Tensor Contraction for Parsimonious Deep Nets

AMAZON AI

Jean Kossaifi, Aran Khanna, Zachary C. Lipton, Tommaso Furlanello and Anima Anandkumar
Traditional approaches

- DATA => CONV => RELU => POOL => Activation tensor
- Flattening looses information
- Can we leverage directly the activation tensor before the flattening?
  - Potential space savings
  - Performance improvement
Tensor Contraction

• Tensor contraction: contract along each mode to obtain a low rank, compact tensor
Tensor Contraction Layers (TCL)

- Take activation tensor as input
- Feed it through a TCL
- Output a low-rank activation tensor
Tensor Contraction Layers (TCL)

- Compact representation less parameters
- Measured in terms of percentage of space savings in the fully connected layers:
  \[ \frac{n_TCL}{n_{original}} \]
- Similar and sometimes better performance
Performance of the TCL

• Trained end-to-end

• On ImageNet with VGG:
  • 65.9% space savings
  • performance drop of 0.6% only

• On ImageNet with AlexNet:
  • 56.6% space savings
  • Performance improvement of 0.5%
Low-rank tensor regression

Tensor Regression Networks, J. Kossaifi, Z.C. Lipton, A. Khanna, T. Furlanello and A. Anandkumar, ArXiv pre-publication
Performance and rank

![Graph showing accuracy vs. space savings]

- **Top-1 accuracy**
- **Top-5 accuracy**

Accuracy (in %) vs. Space savings (in %)
Performance of the TRL

- 92.4% space savings, 4% decrease in Top-1 accuracy
- 68.2% space savings, no decrease in Top-1 accuracy

<table>
<thead>
<tr>
<th>TRL rank</th>
<th>Performance (in %)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-1</td>
<td>Space savings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseline</td>
<td>77.1</td>
<td>93.4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(200, 1, 1, 200)</td>
<td>77.1</td>
<td>93.2</td>
<td>68.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(150, 1, 1, 150)</td>
<td>76</td>
<td>92.9</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(100, 1, 100)</td>
<td>74.6</td>
<td>91.7</td>
<td>84.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(50, 1, 1, 50)</td>
<td>73.6</td>
<td>91</td>
<td>92.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results on ImageNet with a ResNet-101
Implementation

• MXNet as a Deep Learning framework
  http://mxnet.io/

• TensorLy for tensor operations
  https://tensorly.github.io

• Coming soon: mxnet backend for TensorLy
tensor operation on GPU and CPU
Conclusion and future work

- Tensor contraction and tensor regression for Deep Neural Nets
- Add as an additional layer or replace one
- Less parameters
- Similar or even better performance

Future work

- Faster tensor operations
- Explore more tensor operations / networks architectures
Mahalo!

Any question?

jean.kossaifi@gmail.com

@JeanKossaifi