Going deeper with convolutions

GoogLeNet

BIL722 Advanced Vision - Presentation

Mehmet Günel
Christian Szegedy, Google

Pierre Sermanet, Google

Dumitru Erhan, Google

Wei Liu, UNC

Scott Reed, University of Michigan

Vincent Vanhoucke, Google

Yangqing Jia, Google

Dragomir Anguelov, Google

Andrew Rabinovich, Google
Basics

• What is ILSVRC14?
  - ImageNet Large-Scale Visual Recognition Challenge 2014

• What is ImageNet?
  - WordNet hierarchy, concept = "synonym set" or "synset".
  - More than 100,000 synsets in WordNet, on average 1000 images to illustrate each synset

• What are Google Inception and GoogLeNet?
  - Network in network
  - We need to go deeper
Overview of the GoogleNet

- A deep convolutional neural network architecture
- Classification and detection for ILSVRC14
- Improved utilization of the computing resources inside the network while increasing size, both depth and width
- 12x fewer parameters than the winning architecture of Krizhevsky
- Significantly more accurate than state of the art
- 22 layers deep when counting only layers with parameters
- The overall number of layers (independent building blocks) used for the construction of the network is about 100
What is the Problem?

• Aim:
  – To improve the performance of classification and detection

• Restrictions:
  – Usage of CNN
  – Able to train with smaller dataset
  – Limited computational power and memory usage
How to improve classification and detection rates?

- Straightforward approach;

  Just increase the size of network in both direction!

BUT!!!
Straightforward approach, challenge 1

- Larger number of parameters \(\rightarrow\) Requires bigger data;

  Otherwise overfit! High quality training sets can be tricky and expensive...

(a) Siberian husky  

(b) Eskimo dog
Straightforward approach, challenge 2

- Dramatically increased use of computational resources!
- A simple example:
  - If two convolutional layers are chained, any uniform increase in the number of their filters results in a quadratic increase of computation
What is their approach?

- Moving from fully connected to sparsely connected architectures, even inside the convolutions
Handicap of the sparse approach

- Todays computing infrastructures are very inefficient when it comes to numerical calculation on non-uniform sparse data structures
- The gap is widened even further by the use of steadily improving, highly tuned, numerical libraries that allow for extremely fast dense matrix multiplication, exploiting the minute details of the underlying CPU or GPU hardware
- Also, non-uniform sparse models require more sophisticated engineering and computing infrastructure
- Even people go back to fully connected approach!
Their Solution

• An architecture that makes use of the extra sparsity, even at filter level, as suggested by the theory, but exploits our current hardware by utilizing computations on dense matrices.

• Clustering sparse matrices into relatively dense submatrices tends to give state of the art practical performance for sparse matrix multiplication.
Their motivation

- Multi-scale processing namely synergy of deep architectures and classical computer vision, like the R-CNN algorithm by Girshick

- If the probability distribution of the data-set is representable by a large, very sparse deep neural network, then the optimal network topology can be constructed layer by layer by analyzing the correlation statistics of the activations of the last layer and clustering neurons with highly correlated outputs

- Hebbian principle: neurons that fire together, wire together
Hebbian Principle
Cluster according activation statistics

Layer 1

Input
Cluster according correlation statistics
Cluster according correlation statistics

Layer 1

Layer 2

Layer 3

Input
In images, correlations tend to be local
Cover very local clusters by 1x1 convolutions

number of filters

1x1
Less spread out correlations

number of filters

1x1
Cover more spread out clusters by 3x3 convolutions
Cover more spread out clusters by 5x5 convolutions

number of filters

1x1

3x3
Cover more spread out clusters by 5x5 convolutions
A heterogeneous set of convolutions

number of filters

1x1
3x3
5x5
Schematic view (naive version)

number of filters

1x1 convolution
3x3 convolution
5x5 convolution

Filter concatenation
1x1 convolutions
3x3 convolutions
5x5 convolutions
Previous layer
Naive idea
Naive idea (does not work!)
Inception module
• 1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions.

• Besides being used as reductions, they also include the use of rectified linear activation which makes them dual-purpose.
How these 1x1 convolutions work?

- Receptive field
- Not dimensionality reduction in space, but can dimensionality reduction in channel
- ReLU functionality
Solution Details

- Optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components
- Find the optimal local construction and repeat it spatially
GoogLeNet

Convolution
Pooling
Softmax
Other
Width of **inception modules** ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Number of parameters is reduced to 5 million

Computational cost is increased by less than 2X compared to Krizhevsky’s network. (<1.5Bn operations/evaluation)
Auxiliary classifiers

- Encourage discrimination in the lower stages in the classifier
- Increase the gradient signal that gets propagated back
- Provide additional regularization
Auxiliary classifiers

• An average pooling layer with 5x5 filter size and stride 3, resulting in an 4x4x512 output or the (4a), and 4x4x528 for the (4d) stage.

• A 1x1 convolution with 128 filters for dimension reduction and rectified linear activation.

• A fully connected layer with 1024 units and rectified linear activation.

• A dropout layer with 70% ratio of dropped outputs.

• A linear layer with softmax loss as the classifier (predicting the same 1000 classes as the main classifier, but removed at inference time)
Training

- CPU based implementation
- Asynchronous stochastic gradient descent with 0.9 momentum
- Fixed learning rate schedule (decreasing the learning rate by 4% every 8 epochs)
- Polyak averaging at inference time
- Sampling of various sized patches of the image whose size is distributed evenly between 8% and 100%
- Photometric distortions to combat overfitting
- Random interpolation methods (bilinear, area, nearest neighbor and cubic, with equal probability) for resizing
Classification Experimental Setup and Results

- 1000 leaf-node categories
- About 1.2 million images for training. 50,000 for validation and 100,000 images for testing
- Each image is associated with one ground truth category
- Performance is measured based on the highest scoring classifier predictions
Classification Experimental Setup and Results

- Main metrics are;
  - *top-1 accuracy rate*: compares the ground truth against the first predicted class
  - *top-5 error rate*: compares the ground truth against the first 5 predicted classes (image is correctly classified if the ground truth is among the top-5, regardless of its rank in them)

The challenge uses the top-5 error rate for ranking purposes
Classification Experimental Setup and Results

- Tricks and techniques;
  - *Ensemble*: 7 versions of the same GoogLeNet, trained with the same initialization & learning rate. Only differ in sampling methodologies and the random order in which they see input images.
  - *Data manipulation*: Aggressive cropping, resize the image to 4 scales (256, 288, 320 and 352) and take squares of these resized images. Result is $4 \times 3 \times 6 \times 2 = 144$ crops per image.
  - *Averaging*: softmax probabilities are averaged over multiple crops and over all the individual classifiers to obtain the final prediction.
## Classification results on ImageNet

<table>
<thead>
<tr>
<th>Number of Models</th>
<th>Number of Crops</th>
<th>Computational Cost</th>
<th>Top-5 Error</th>
<th>Compared to Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 (center crop)</td>
<td>1x</td>
<td>10.07%</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>10*</td>
<td>10x</td>
<td>9.15%</td>
<td>-0.92%</td>
</tr>
<tr>
<td>1</td>
<td>144 (Our approach)</td>
<td>144x</td>
<td>7.89%</td>
<td>-2.18%</td>
</tr>
<tr>
<td>7</td>
<td>1 (center crop)</td>
<td>7x</td>
<td>8.09%</td>
<td>-1.98%</td>
</tr>
<tr>
<td>7</td>
<td>10*</td>
<td>70x</td>
<td>7.62%</td>
<td>-2.45%</td>
</tr>
<tr>
<td>7</td>
<td>144 (Our approach)</td>
<td>1008x</td>
<td>6.67%</td>
<td>-3.41%</td>
</tr>
</tbody>
</table>

*Cropping by [Krizhevsky et al 2014]*
## Classification results on ImageNet

<table>
<thead>
<tr>
<th>Team</th>
<th>Year</th>
<th>Place</th>
<th>Error (top-5)</th>
<th>Uses external data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuperVision</td>
<td>2012</td>
<td>-</td>
<td>16.4%</td>
<td>no</td>
</tr>
<tr>
<td>SuperVision</td>
<td>2012</td>
<td>1st</td>
<td>15.3%</td>
<td>ImageNet 22k</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>-</td>
<td>11.7%</td>
<td>no</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>1st</td>
<td>11.2%</td>
<td>ImageNet 22k</td>
</tr>
<tr>
<td>MSRA</td>
<td>2014</td>
<td>3rd</td>
<td>7.35%</td>
<td>no</td>
</tr>
<tr>
<td>VGG</td>
<td>2014</td>
<td>2nd</td>
<td>7.32%</td>
<td>no</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>2014</td>
<td>1st</td>
<td>6.67%</td>
<td>no</td>
</tr>
</tbody>
</table>
Detection Experimental Setup and Results

- Produce bounding boxes around objects in images
- 200 possible classes.
- Detected objects count as correct if they match the class of the groundtruth and their bounding boxes overlap by at least 50%
- Extraneous detections count as false positives and are penalized
- Each image may contain many objects or none, and their scale may vary from large to tiny
Detection Experimental Setup and Results

• Tricks and techniques;
  – Similar to R-CNN, Inception model as the region classifier
  – Selective Search approach combined with multi-box predictions
  – Superpixel size was increased by 2x in order to decrease false positives
  – Ensemble of 6 ConvNets
# Detection results without ensembling

<table>
<thead>
<tr>
<th>Team</th>
<th>mAP</th>
<th>external data</th>
<th>contextual model</th>
<th>bounding-box regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trimps-Soushen</td>
<td>31.6%</td>
<td>ILSVRC12 Classification</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>Berkeley Vision</td>
<td>34.5%</td>
<td>ILSVRC12 Classification</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>UvA-Euvision</td>
<td>35.4%</td>
<td>ILSVRC12 Classification</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>CUHK DeepID-Net2</td>
<td>37.7%</td>
<td>ILSVRC12 Classification+ Localization</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>38.0%</td>
<td>ILSVRC12 Classification</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Deep Insight</td>
<td>40.2%</td>
<td>ILSVRC12 Classification</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
## Final Detection Results

<table>
<thead>
<tr>
<th>Team</th>
<th>Year</th>
<th>Place</th>
<th>mAP</th>
<th>external data</th>
<th>ensemble</th>
<th>contextual model</th>
<th>approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>UvA-Euvision</td>
<td>2013</td>
<td>1st</td>
<td>22.6%</td>
<td>none</td>
<td>?</td>
<td>yes</td>
<td>Fisher vectors</td>
</tr>
<tr>
<td>Deep Insight</td>
<td>2014</td>
<td>3rd</td>
<td>40.5%</td>
<td>ILSVRC12 Classification + Localization</td>
<td>3 models</td>
<td>yes</td>
<td>ConvNet</td>
</tr>
<tr>
<td>CUHK DeepID-Net</td>
<td>2014</td>
<td>2nd</td>
<td>40.7%</td>
<td>ILSVRC12 Classification + Localization</td>
<td>?</td>
<td>no</td>
<td>ConvNet</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>2014</td>
<td>1st</td>
<td>43.9%</td>
<td>ILSVRC12 Classification</td>
<td>6 models</td>
<td>no</td>
<td>ConvNet</td>
</tr>
</tbody>
</table>
GoogLeNet vs State of the art

GoogLeNet

Zeiler-Fergus Architecture (1 tower)

Convolution
Pooling
Softmax
Other
Classification failure cases

Groundtruth: ????
Classification failure cases

Groundtruth: coffee mug
Classification failure cases

**Groundtruth:** coffee mug

**GoogLeNet:**
- table lamp
- lamp shade
- printer
- projector
- desktop computer
Classification failure cases

Groundtruth: ???
Classification failure cases

Groundtruth: Police car
Classification failure cases

**Groundtruth:** Police car  
**GoogLeNet:**  
- laptop  
- hair drier  
- binocular  
- ATM machine  
- seat belt
Classification failure cases

Groundtruth: ???
Classification failure cases

Groundtruth: hay
Classification failure cases

Groundtruth: **hay**

GoogLeNet:

- **sorrel (horse)**
- **hartebeest**
- **Arabian camel**
- **warthog**
- **gaselle**
Cons and doubts

- One must be cautious though: although the proposed architecture has become a success for computer vision, it is still questionable whether its quality can be attributed to the guiding principles that have lead to its construction.

- No specific training methodology.
Conclusion and future work

- Approximating the expected optimal sparse structure by readily available dense building blocks is a viable method for improving neural networks for computer vision
- Low computational requirements
- Thus, moving to sparser architectures is feasible and useful idea in general
- Future work: creating sparser and more refined structures in automated ways
Thanks

Questions?