Survey of DNN Development Resources

MICRO Tutorial (2016)
Website: http://eyeriss.mit.edu/tutorial.html
Joel Emer, Vivienne Sze, Yu-Hsin Chen
Popular DNNs

- LeNet (1998)
- AlexNet (2012)
- OverFeat (2013)
- VGGNet (2014)
- GoogleNet (2014)

ImageNet: Large Scale Visual Recognition Challenge (ILSVRC)

Accuracy (Top 5 error)

[O. Russakovsky et al., IJCV 2015]
LeNet-5

CONV Layers: 2
Fully Connected Layers: 2
Weights: 431k
MACs: 2.3M

Digit Classification!

[Y. Lecun et al, Proceedings of the IEEE, 1998]
AlexNet

CONV Layers: 5
Fully Connected Layers: 3
Weights: 61M
MACs: 724M

Uses Local Response Normalization (LRN)

ILSCVR12 Winner

[Krizhevsky et al., NIPS, 2012]
### AlexNet Convolutional Layer Configurations

<table>
<thead>
<tr>
<th>Layer</th>
<th>Filter Size (RxS)</th>
<th># Filters (M)</th>
<th># Channels (C)</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11x11</td>
<td>96</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5x5</td>
<td>256</td>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3x3</td>
<td>384</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3x3</td>
<td>384</td>
<td>192</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3x3</td>
<td>256</td>
<td>192</td>
<td>1</td>
</tr>
</tbody>
</table>

Layer 1
- 34k Params
- 105M MACs

Layer 2
- 307k Params
- 224M MACs

Layer 3
- 885k Params
- 150M MACs

[Krizhevsky et al., NIPS, 2012]
**OverFeat (fast model)**

CONV Layers: 5  
Fully Connected Layers: 3  
Weights: 144M  
MACs: 5.4G

<table>
<thead>
<tr>
<th>Layer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
<td>conv + max</td>
<td>conv + max</td>
<td>conv</td>
<td>conv</td>
<td>conv + max</td>
<td>full</td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td># channels</td>
<td>96</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>1024</td>
<td>3072</td>
<td>4096</td>
<td>1000</td>
</tr>
<tr>
<td>Filter size</td>
<td>11x11</td>
<td>5x5</td>
<td>3x3</td>
<td>3x3</td>
<td>3x3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Conv. stride</td>
<td>4x4</td>
<td>1x1</td>
<td>1x1</td>
<td>1x1</td>
<td>1x1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pooling stride</td>
<td>2x2</td>
<td>2x2</td>
<td>-</td>
<td>-</td>
<td>2x2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pooling size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zero-Padding size</td>
<td>-</td>
<td>-</td>
<td>1x1x1x1</td>
<td>1x1x1x1</td>
<td>1x1x1x1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Spatial input size</td>
<td>231x231</td>
<td>24x24</td>
<td>12x12</td>
<td>12x12</td>
<td>12x12</td>
<td>6x6</td>
<td>1x1</td>
<td>1x1</td>
</tr>
</tbody>
</table>

[Sermanet et al., ArXiv 2013, ICLR 2014]
VGG-16

CONV Layers: 16
Fully Connected Layers: 3
Weights: 138M
MACs: 15.5G

Also, 19 layer version
Reduce # of weights

More Layers → Deeper!

Image Source: http://www.cs.toronto.edu/~frossard/post/vgg16/

[Simonyan et al., ArXiv 2014, ICLR 2015]
GoogLeNet (v1)

CONV Layers: 21
Fully Connected Layers: 1
Weights: 7.0M
MACs: 1.43G

Also, v2, v3 and v4
ILSVRC14 Winner

Inception module

[Szegedy et al., ArXiv 2014, CVPR 2015]
ResNet-50

CONV Layers: 49
Fully Connected Layers: 1
Weights: 25.5M
MACs: 3.9G

Also, 34,152 and 1202 layer versions
ILSVRC15 Winner

\[
F(x) \xrightarrow{\text{relu}} \text{weight layer} \xrightarrow{\text{relu}} \text{weight layer} \xrightarrow{+} x \xrightarrow{\text{relu}} H(x) = F(x) + x
\]

[He et al., ArXiv 2015, CVPR 2016]
Revolution of Depth

## Summary of Popular DNNs

<table>
<thead>
<tr>
<th>Metrics</th>
<th>LeNet-5</th>
<th>AlexNet</th>
<th>OverFeat (fast)</th>
<th>VGG-16</th>
<th>GoogLeNet (v1)</th>
<th>ResNet-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-5 error</td>
<td>n/a</td>
<td>16.4</td>
<td>14.2</td>
<td>7.4</td>
<td>6.7</td>
<td>5.3</td>
</tr>
<tr>
<td>Input Size</td>
<td>28x28</td>
<td>227x227</td>
<td>231x231</td>
<td>224x224</td>
<td>224x224</td>
<td>224x224</td>
</tr>
<tr>
<td># of CONV Layers</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>16</td>
<td>21</td>
<td>49</td>
</tr>
<tr>
<td>Filter Sizes</td>
<td>5</td>
<td>3, 5, 11</td>
<td>3, 7</td>
<td>3</td>
<td>1, 3, 5, 7</td>
<td>1, 3, 7</td>
</tr>
<tr>
<td># of Channels</td>
<td>1, 6</td>
<td>3 - 256</td>
<td>3 - 1024</td>
<td>3 - 512</td>
<td>3 - 1024</td>
<td>3 - 2048</td>
</tr>
<tr>
<td># of Filters</td>
<td>6, 16</td>
<td>96 - 384</td>
<td>96 - 1024</td>
<td>64 - 512</td>
<td>64 - 384</td>
<td>64 - 2048</td>
</tr>
<tr>
<td>Stride</td>
<td>1</td>
<td>1, 4</td>
<td>1, 4</td>
<td>1</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td># of Weights</td>
<td>26k</td>
<td>2.3M</td>
<td>16M</td>
<td>14.7M</td>
<td>6.0M</td>
<td>23.5M</td>
</tr>
<tr>
<td># of MACs</td>
<td>1.9M</td>
<td>666M</td>
<td>2.67G</td>
<td>15.3G</td>
<td>1.43G</td>
<td>3.86G</td>
</tr>
<tr>
<td># of FC layers</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># of Weights</td>
<td>406k</td>
<td>58.6M</td>
<td>130M</td>
<td>124M</td>
<td>1M</td>
<td>2M</td>
</tr>
<tr>
<td># of MACs</td>
<td>405k</td>
<td>58.6M</td>
<td>130M</td>
<td>124M</td>
<td>1M</td>
<td>2M</td>
</tr>
<tr>
<td>Total Weights</td>
<td>431k</td>
<td>61M</td>
<td>146M</td>
<td>138M</td>
<td>7M</td>
<td>25.5M</td>
</tr>
<tr>
<td>Total MACs</td>
<td>2.3M</td>
<td>724M</td>
<td>2.8G</td>
<td>15.5G</td>
<td>1.43G</td>
<td>3.9G</td>
</tr>
</tbody>
</table>

CONV Layers increasingly important!
Summary of Popular DNNs

• AlexNet
  – First CNN Winner of ILSVRC
  – Uses LRN (deprecated after this)

• VGG-16
  – Goes Deeper (16+ layers)
  – Uses only 3x3 filters (stack for larger filters)

• GoogLeNet (v1)
  – Reduces weights with Inception and only one FC layer
  – Inception: 1x1 and DAG (parallel connections)
  – Batch Normalization

• ResNet
  – Goes Deeper (24+ layers)
  – Shortcut connections
Frameworks

Caffe
Berkeley / BVLC
(C, C++, Python, MATLAB)

TensorFlow
Google
(C++, Python)

Theano
U. Montreal
(Python)

torch
Facebook / NYU
(C, C++, Lua)

Also, CNTK, MXNet, etc.
More at: https://developer.nvidia.com/deep-learning-frameworks
Example: Layers in Caffe

Convolution Layer

```
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
...  
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
...  
```

Non-Linearity

```
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "conv1"
  top: "conv1"
}
```

Pooling Layer

```
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2 ...
```

[http://caffe.berkeleyvision.org/tutorial/layers.html](http://caffe.berkeleyvision.org/tutorial/layers.html)
Image Classification Datasets

• **Image Classification/Recognition**
  – Given an entire image → Select 1 of N classes
  – No localization (detection)

Datasets affect difficulty of task
MNIST

Digit Classification
28x28 pixels (B&W)
10 Classes
60,000 Training
10,000 Testing

LeNet in 1998
(0.95% error)

ICML 2013
(0.21% error)

http://yann.lecun.com/exdb/mnist/
CIFAR-10/CIFAR-100

Object Classification
32x32 pixels (color)
10 or 100 Classes
50,000 Training
10,000 Testing

CIFAR-10
RBM+finetuning in 2009
(35.16% error)

ArXiv 2015
(3.47% error)

Subset of 80 Tiny Images Dataset (Torrabla)

https://www.cs.toronto.edu/~kriz/cifar.html
Object Classification
~256x256 pixels (color)
1000 Classes
1.3M Training
100,000 Testing (50,000 Validation)

Image Source: http://karpathy.github.io/
Fine grained Classes (120 breeds)

Top-5 Error
Winner 2012 (16.42% error)
Winner 2016 (2.99% error)

Image Source: http://www.image-net.org/challenges/LSVRC/

Image Source: http://karpathy.github.io/

Image Source: Krizhevsky et al., NIPS 2012
## Image Classification Summary

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>IMAGENET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Year</strong></td>
<td>1998</td>
<td>2009</td>
<td>2009</td>
<td>2012</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>28x28</td>
<td>32x32</td>
<td>32x32</td>
<td>256x256</td>
</tr>
<tr>
<td><strong>Classes</strong></td>
<td>10</td>
<td>10</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>60k</td>
<td>50k</td>
<td>50k</td>
<td>1.3M</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>10k</td>
<td>10k</td>
<td>10k</td>
<td>100k</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>0.21% error (ICML 2013)</td>
<td>3.47% error (arXiv 2015)</td>
<td>24.28% error (arXiv 2015)</td>
<td>2.99% top-5 error (2016 winner)</td>
</tr>
</tbody>
</table>

[http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html](http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html)
Next Tasks: Localization and Detection

- **Image classification**
  - Ground truth: Steel drum
  - Accuracy: 1
  - Steel drum, Folding chair, Loudspeaker
  - Scale, T-shirt, Steel drum, Drumstick, Mud turtle
  - Scale, T-shirt, Giant panda, Drumstick, Mud turtle

- **Single-object localization**
  - Ground truth: Steel drum
  - Accuracy: 1
  - Steel drum, Folding chair, Picket fence, Screwdriver
  - Steel drum, Folding chair, Picket fence, Microphone
  - Steel drum, Folding chair, Person, Microphone

- **Object detection**
  - Ground truth: Steel drum, Person, Folding chair, Microphone
  - AP: 1.0 1.0 1.0 1.0
  - AP: 0.0 0.5 1.0 0.3
  - AP: 1.0 0.7 0.5 0.9

[Russakovsky et al., IJCV, 2015]
Others Popular Datasets

- **Pascal VOC**
  - 11k images
  - Object Detection
  - 20 classes

- **MS COCO**
  - 300k images
  - Detection, Segmentation
  - Recognition in context

http://host.robots.ox.ac.uk/pascal/VOC/
http://mscoco.org/
Recently Introduced Datasets

• Announced Sept 2016:
  • Google Open Images (~9M images)
    – https://github.com/openimages/dataset
  • Youtube-8M (8M videos)
    – https://research.google.com/youtube8m/