connection
  – layer norm

• Transformer is effective for machine translation, and many other tasks
Next Up

- Pretraining for NLP