Deep Predictive Coding Network for Object Recognition

Haiguang Wen, Kuan Han, Junxing Shi, Yizhen Zhang, Eugenio Culurciello, Zhongming Liu

Presented by Markus Loide
Overview

- Image recognition
- Convolutional neural networks
- Predictive coding
- Predictive coding network
- Results
- The FUTURE
Image recognition (and others)

(a) Object Classification

(b) Generic Object Detection (Bounding Box)

(c) Semantic Segmentation

(d) Object Instance Segmentation
Convolutional neural networks

- AlexNet
- VGG
- GoogLeNet
- ResNet
- SENets
- Et cetera
Predictive coding

- Influential theory of neural processing in vision (and beyond)
- Hierarchical internal model
- Not only “bottom-up”, but also top-down
- Temporal processes
Predictive coding network

- (Obviously) based on the predictive coding mechanism
- A bi-directional recurrent neural network
- Forward pass - “regular” CNN style, but not really
- Backward pass - predict internal representation of lower level
- Tied parameters vs not tied
Algorithm

- First pass only forward (and on internal representations)
- Then back -> forward
- In the end prediction

Algorithm 1: Deep Predictive Coding Network

1. Input static image: $x$
2. $r_0(t) \leftarrow x$
3. % initialize representations
4. for $l = 0$ to $L-1$ do
5. \hspace{1em} $r_{l+1}(0) \leftarrow \text{ReLU}(\text{FFConv}(r_l(0)))$
6. % recurrent computation with $T$ cycles
7. for $t = 1$ to $T$ do
8. \hspace{1em} % nonlinear feedback process
9. \hspace{2em} for $l = L$ to $1$ do
10. \hspace{3em} $p_{l-1}(t-1) \leftarrow \text{FBConv} \left( r_l(t-1) \right)$
11. \hspace{3em} if $l > 1$ do
12. \hspace{4em} $r_{l-1}(t-1) \leftarrow \text{ReLU} \left( \left( 1-b \right) r_{l-1}(t-1) + b p_{l-1}(t-1) \right)$
13. % nonlinear feedforward process
14. \hspace{2em} for $l = 0$ to $L-1$ do
15. \hspace{3em} $e_l(t) \leftarrow r_l(t) - p_l(t-1)$
16. \hspace{3em} $r_{l+1}(t) \leftarrow \text{ReLU} \left( r_{l+1}(t-1) + a \text{FFConv}(e_l(t)) \right)$
17. % classification
18. Output $r_L(T)$ for classification
Datasets and Trained Networks

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Network variations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>Simple network</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>Number of recursive updates for PCN</td>
</tr>
<tr>
<td>SVHN - (Google’s) Street View House Numbers</td>
<td>Tied vs not tied weights for PCN</td>
</tr>
<tr>
<td>MNIST</td>
<td></td>
</tr>
</tbody>
</table>
Results
What next?

- Parameter tuning

- Some criticisms: did not compare networks with same amount of parameters (when weights not tied)