Building high-level features using large scale unsupervised learning

Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, A.Y. Ng

VGG reading group
September 2012
Google computer works out how to spot cats

A Google research team has trained a network of 1,000 computers wired up like a brain to recognise cats.

The team built a neural network, which mimics the working of a biological brain, that worked out how to spot pictures of cats in just three days.

The cat-spotting computer was created as part of a larger project to investigate machine learning.

Google is planning to use the learning system to help with its indexing systems and with language translation.
Deep learning

The big idea

Replace with learned features?

Images → Hand-crafted features (SIFT, SURF) → Detectors and classifiers
Contributions

1. The model
   Deep
   Large

2. The results
   Good detectors when unsupervised
   Good classifiers when supervised

3. Implementation
   Model parallelism
   Data parallelism
Contributions

1. The model
   Deep
   Large

2. The results
   Good detectors when unsupervised
   Good classifiers when supervised

3. The implementation
   Model parallelism
   Data parallelism
The model
Architecture

Layer 3 units
8 channels

Layer 2 units
8 channels

Layer 1 units
3 channels (RGB)

Image units
The model
The first layer

- Image units
- Layer 1 units

Input to another layer above
(image with 8 channels)

- Number of output channels = 8
- LCN Size = 5
- Pooling Size = 5
- RF size = 18
- Number of maps = 8
- Number of input channels = 3
- Image Size = 200
The model

Training objective

Topographic Independent Component Analysis
The model

Training objective

Topographic Independent Component Analysis

\[
\text{minimize} \sum_{i=1}^{m} \left( \|W_2 W_1^T x^{(i)} - x^{(i)} \|_2^2 + \right) + \lambda \sum_{j=1}^{k} \sqrt{\epsilon + H_j (W_1^T x^{(i)})^2}.
\]

- **Encode**
- **Reconstruct**
- **Tradeoff between sparsity and reconstruction**
- **Sum over pooling units**
- **Fixed to uniform**
The model
Local contrast normalisation

Subtractive

\[ g_{i,j,k} = h_{i,j,k} - \sum_{iuv} G_{uv} h_{i,j+u,i+v} \]

Divisive

\[ y_{i,j,k} = \frac{g_{i,j,k}}{\max\{c, (\sum_{iuv} G_{uv} g_{i,j+u,i+v}^2)^{0.5}\}} \]
The model
Architecture

Layer 1 units

Image units
The model

Architecture

Layer 3 units

Layer 2 units

Layer 1 units

Image units

Model
1 billion parameters

Visual Cortex
1 million billion parameters
Contributions

1. The model
   Deep
   Large

2. The results
   Good detectors when \textit{unsupervised}
   Good classifiers when \textit{supervised}

3. The implementation
   Model parallelism
   Data parallelism
The results
Training data

- 10 million YouTube videos
- One image per video
- 200 x 200 pixels
- Approximately 3% contain faces
- Determined using OpenCV
- Random patches at 60 x 60 pixels
The results
Face detection test data

- 37,000 images from LFW and ImageNet
- ~35% positive
- ~65% negative
The results

Best face detector neuron

Protocol

- Record activations of every neuron for every test image
- Compute accuracy of each neuron as a face detector
  - At 20 different thresholds
- Identify neuron that achieves highest accuracy
The results
Good detectors when unsupervised

<table>
<thead>
<tr>
<th>Concept</th>
<th>Random guess</th>
<th>Same architecture with random weights</th>
<th>Best linear filter</th>
<th>Best first layer neuron</th>
<th>Best neuron</th>
<th>Best neuron without contrast normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faces</td>
<td>64.8%</td>
<td>67.0%</td>
<td>74.0%</td>
<td>71.0%</td>
<td>81.7%</td>
<td>78.5%</td>
</tr>
<tr>
<td>Human bodies</td>
<td>64.8%</td>
<td>66.5%</td>
<td>68.1%</td>
<td>67.2%</td>
<td>76.8%</td>
<td>71.8%</td>
</tr>
<tr>
<td>Cats</td>
<td>64.8%</td>
<td>66.0%</td>
<td>67.8%</td>
<td>67.1%</td>
<td>74.6%</td>
<td>69.3%</td>
</tr>
</tbody>
</table>

Comparison with OpenCV?
The results

Best face detector neuron

Which image in the test dataset leads to the largest activation of this neuron?
The results

Best face detector neuron

What kind of image leads to the highest activation of this neuron?

\[ x^* = \arg \min_x f(x; W, H), \text{ subject to } ||x||_2 = 1. \]
The results

Best face detector neuron

What kind of image leads to the highest activation of this neuron?
The results
Best face detector neuron
To what extent is it ‘invariant’?

- 10 translated, scaled faces
- 10 rotated faces
The results

Best face detector neuron

To what extent is it ‘invariant’?
The results

Good detectors when unsupervised

<table>
<thead>
<tr>
<th>Concept</th>
<th>Our network</th>
<th>Deep autoencoders 3 layers</th>
<th>Deep autoencoders 6 layers</th>
<th>K-means on 40x40 images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faces</td>
<td>81.7%</td>
<td>72.3%</td>
<td>70.9%</td>
<td>72.5%</td>
</tr>
<tr>
<td>Human bodies</td>
<td>76.7%</td>
<td>71.2%</td>
<td>69.8%</td>
<td>69.3%</td>
</tr>
<tr>
<td>Cats</td>
<td>74.8%</td>
<td>67.5%</td>
<td>68.3%</td>
<td>68.5%</td>
</tr>
</tbody>
</table>
The results
Good detectors when supervised

• Train as before on YouTube + ImageNet
• Add one-vs-all logistic classifier on the top units
• Optimise:
  1. parameters of logistic classifiers, then
  2. parameters of layers
## The results

Good detectors when supervised

<table>
<thead>
<tr>
<th>Dataset version</th>
<th>2009 (~9M images, ~10K categories)</th>
<th>2011 (~14M images, ~22K categories)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-of-the-art</strong></td>
<td>16.7% (Sanchez &amp; Perronnin, 2011)</td>
<td>9.3% (Weston et al., 2011)</td>
</tr>
<tr>
<td><strong>Our method</strong></td>
<td>16.1% (without unsupervised pretraining)</td>
<td>13.6% (without unsupervised pretraining)</td>
</tr>
<tr>
<td></td>
<td><strong>19.2% (with unsupervised pretraining)</strong></td>
<td><strong>15.8% (with unsupervised pretraining)</strong></td>
</tr>
</tbody>
</table>
The results

Good detectors when supervised

Which image in the test dataset leads to the largest activation of this neuron?
The results
Good detectors when supervised
Which image in the test dataset leads to the largest activation of this neuron?

Features here have been hand-selected
Contributions

1. The model
   Deep
   Large

2. The results
   Good detectors when unsupervised
   Good classifiers when supervised

3. The implementation
   Model parallelism
   Data parallelism
The implementation

Data parallelism

Parameter updates

- Asynchronous stochastic gradient descent
- Train on 1,000 machines (16,000 cores)
- Train for 3 days
The implementation

Data parallelism

Cluster updates
The implementation

Data parallelism

Parameter updates
The implementation
Model parallelism