Exploring convnet architectures
Deeper is better
Deeper is better

Challenge winner's accuracy

- 2010: Alexnet
- 2011: Alexnet
- 2012: Alexnet
- 2013: VGG16
- 2014: VGG16
The VGG pattern

• Every convolution is 3x3, padded by 1
• Every convolution followed by ReLU
• ConvNet is divided into “stages”
  • Layers within a stage: no subsampling
  • Subsampling by 2 at the end of each stage
• Layers within stage have same number of channels
• Every subsampling → double the number of channels
Example network
Deeper is better

Challenge winner's accuracy

Can we go deeper?

Alexnet

VGG16
Is deeper better?

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena
Challenges in training: exploding / vanishing gradients

• Vanishing / exploding gradients

\[
\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \ldots \frac{\partial z_{i+1}}{\partial z_i}
\]

• If each term is (much) greater than 1 \(\rightarrow\) explosion of gradients

• If each term is (much) less than 1 \(\rightarrow\) vanishing gradients
Challenges in training: noisy gradients

- Vanishing / exploding gradients

\[
\frac{\partial z}{\partial z_i} = \frac{\partial z}{\partial z_{n-1}} \frac{\partial z_{n-1}}{\partial z_{n-2}} \ldots \frac{\partial z_{i+1}}{\partial z_i}
\]

- Gradient for i-th layer depends on all subsequent layers

- But subsequent layers are initially random
  - Implies noisy gradients for earlier layers
Residual connections

- Instead of:
  \[ z_{i+1} = f_{i+1}(z_i, w_{i+1}) \]

- We will have:
  \[ z_{i+1} = g_{i+1}(z_i, w_{i+1}) + z_i \]
With and without residual connections

- Without residual connections

\[ z_{i+1} = f_{i+1}(z_i, w_{i+1}) \]

\[ \frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial f_{i+1}(z_i, w_{i+1})}{\partial z_i} \]

- With residual connections

\[ z_{i+1} = g_{i+1}(z_i, w_{i+1}) + z_i \]

\[ \frac{\partial z_{i+1}}{\partial z_i} = \frac{\partial g_{i+1}(z_i, w_{i+1})}{\partial z_i} + I \]
Residual block
Residual connections

• Assumes all $z_i$ have the same size
• True within a stage
• Across stages?
  • Doubling of feature channels
  • Subsampling
• Increase channels by 1x1 convolution
• Decrease spatial resolution by subsampling

$$z_{i+1} = g_{i+1}(z_i, w_{i+1}) + \text{subsample}(Wz_i)$$
The ResNet pattern

• Decrease resolution substantially in first layer
  • Reduces memory consumption due to intermediate outputs

• Divide into stages
  • maintain resolution, channels in each stage
  • halve resolution, double channels between stages

• Divide each stage into residual blocks

• At the end, compute average value of each channel to feed linear classifier
Putting it all together - Residual networks

Challenge winner's accuracy

- 2010
- 2011
- 2012
- 2013
- 2014
- 2015

0 5 10 15 20 25 30

0 50 100 150 200
Transfer learning with convolutional networks

Trained feature extractor

Linear classifier

Horse
Transfer learning with convolutional networks

• What do we do for a new image classification problem?

• Key idea:
  • *Freeze* parameters in feature extractor
  • *Retrain* classifier
Transfer learning with convolutional networks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best Non-Convnet perf</th>
<th>Pretrained convnet + classifier</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech 101</td>
<td>84.3</td>
<td>87.7</td>
<td>+3.4</td>
</tr>
<tr>
<td>VOC 2007</td>
<td>61.7</td>
<td>79.7</td>
<td>+18</td>
</tr>
<tr>
<td>CUB 200</td>
<td>18.8</td>
<td>61.0</td>
<td>+42.2</td>
</tr>
<tr>
<td>Aircraft</td>
<td>61.0</td>
<td>45.0</td>
<td>-16</td>
</tr>
<tr>
<td>Cars</td>
<td>59.2</td>
<td>36.5</td>
<td>-22.7</td>
</tr>
</tbody>
</table>
Why transfer learning?

• Availability of training data

• Computational cost

• Ability to pre-compute feature vectors and use for multiple tasks

• *Con*: NO end-to-end learning
Finetuning
Finetuning

Initialize with pre-trained, then train with low learning rate
## Finetuning

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best Non-Convnet perf</th>
<th>Pretrained convnet + classifier</th>
<th>Finetuned convnet</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech 101</td>
<td>84.3</td>
<td>87.7</td>
<td>88.4</td>
<td>+4.1</td>
</tr>
<tr>
<td>VOC 2007</td>
<td>61.7</td>
<td>79.7</td>
<td>82.4</td>
<td>+20.7</td>
</tr>
<tr>
<td>CUB 200</td>
<td>18.8</td>
<td>61.0</td>
<td>70.4</td>
<td>+51.6</td>
</tr>
<tr>
<td>Aircraft</td>
<td>61.0</td>
<td>45.0</td>
<td>74.1</td>
<td>+13.1</td>
</tr>
<tr>
<td>Cars</td>
<td>59.2</td>
<td>36.5</td>
<td>79.8</td>
<td>+20.6</td>
</tr>
</tbody>
</table>