ImageNet Classification with Deep Convolutional Neural Networks
Mohammad Motamedi
Convolutional Neural Networks (CNNs)

Easier to Train

Much Fewer Connection

Using locality of pixel dependency

Capacity is function of depth and breadth

Image source: stackexchange.com
Training Examples

ImageNet
- Dataset of 15 million labeled high resolution images
- 22000 categories
- Various image resolutions

Data size
- 1.2 million training examples
- 50000 validation images
- 150000 testing images

Preprocessing
- Down-sampled to $256 \times 256$
- Subtracting mean activity over training set from each pixel
The Architecture

Innovations and Details
Rectified Linear Units (ReLU)

Using $f(x) = \max(0, x)$ instead of $\tanh(x)$
- No input normalization is required for saturation prevention

Image source: cs231n.github.io
Local Response Normalization

Normalizing over $n$ adjacent feature maps at the same spatial position.
- It is performed after applying ReLU.

Effect
- Reduces top 1 error by 1.4 %
- Reduces top 5 error by 1.2 %

Image source: computer.org
Overlapping Pooling

Pooling grid of space 2 are used for summarizing neighborhoods of size $3 \times 3$.

Effects
- Reduces the top 1 error rate by 0.4 %
- Reduces the top 5 error rate by 0.4 %
Architecture

- Response normalization: After first and Second Layer
- Max Pooling: After both response normalizations and fifth layer
- ReLU: After each layer
Overfitting

Techniques to Reduce Overfitting
Data Augmentation

Data is augmented by
- Extracting random $224 \times 224$ patches
- Using both patches and their horizontal reflection
- The same approaches is used in the test time (10 patches)

Altering the intensity of RGB channels
- Add found principle components times a random variable proportional to the corresponding eigenvalue

Effect
- Reduces the top 1 error by over 1%
Dropout

Setting the output of each hidden neuron with probability of 0.5
  ◦ This neuron is not effective in the forward path and does not play a role in the backpropagation.

Reduces complex co-adaptation
  ◦ No neuron can rely on the presence of another neuron
Implementation

Training time: six days on two GTX 580 3 GB GPUs

Effect on network
- It is required to minimize the inter chip communication

Augmenting the data on CPU in parallel with training on GPU
- Augmented data does not need to be stored on the disk

Effect:
- Reduces the top 1 error by 1.7 %
- Reduces the top 5 error by 1.2 %
Training

Network is trained with stochastic gradient descent
- Weight decay: 0.0005
- Momentum: 0.9
- Weights are initialized by random numbers from a zero – mean Gaussian distribution with standard deviation of 0.01
- Divide learning rate by 10 when error stops improving
Results
Kernel values after training
ILSVRC 2010

![Bar chart showing error rates for different methods in Top-1 and Top-5 positions. The methods compared are CNN, SIFT + FVs [24], and Sparse coding [2].]
ILSVRC 2012

<table>
<thead>
<tr>
<th>Model</th>
<th>Top – 1 Error (Val)</th>
<th>Top – 5 Error (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT + FVs [7]</td>
<td>-</td>
<td>26.2%</td>
</tr>
<tr>
<td>1 CNN</td>
<td>18.2%</td>
<td>-</td>
</tr>
<tr>
<td>5 CNNs</td>
<td>16.4%</td>
<td>16.4%</td>
</tr>
<tr>
<td>7 CNNs</td>
<td>15.4%</td>
<td>15.3%</td>
</tr>
</tbody>
</table>
ILSVRC 2010
ILSVRC