RECURRENT CONVOLUTIONAL NEURAL NETWORK FOR OBJECT RECOGNITION

Ming Lang and Xialoin Hu

Presenter: Ceren Guzel Turhan

May 3, 2016
Overview
Problem statement
Motivation
Overview of approach
Related studies
RCNN model
Implementations
Experimental setups
Experimental results
Conclusion
OVERVIEW

- Inspired by the fact that the number of recurrent synapses outnumber feed-forward and top-down synapses in the brain

- Idea: recurrent connections within convolutional layers
  - Activity of each unit can be modulated by activities of its neighboring units
    - Enhancing capability of context information
  - Recurrence connections provide multiple paths: facilitating learning
PROBLEM STATEMENT

- Task: object recognition

Easyish, these days
from Fast R-CNN Object detection with caffe by Ross Girshick

Still quite a lot harder
MOTIVATION

- State-of-the-art results using CNN in object recognition
  - in ImageNet [26]
  - in Pascal VOC-2007 [43]
  - in ILSVRC-2014 [50]
  - in CIFAR-10, CIFAR-100, MNIST [33]
**MOTIVATION**

- Brain-CNN and Brain-RNN relationship
  - **CNN**
    - originates from neuroscience (the first artificial neuron)
    - is related to cells in primary visual cortex

---

From Daniel L. K. Yamins and James J. DiCarlo
Brain-CNN and Brain-RNN relationship

- RNN
  - Recurrent synapsis in neocortex
  - Outnumbers feed-forward and top-down synapsis
  - Play an role in context modulation
MOTIVATION

- Object recognition – RNN relationship:
  - Object recognition acts a dynamic process thanks to recurrent and top-down synapsis
  - The processing of visual signals is related to context information
  - The response properties of neurons related to context around RFs
MOTIVATION

- Context information:
  - important for object recognition
  - can be obtained in higher layers of feed-forward models with larger RFs
  - cannot modulated in lower layer for smaller objects

- Strategies for context information
  - top-down connections
  - recurrent connections (in this study)
    - recurrent connections in the same layer
OVERVIEW OF APPROACH

- Similar to RMLP:
  - instead of full connections in RMLP shared local connections

- RCNN: Feed-forward CNN and recurrent connections inside CNN
 RELATED STUDIES

- Similar named studies:
RELATED STUDIES

- **MDRNN [20]:**
  - takes images as 2D sequential data
  - only one hidden layer
  - could not generate features like CNN

- **Hierarchical RNN (NAP) [2]:**
  - Recurrent and feedback connections
    - Vertical and lateral recurrent connections
  - Abstract image representation
  - Network with excitatory and inhibitory units
  - Only feed-forward version in test phase
  - Recurrent version for image reconstruction
RELATED STUDIES

- **CDBN [31]:**
  - top-down connections
  - unsupervised feature learning by propagation of information from top layer to bottom layer

- **rCNN for scene labeling [36]:**
  - Recurrent connection in different layers
  - $rCNN_n : n$ network instance of $CNN_n$
  - Each network instance takes RBG image and previous network output as input

From Pedro O. Pinheiro and Ronan Collobert [36]
 RELATED STUDIES

- Sparse coding models [15]
  - iterative optimization procedures implicitly defines recurrent neural networks

- Recursive CNN [9]
  - time-unfolded version of RCNN
RCNN MODEL: RCL LAYER

- $u^{(i,j)}(t)$: feed-forward input
- $x^{(i,j)}(t - 1)$: recurrent input
- $(i, j)$: location of unit
- $k$: feature map
- $w^f_k$: feed-forward weight
- $w^r_k$: recurrent weight
- $b_k$: bias
- $f$: rectified linear function
- $g$: local response normalization
RCNN MODEL

[Diagram of RCNN Model]

RECURRENT CONVOLUTIONAL NEURAL NETWORK FOR OBJECT RECOGNITION
RCNN MODEL ARCHITECTURE

- Standard convolutional layer, 2 RCLs, pooling, 2 RCLs, pooling, FC layer
- Dropout after each pooling layer except layer 5
- Cross-entropy loss using BPTT
- \( (T+1) \): the depth of each RTL
- \( 4(T+1)+2 \): the length of longest path
IMPLEMENTATIONS

- Cuda-convnet2
- 2 Titan GPU

Hyper-parameters:
- $k$: 96
- Feed-forward filter size in layer: $5 \times 5$
- Feed-forward and recurrent filter size in layer 2 to 4: $3 \times 3$
- For LRN
  - $\alpha$: 0.001
  - $\beta$: 0.75
  - $N = k/8 + 1$
EXPERIMENTAL SETUPS

- Datasets:
  - CIFAR-10
  - CIFAR-100
  - MNIST
  - SVHN

- Trained using BPTT in combination with stochastic gradient descent

- Learning rate: 0.01
  - When accuracy stopped improving, it is decreased to its 1/10
  - Final learning rate is set to 0.0001

- Momentum: 0.9

- Iteration number: 3
EXPERIMENTAL RESULTS: CIFAR-10

- Dataset:
  - 60000 images (50000/10000/10000)
  - 32 × 32 pixel resolutions
  - 10 classes

- Baseline models:
  - WCNN-128: (removed recurrent connections version of RNN with 3 × 3 filters)
  - rCNN-96: (removed recurrent connections of RCLs but adding cascade of duplicated convolutional layers)
EXPERIMENTAL RESULTS: CIFAR-10

- Comparison with baseline models:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training</td>
</tr>
<tr>
<td>rCNN-96 (1 iter)</td>
<td>0.67 M</td>
<td>4.61</td>
</tr>
<tr>
<td>rCNN-96 (1 iter)</td>
<td>0.67 M</td>
<td>2.26</td>
</tr>
<tr>
<td>rCNN-96 (1 iter)</td>
<td>0.67 M</td>
<td>1.24</td>
</tr>
<tr>
<td>WCNN-128 (1 iter)</td>
<td>0.60 M</td>
<td>3.45</td>
</tr>
<tr>
<td>RCNN-96 (1 iter)</td>
<td>0.67 M</td>
<td>4.99</td>
</tr>
<tr>
<td>RCNN-96 (2 iter)</td>
<td>0.67 M</td>
<td>3.58</td>
</tr>
<tr>
<td>RCNN-96 (3 iter)</td>
<td>0.67 M</td>
<td>3.06</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: CIFAR-10

- Comparison with state-of-the-art models without data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maxout[17]</td>
<td>&gt; 5 M</td>
<td>11.68</td>
</tr>
<tr>
<td>Prob maxout [47]</td>
<td>&gt; 5 M</td>
<td>11.35</td>
</tr>
<tr>
<td>NIN [33]</td>
<td>0.97 M</td>
<td>10.41</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>0.97 M</td>
<td>9.69</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>0.67 M</td>
<td>9.31</td>
</tr>
<tr>
<td>RCNN-128</td>
<td>1.19 M</td>
<td>8.98</td>
</tr>
<tr>
<td>RCNN-160</td>
<td>1.86 M</td>
<td>8.69</td>
</tr>
<tr>
<td>RCNN-96 (no dropout)</td>
<td>0.67 M</td>
<td>13.56</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: CIFAR-10

- Comparison with state-of-the-art models with data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob maxout [47]</td>
<td>&gt; 5 M</td>
<td>9.39</td>
</tr>
<tr>
<td>Maxout [17]</td>
<td>&gt; 5 M</td>
<td>9.38</td>
</tr>
<tr>
<td>DropConnect (12 nets) [51]</td>
<td>-</td>
<td>9.32</td>
</tr>
<tr>
<td>NIN [33]</td>
<td>0.97 M</td>
<td>8.81</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>0.97 M</td>
<td>7.97</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>0.67 M</td>
<td>7.37</td>
</tr>
<tr>
<td>RCNN-128</td>
<td>1.19 M</td>
<td>7.24</td>
</tr>
<tr>
<td>RCNN-160</td>
<td>1.86 M</td>
<td>7.09</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: CIFAR-100

- Dataset:
  - 60000 images (50000 | 10000 | 10000)
  - 32 × 32 pixel resolutions
  - 100 classes
  - Same settings as CIFAR-10 without further tuning hyper-parameters
# EXPERIMENTAL RESULTS: CIFAR-100

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maxout [17]</td>
<td>&gt; 5 M</td>
<td>38.57</td>
</tr>
<tr>
<td>Prob maxout [47]</td>
<td>&gt; 5 M</td>
<td>38.14</td>
</tr>
<tr>
<td>Tree based priors [49]</td>
<td>-</td>
<td>36.85</td>
</tr>
<tr>
<td>NIN [33]</td>
<td>0.98 M</td>
<td>35.68</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>0.98 M</td>
<td>34.57</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>0.68 M</td>
<td>34.18</td>
</tr>
<tr>
<td>RCNN-128</td>
<td>1.20 M</td>
<td>32.59</td>
</tr>
<tr>
<td>RCNN-160</td>
<td>1.87 M</td>
<td>31.75</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: CIFAR-100

- Comparison with state-of-the-art models with data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob maxout [47]</td>
<td>&gt; 5 M</td>
<td>9.39</td>
</tr>
<tr>
<td>Maxout[17]</td>
<td>&gt; 5 M</td>
<td>9.38</td>
</tr>
<tr>
<td>DropConnect (12 nets) [51]</td>
<td>-</td>
<td>9.32</td>
</tr>
<tr>
<td>NIN [33]</td>
<td>0.97 M</td>
<td>8.81</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>0.97 M</td>
<td>7.97</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>0.67 M</td>
<td>7.37</td>
</tr>
<tr>
<td>RCNN-128</td>
<td>1.19 M</td>
<td>7.24</td>
</tr>
<tr>
<td>RCNN-160</td>
<td>1.86 M</td>
<td>7.09</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: MNIST

- Dataset
  - 10 classes
  - 70000 images (60000|10000)
  - 28 × 28 pixel

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIN [33]</td>
<td>0.35 M</td>
<td>0.47</td>
</tr>
<tr>
<td>Maxout [17]</td>
<td>0.42 M</td>
<td>0.45</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>0.35 M</td>
<td>0.39</td>
</tr>
<tr>
<td>RCNN-32</td>
<td>0.08 M</td>
<td>0.42</td>
</tr>
<tr>
<td>RCNN-64</td>
<td>0.30 M</td>
<td>0.32</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>0.67 M</td>
<td>0.32</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: SVHN

- Dataset:
  - 10 classes
  - 630420 images (73257 | 26032 | 531131)
  - 32 × 32 pixel

- Without data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maxout [17]</td>
<td>&gt; 5 M</td>
<td>2.47</td>
</tr>
<tr>
<td>Prob Maxout [47]</td>
<td>&gt; 5 M</td>
<td>2.39</td>
</tr>
<tr>
<td>NIN [33]</td>
<td>1.98 M</td>
<td>2.35</td>
</tr>
<tr>
<td>DSN [30]</td>
<td>1.98 M</td>
<td>1.92</td>
</tr>
<tr>
<td>RCNN-32</td>
<td>1.19 M</td>
<td>1.87</td>
</tr>
<tr>
<td>RCNN-64</td>
<td>1.86 M</td>
<td>1.80</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>2.67 M</td>
<td>1.77</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS: SVHN

- With data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-digit number recognition [16]</td>
<td>&gt; 5 M</td>
<td>2.16</td>
</tr>
<tr>
<td>Drop Connect (5 nets) [51]</td>
<td>-</td>
<td>1.94</td>
</tr>
</tbody>
</table>

- Without data augmentation:

<table>
<thead>
<tr>
<th>Model</th>
<th># of parameters</th>
<th>Testing error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCNN-32</td>
<td>1.19 M</td>
<td>1.87</td>
</tr>
<tr>
<td>RCNN-64</td>
<td>1.86 M</td>
<td>1.80</td>
</tr>
<tr>
<td>RCNN-96</td>
<td>2.67 M</td>
<td>1.77</td>
</tr>
</tbody>
</table>
CONCLUSION

- Inspired by recurrent synapsis in the brain
- Idea: adding recurrent connection within convolutional layer
- Enhanced capability of context information about objects
  - facilitating learning by multiple paths thanks to time-unfolded RCNN
- Increasing network depth with constant adjustable parameters
  - going deeper with relatively small number of parameters
- With fewer parameter outperforms state-of-the-art models over four benchmark
- Increasing parameter causes even better performance
THANK YOU

- Any question?