ImageNet Classification with Deep Convolutional Neural Networks

Authors: Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton
University of Toronto

Presenter: Yuanzhe Li
10/4/2016
Outline

• Introduction
• Network Architecture
  • ReLU Nonlinearity
  • Local Response Normalization
  • Overlapping Pooling
  • Overall Architecture
• Reducing Overfitting
  • Data Augmentation
  • Dropout
• Learning
• Results and Discussion
Introduction

• ImageNet
  • Over 15 million labeled high-resolution images
  • Over 22,000 categories

• ILSVRC
  • (In 2010 contest) 1.2 million images depicting 1000 object categories
  • 50,000 validation images, 150,000 testing images
  • Customary to report top-1 and top-5 error rates
Network Architecture

• Overall Architecture
Network Architecture

- Convolution operation

Assume that we use \( m \) by \( m \) kernels in the \( l \)-th layer

\[
x_{ij}^l = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{l-1}
\]

Andrew Gibiansky

Goodfellow et al., 2016
Network Architecture

- 2-D Convolution

$H \times W \times C$

$\# \text{ filters}$

$\# \text{ units} (\hat{W})$

$\# \text{ units} (\hat{H})$

$f_1 = (w, h, C)$
Network Architecture

• Convolutional Layer (1\textsuperscript{st} ConvLayer)
  • Images: 224 x 224 x 3
  • Kernels: 11 x 11
  • Stride = 4 x 4
  • Depth: 48
  • Output: 55 x 55 x 48

For more about filter size, depth, stride, zero-padding, and the convolutional operation, check [Stanford - cs231n](https://cs231n.stanford.edu/).
Network Architecture

• ReLU (Rectified Linear Units) Nonlinearity
  • \( f(x) = \max(0, x) \)
Network Architecture

• Local Response Normalization (LRN)

\[ b_{x,y}^i = \frac{a_{x,y}^i}{\left( k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^j)^2 \right)^\beta} \]

• b: Response-normalized activity
• a: Activity computed by applying kernel i at position (x, y) and the ReLU
• Hyper-parameters: \( k = 2, n = 5, \alpha = 10^{-4}, \beta = 0.75 \)
• Reduces the top-1 and top-5 error rate by 1.4% and 1.2%, respectively
Network Architecture

• Local Pooling
  • Max Pooling Layer
  • Grid of pooling units spaced $s$ pixels apart, summarizing neighborhoods of size $z \times z$
  • $s = z$ means no overlapping

• $s < z$ Overlapping Pooling, same unit can be pooled more than once
• $s = 2, z = 3$
• Reduction on top-1 and top-5 error rates by 0.4% and 0.3% respectively, compared with traditional max pooling

Traditional local max pooling (cs231n)
Network Architecture

• Training on Multiple GPUs
Reducing Overfitting

• Data Augmentation
  • 60 million parameters and 650,000 neurons ⇒ Overfitting
  • Image translation
    • Extracting $224 \times 224$ patches from $256 \times 256$ images, and horizontal reflections
    • Use the average of 10 patches to predict for testing set
  • Altering the intensities of the RGB channels
    • Image pixels $I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$
    • Add $[p_1, p_2, p_3][\alpha_1 \lambda_1, \alpha_2 \lambda_2, \alpha_3 \lambda_3]^T$
    • Reduces top-1 error rate by over 1%

John Loomis
Reducing Overfitting

• Dropout
  • During training, set to zero the output of randomly selected hidden neurons with probability 0.5
  • At test time, use all the neurons but multiply their outputs by 0.5

Tasci and Kim
Details of Learning

• Stochastic Gradient Descent

\[ v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i \]

\[ w_{i+1} := w_i + v_{i+1} \]

\[ \text{Learning rate (initialized at 0.01)} \]

\[ \text{Gradient of Loss w.r.t weight} \]

\[ \text{Averaged over batch} \]

\[ \text{divide by 10 when validation error rate stopped improving with the current rate} \]

• Batch size: 128

• 90 epochs over 1.2 million images (5-6 days on two NVIDIA GTX 580 3GB GPUs)
Results

• Results on ILSVRC-2010

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse coding</td>
<td>47.1%</td>
<td>28.2%</td>
</tr>
<tr>
<td>SIFT + FVs</td>
<td>45.7%</td>
<td>25.7%</td>
</tr>
<tr>
<td>CNN</td>
<td>37.5%</td>
<td>17.0%</td>
</tr>
</tbody>
</table>

• Results on ILSVRC-2012

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT + FVs</td>
<td>26.2%</td>
</tr>
<tr>
<td>1 CNN</td>
<td>16.4%</td>
</tr>
<tr>
<td>5 CNNs</td>
<td>15.3%</td>
</tr>
</tbody>
</table>
Results
Results

• 4096 dimensional features from the last fully-connected layers
Discussion

• To go deep is really important for achieving good results
  • Removing one convolutional layer degrades the result by about 2% for top-1 performance

• Ultimately to use very large and deep convolutional nets

• To explore temporal structure from video sequences, which is missing in static images
Questions?