A Tensorized Transformer for Language Modeling

Tianjin University

Xindian Ma, Peng Zhang*, Shuai Zhang, Nan Duan, Yuexian Hou, Dawei Song, Ming Zhou

NeurIPS 2019
Background

- Transformer has led to breakthroughs in natural language processing tasks.
- Transformer, and its variant model BERT, limit the effective deployment of the model to limited resource setting.

- Some compression methods have been proved.
  - TT-embedding
  - BTRNN
  - Tensorizing Neural Networks
Some Compression Methods

- **TT-Embedding [1]**
  - Tensor-Train decomposition is used to compress the embedding layer (look-up table).

- **BTRNN [2]**
  - Block-term tensor decomposition is used to compress the input layers in LSTM

- **Tensorizing Neural Networks [3]**
  - Tensor Train format is used to compress the fully-connected layers.

Problem Formulation

• The goals are:
  • To linearly represent a self-attention by a group of basic vectors
  • To compress multi-head attention in Transformer
  • After compressing, it can be directly integrated into the encoder and decoder framework of Transformer
Methods

Basic Ideas

- Low-rank decomposition
- Parameters sharing

Using **Tucker decomposition** formulation is to construct Single-block attention

Using **Block-term decomposition + Parameters sharing** formulation is to construct multi-head mechanisms (Multi-linear attention)
Transformer Language Modeling

- Scaled Dot Production Attention

\[ \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QKT}{\sqrt{d}} \right) V \]

- Multi-head Attention

\[ \text{MultiHeadAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_k)W^o \]

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

Linear Representation

- **Theorem:** “Scaled Dot Production Attention” can be represented by a linear combination of a set of basis vectors.

\[
Attention(Q, K, V) = (e_1, \cdots, e_n)M
\]

- where \( M \in \mathbb{R}^{n \times d} \) is a coefficient matrix.
Tucker Decomposition

\[ \mathcal{A} \approx \mathcal{G} \times_{1} \mathbf{X}_1 \times_{2} \mathbf{X}_2 \times_{3} \mathbf{X}_3 \]

- \( \mathcal{A} \) is the tensor.
- \( \mathcal{G} \) is the core tensor.
- \( \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \) are the factor matrices.
- Basic vectors represent the dimensions of the original tensor.

The diagram illustrates the Tucker decomposition, showing how a tensor can be approximated by a core tensor and factor matrices.
Single-Block Attention

\[ \mathcal{A} \]

**Core tensor**

\[ \approx \]

**Factor Matrices**

\[ \circ \text{ is the outer product.} \]

Attention_{TD}(\mathcal{G}; Q, K, V) = \mathcal{G} \cdot_1 Q \cdot_2 K \cdot_3 V

\[ = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{m=1}^{M} G_{ijm} Q_i \circ K_j \circ V_m \]
Single-Block Attention in Transformer
The Core-tensor $\mathcal{G}$ is defined as follows.

\[ G_{ijm} = \begin{cases} \text{rand}(0,1), & i = j = m \\ 0 & \text{otherwise} \end{cases} \]

In this case, it can be set as $I = J = M = R$.

$R$ is the Rank.

The time complexity of Single-block attention is $O(N^3)$.

The time complexity of Scaled dot production attention is $O(N^2d)$.
Reconstruction for Scaled dot product attention

• **Corollary:** Scaled dot product attention can be reconstructed by Single block attention

\[
\text{Attention}(Q, K, V)_{i,m} = \sum_{j=1}^{J} \text{Attention}_{TD}(G; Q, K, V)_{i,j,m}
\]

• where \(i, j\) and \(m\) are the indices of the single – block attention's output.
Graphing of Reconstruction

Figure 1: Tensor $\mathcal{A}$ is a 3-order tensor, which represents the Single-block attention in the left. $\mathcal{A}_{i,j,k}$ is the entry of the tensor $\mathcal{A}$. In the right, the graph represents that the summing of tensor slices which is from the tensor splitting in index $j$. 

$\mathcal{A}_{i,j,k} \Rightarrow \mathcal{A}_{i,1,k} + \mathcal{A}_{i,2,k} + \ldots + \mathcal{A}_{i,N,k} = X_{i,k}$
How to Get Richer Representation

- Tensor Split
- Matrices Concat
Multi-linear Attention by Block-term Decomposition

• It is important to constructed the multi-head mechanism for modeling long-range dependency.

• How to design the model with higher compression ratios?
  • 1) Block-term decomposition (method)
  • 2) Parameters sharing (idea)
Multi-linear Attention by Block-term Decomposition

In order to construct the multi-head mechanism, Multi-linear attention can be formulated as follows:

$$MultiLinearAttention(G; Q, K, V) = SplitConcat\left(\frac{1}{h} \ast (T_1 + \cdots + T_h)\right)W^O$$

where $$T_j = Attention_{TD}(G_j; Q^QW^q, K^KW^K, V^VW^V)$$

Parameters Sharing
Multi-Linear Attention
# Experimental Results in Language Modeling

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>Test PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-2018-512 [17]</td>
<td>0.83B</td>
<td>43.7</td>
</tr>
<tr>
<td>GCNN-14 bottleneck [8]</td>
<td>–</td>
<td>31.9</td>
</tr>
<tr>
<td>LSTM-8192-1024+CNN Input [17]</td>
<td>1.04B</td>
<td>30.0</td>
</tr>
<tr>
<td>High-Budget MoE [31]</td>
<td>5B</td>
<td>28.0</td>
</tr>
<tr>
<td>LSTM+Mos [36]</td>
<td>113M</td>
<td>37.10</td>
</tr>
<tr>
<td>Transformer+adaptive input [1]</td>
<td>0.46B</td>
<td>23.7</td>
</tr>
<tr>
<td>Transformer-XL Base [7]</td>
<td>0.46B</td>
<td>23.5</td>
</tr>
<tr>
<td>Transformer-XL Large [7]</td>
<td>0.8B</td>
<td>21.8</td>
</tr>
<tr>
<td>Tensorized Transformer core-1</td>
<td>0.16B</td>
<td>20.5</td>
</tr>
<tr>
<td>Tensorized Transformer core-2</td>
<td>0.16B</td>
<td>19.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>PTB Val PPL</th>
<th>WikiText-103 Val PPL</th>
<th>PTB Test PPL</th>
<th>WikiText-103 Test PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM+augmented loss [15]</td>
<td>24M</td>
<td>75.7</td>
<td>48.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Variational RHN [38]</td>
<td>23M</td>
<td>67.9</td>
<td>65.4</td>
<td>–</td>
<td>45.2</td>
</tr>
<tr>
<td>4-layer QRNN [21]</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>151M</td>
<td>–</td>
</tr>
<tr>
<td>AWD-LSTM-MoS [36]</td>
<td>22M</td>
<td>58.08</td>
<td>55.97</td>
<td>–</td>
<td>29.0</td>
</tr>
<tr>
<td>Transformer+adaptive input [1]</td>
<td>24M</td>
<td>59.1</td>
<td>57</td>
<td>247M</td>
<td>19.8</td>
</tr>
<tr>
<td>Transformer-XL [7]</td>
<td>24M</td>
<td>56.72</td>
<td>54.52</td>
<td>151M</td>
<td>23.1</td>
</tr>
<tr>
<td>Transformer-XL+TT [18]</td>
<td>18M</td>
<td>57.9*</td>
<td>55.4*</td>
<td>130M</td>
<td>23.61*</td>
</tr>
<tr>
<td>Tensorized Transformer core-1</td>
<td>12M</td>
<td>60.5</td>
<td>57.9</td>
<td>80.5M</td>
<td>22.7</td>
</tr>
<tr>
<td>Tensorized Transformer core-2</td>
<td>12M</td>
<td><strong>54.25</strong></td>
<td><strong>49.8</strong></td>
<td>86.5M</td>
<td><strong>19.7</strong></td>
</tr>
</tbody>
</table>

*Note: The results marked with an asterisk (*) are statistically significant.
Experimental Results in Language Modeling

WMT-16 English-to-German

<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base-line [30]</td>
<td>−</td>
<td>26.8</td>
</tr>
<tr>
<td>Linguistic Input Featurec [29]</td>
<td>−</td>
<td>28.4</td>
</tr>
<tr>
<td>Attentional encoder-decoder + BPE [30]</td>
<td>−</td>
<td>34.2</td>
</tr>
<tr>
<td>Transformer [34]</td>
<td>52M</td>
<td>34.5*</td>
</tr>
<tr>
<td>Tensorized Transformer core-1</td>
<td>21M</td>
<td>34.10</td>
</tr>
<tr>
<td>Tensorized Transformer core-2</td>
<td>21.2M</td>
<td>34.91</td>
</tr>
</tbody>
</table>

Conclusion

• We provided a novel self-attention method, namely Multi-linear attention.
• The Tensorized Transformer model combines two compression ideas, parameters sharing and low-rank decomposition.
• Our methods achieve higher compression ratio and better experimental results in language modeling.
• The Tensorized Transformer model can be implied to more NLP tasks with limited resources through further optimization.
Thanks!