DEEP RESIDUAL OUTPUT LAYERS FOR NEURAL LANGUAGE GENERATION

Nikolaos Pappas, James Henderson
Idiap Research Institute, Martigny, Switzerland

PROPOSED ARCHITECTURE OVERVIEW

Output text
\[ w_1, w_2, ..., w_T \rightarrow \tilde{g}_{out} \rightarrow E^{in} \]

E

\[ \hat{y} \]

Input text
\[ w_1, w_2, ..., w_T \rightarrow g_h \rightarrow h_T \]

b

We propose a deep output layer architecture based on the general form and the basic principles of previous work, the power of which no longer depends on the classifier rank d:

\[ p(y_1|y_{1:t-1}) \propto \exp(E^k h_t + b) \]

LABEL ENCODER NETWORK

\[ g_{out}(E)g_{hid}(h_t) + b \]

Shares parameters across outputs through a deep residual output mapping with depth k while keeping the rank d fixed:

\[ E^{(k)} = f_{out}^{(k)}(E^{(k-1)}) = \sigma(E^{(1)}U^{(1)} + b^{(1)}) \]

PROPERTIES

Preserving information with residual connections to the word embedding and, optionally, to the outputs of previous layers:

\[ E^{(k)} = f_{out}^{(k)}(E^{(k-1)}) + E^{(k-1)} + E \]

Controlling power by increasing the projection depth k:

\[ |\Theta_{\text{total}}| \approx k \times (d \times d) \]

Avoiding overfitting with standard or variational dropout in between each of the k projection layers:

\[ f_{out}^{(i)}(E^{(i-1)}) = \delta(f_{out}^{(i)}(E^{(i-1)})) \odot f_{out}^{(i)}(E^{(i-1)}) \]

REFERENCES


EVALUATION

We evaluate on two language generation tasks using state-of-the-art architectures, namely AWD-LSTM [M18] and Transformer [V18].

<table>
<thead>
<tr>
<th>Model</th>
<th>PennTreebank ppl sec/ep</th>
<th>WikiText-2 ppl sec/ep</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWD-LSTM [M18]</td>
<td>57.3 47 (1.0x)</td>
<td>65.8 89 (1.0x)</td>
</tr>
<tr>
<td>AWD-LSTM-DRILL</td>
<td>55.7 53 (1.1x)</td>
<td>61.9 106 (1.2x)</td>
</tr>
<tr>
<td>AWD-LSTM-MoE [V18]</td>
<td>54.44 139 (3.0x)</td>
<td>61.45 862 (9.7x)</td>
</tr>
</tbody>
</table>

MACHINE TRANSLATION

<table>
<thead>
<tr>
<th>Model</th>
<th>En→De (32K) bleu min/ep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (base) [V17]</td>
<td>27.3 111 (1.0x)</td>
</tr>
<tr>
<td>Transformer-DRILL (base)</td>
<td>28.1 189 (1.7x)</td>
</tr>
<tr>
<td>Transformer (big) [V17]</td>
<td>28.4 179 (1.7x)</td>
</tr>
</tbody>
</table>

ABLATION ANALYSIS

Deeper output mappings for neural language generation:

- Improve recurrent or self-attentional architectures without increasing their rank which often leads to high overhead
- Lead to better transfer across the output labels, especially the low-resource ones

Future work: Explore other generation tasks, learn elaborate/multi-level descriptions, investigate transferability

CONCLUSION