ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

Presented by
Tugce Tasci, Kyunghee Kim

05/18/2015
Outline

- Goal
- DataSet
- Architecture of the Network
- Reducing overfitting
- Learning
- Results
- Discussion
Goal

Classification

leopard
jaguar
cheetah
snow leopard
Egyptian cat
ImageNet

- Over 15M labeled high resolution images
- Roughly 22K categories
- Collected from web and labeled by Amazon Mechanical Turk

http://image-net.org/
ILSVRC

- Annual competition of image classification at large scale
- 1.2M images in 1K categories
- Classification: make 5 guesses about the image label

EntleBucher

Appenzeller
Convolutional Neural Networks

- Model with a large learning capacity
- Prior knowledge to compensate all data we do not have
ImageNet Classification error throughout years and groups

SuperVision (SV)

Image classification with deep convolutional neural networks

- 7 hidden “weight” layers
- 650K neurons
- 60M parameters
- 630M connections

- Rectified Linear Units, overlapping pooling, dropout trick
- Randomly extracted 224x224 patches for more data

Architecture

5 Convolutional Layers

3 Fully Connected Layers

1000-way softmax
Layer 1 (Convolutional)

- Images: 227x227x3
- F (receptive field size): 11
- S (stride) = 4
- Conv layer output: 55x55x96
Layer 1 (Convolutional)

• \(55 \times 55 \times 96 = 290,400\) neurons
• each has \(11 \times 11 \times 3 = 363\) weights and 1 bias
• \(290400 \times 364 = 105,705,600\) parameters on the first layer of the AlexNet alone!
Architecture

RELU Nonlinearity

- Standard way to model a neuron
  \[ f(x) = \tanh(x) \quad \text{or} \quad f(x) = (1 + e^{-x})^{-1} \]
  
  Very slow to train

- Non-saturating nonlinearity (RELU)
  \[ f(x) = \max(0, x) \]
  
  Quick to train
A 4 layer CNN with ReLUs (solid line) converges six times faster than an equivalent network with tanh neurons (dashed line) on CIFAR-10 dataset.
Architecture

Training on Multiple GPUs

GPU #1

intra-GPU connections

GPU #2

inter-GPU connections
Training on Multiple GPUs

Top-1 and Top-5 error rates decreases by 1.7% & 1.2% respectively, comparing to the net trained with one GPU and half neurons!!
Architecture

Overlaping Pooling

Max-pooling layers

Response normalization layers
Architecture

Local Response Normalization

- No need to input normalization with ReLUs.
- But still the following local normalization scheme helps generalization.

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

- Response normalization reduces top-1 and top-5 error rates by 1.4% and 1.2%, respectively.
• Traditional pooling ($s = z$)

• $s < z \Rightarrow$ overlapping pooling

• Top-1 and top-5 error rates decrease by 0.4% and 0.3%, respectively, compared to the non-overlapping scheme $s = 2, z = 2$
Architecture

- Input image: 224x224x3
- 256 kernels: 5x5x48
- 384 kernels: 3x3x192
- 2048 neurons each
# Architecture Overview

<table>
<thead>
<tr>
<th>Layer</th>
<th>Filters</th>
<th>Activation</th>
<th>Size</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL CONNECT</td>
<td></td>
<td></td>
<td>4M</td>
<td>4Mflop</td>
</tr>
<tr>
<td>FULL 4096/ReLU</td>
<td></td>
<td></td>
<td>16M</td>
<td></td>
</tr>
<tr>
<td>FULL 4096/ReLU</td>
<td></td>
<td></td>
<td>37M</td>
<td></td>
</tr>
<tr>
<td>MAX POOLING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONV 3x3/ReLU 256fm</td>
<td></td>
<td></td>
<td>442K</td>
<td>74M</td>
</tr>
<tr>
<td>CONV 3x3/ReLU 384fm</td>
<td></td>
<td></td>
<td>1.3M</td>
<td>224M</td>
</tr>
<tr>
<td>CONV 3x3/ReLU 384fm</td>
<td></td>
<td></td>
<td>884K</td>
<td>149M</td>
</tr>
<tr>
<td>MAX POOLING 2x2sub</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL CONTRAST NORM</td>
<td></td>
<td></td>
<td>307K</td>
<td>223M</td>
</tr>
<tr>
<td>CONV 11x11/ReLU 256fm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX POOL 2x2sub</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL CONTRAST NORM</td>
<td></td>
<td></td>
<td>35K</td>
<td>105M</td>
</tr>
<tr>
<td>CONV 11x11/ReLU 96fm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Introduction
• DataSet
• Architecture of the Network
• Reducing overfitting
• Learning
• Results
• Discussion
Reducing Overfitting

Data Augmentation

• 60 million parameters, 650,000 neurons
  ➔ Overfits a lot.

• Crop 224x224 patches (and their horizontal reflections.)
Reducing Overfitting

Data Augmentation

• At test time, average the predictions on the 10 patches.
Reducing Overfitting

- Softmax

\[
L = \frac{1}{N} \sum_i -\log \left( \frac{e^{f_{y_j}}}{\sum_j e^{f_j}} \right) + \lambda \sum_k \sum_l W_{k,l}^2
\]

- No need to calibrate to average the predictions over 10 patches.

\[
L = \frac{1}{N} \sum_i \sum_{j \neq y_i} \left[ \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + \Delta) + \lambda \sum_k \sum_l W_{k,l}^2 \right]
\]

cf. SVM

Slide credit from Stanford CS231N Lecture 3
Reducing Overfitting

Data Augmentation

• Change the intensity of RGB channels

$$I_{xy} = \begin{bmatrix} I_{xy}^R, & I_{xy}^G, & I_{xy}^B \end{bmatrix}^T$$

add multiples of principle components

$$[p_1, p_2, p_3][\alpha_1 \lambda_1, \alpha_2 \lambda_2, \alpha_3 \lambda_3]^T$$

$$\alpha_i \sim N(0, 0.1)$$
Reducing Overfitting

**Dropout**

- With probability 0.5
- last two 4096 fully-connected layers.

Figure credit from [Srivastava et al.](http://www.srivastava.org/dropout/).
The training took 5 to 6 days on two NVIDIA GTX 580 3GB GPUs.
Results: ILSVRC-2010

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Sparse coding [2]</em></td>
<td>47.1%</td>
<td>28.2%</td>
</tr>
<tr>
<td><em>SIFT + FVs [24]</em></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>

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.
## Results: ILSVRC-2012

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 (val)</th>
<th>Top-5 (val)</th>
<th>Top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>SIFT + FVs [7]</em></td>
<td>—</td>
<td>—</td>
<td>26.2%</td>
</tr>
<tr>
<td>1 CNN</td>
<td>40.7%</td>
<td>18.2%</td>
<td>—</td>
</tr>
<tr>
<td>5 CNNs</td>
<td>38.1%</td>
<td>16.4%</td>
<td>16.4%</td>
</tr>
<tr>
<td>1 CNN*</td>
<td>39.0%</td>
<td>16.6%</td>
<td>—</td>
</tr>
<tr>
<td>7 CNNs*</td>
<td>36.7%</td>
<td>15.4%</td>
<td>15.3%</td>
</tr>
</tbody>
</table>

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were “pre-trained” to classify the entire ImageNet 2011 Fall release. See Section 6 for details.
96 Convolutional Kernels

- 11 x 11 x 3 size kernels.
- top 48 kernels on GPU 1: color-agnostic
- bottom 48 kernels on GPU 2: color-specific.
Eight ILSVRC-2010 test images
Five ILSVRC-2010 test images

The output from the last 4096 fully-connected layer: 4096 dimensional feature.
Discussion

- Depth is really important.

removing a single convolutional layer degrades the performance.

K. Simonyan, A. Zisserman. 

→ 16-layer model, 19-layer model. 7.3% top-5 test error on ILSVRC-2012
Discussion

- Still have many orders of magnitude to go in order to match the infero-temporal (IT) pathway of the human visual system.
Discussion

• Classification on video.

video sequences provide temporal structure missing in static images.


→ separating two pathways for spatial and temporal networks analogous to the ventral and dorsal pathways.