Flexible, High Performance Convolutional Neural Networks for Image Classification

Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Jürgen Schmidhuber

IDSIA, USI and SUPSI
Manno-Lugano, Switzerland
{dan,ueli,jonathan,luca,juergen}@idsia.ch
Introduction

- Image recognition: text, objects.
- Convolutional Neural Networks.
- How to train huge nets? GPUs ...
- Evaluation protocol for standard benchmarks.
- Other applications.
Multi Layer Perceptrons

- simple, uniform architecture
- fully connected
- many weights, one weight per connection
- very general
- disregards 2D information (neighboring info)
- not good for classifying complex images
- We trained the first big and deep MLP, breaking the record on MNIST (Neural Computation 2010)
Convolutional Neural Networks (CNNs)

- Hierarchical architecture designed for image recognition, loosely inspired from biology.
- Introduced by Fukushima (80) and refined by LeCun et al.(98), Riesenhuber et al.(99), Simard et al.(03), Behnke (03).
- Uses the neighboring information (preserves the 2D information).
- Shared weights.
- Fully supervised, with randomly initialized filters, trained minimizing the misclassification error.
- Flexible architecture.

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Convolutional Neural Networks (CNNs)

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Convolutional Neural Networks (CNNs)

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Convolutional layer

- 3x3 kernel (filter) => only 10 weights (3x3+1) shared by all 900 neurons
- each neuron has 9 source neurons
- 900 x (9+1) = 9000 connections
Convolutional layer

32x32  
L0-Input

9x9  
Filters

24x24  
L1-Convolutional Layer

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Convolutional layer

48x48

9x9

40x40

L0-Input

Filters

L1-Convolutional Layer

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Max-pooling layer

- Introduces small translation invariance
- Improves generalization
Max-pooling layer

- Introduces small translation invariance
- Improves generalization
Max-pooling layer

- Introduces small translation invariance
- Improves generalization
Fully connected layer

- One output neuron per class normalized with soft-max activation function
Fully connected layer

- One output neuron per class normalized with soft-max activation function
Training is computational intensive
Graphics processing units (GPUs)

- 8 x GTX 480/580 1.5GB RAM
- >12 TFLOPS (theoretical speed)
- 40-80x speed-up compared with a single threaded CPU version of the CNN program (one day on GPU instead of two months on CPU)
- ~3000$ for one workstation with 4 GPUs
Back-propagation of errors

- Uses pooling of errors (deltas).
Experiments

• Distorting images

• Datasets:
  – Handwritten digits: MNIST
  – Handwritten Latin characters: NIST SD 19
  – Handwritten Chinese characters
  – 3D models: NORB
  – Natural images: CIFAR10
  – Traffic signs
Distortions

- **MNIST**
  - affine: translation, rotation, scaling
  - elastic

<table>
<thead>
<tr>
<th>Original</th>
<th>Epoch 0</th>
<th>Epoch 1</th>
<th>Epoch 2</th>
<th>Epoch 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 0 4 1 9 2 1 3 1 4 3 5 3 6 1 7 2 8 6 9</td>
<td>5 0 4 1 9 2 1 3 1 4 3 5 3 6 1 7 2 8 6 9</td>
<td>5 0 4 1 9 2 1 3 1 4 3 5 3 6 1 7 2 8 6 9</td>
<td>5 0 4 1 9 2 1 3 1 4 3 5 3 6 1 7 2 8 6 9</td>
<td></td>
</tr>
</tbody>
</table>

- Elastic only for characters and digits

- Greatly improves generalization and recognition rate

- CIFAR10/traffic signs have non-uniform backgrounds => border effects
**MNIST**

- Very competitive dataset
- Handwritten digits
- 28x28 grayscale images
- 60000 for training and 10000 for testing
- Tens of papers: http://yann.lecun.com/exdb/mnist/
MNIST

- Simard et al. (2003) – 0.40%, Ciresan et al. (2010) – 0.35% (big deep MLP)
- Big deep CNN – 0.35% (2011), far less weights than the MLP
- 30 out of 35 digits have a correct second prediction

<table>
<thead>
<tr>
<th>#M, #N in Hidden Layers</th>
<th>Test error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>20M-60M</td>
<td>1.02</td>
</tr>
<tr>
<td>20M-60M-150N</td>
<td>0.55</td>
</tr>
<tr>
<td>20M-60M-100M-150N</td>
<td>0.38</td>
</tr>
<tr>
<td><strong>20M-40M-60M-80M-100M-120M-150N</strong></td>
<td><strong>0.35</strong></td>
</tr>
</tbody>
</table>

35 errors
## NIST SD 19

- More than 800000 handwritten digits and letters
- our CNN have state of the art results on all tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Test error [?]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small letters</td>
<td>7.71</td>
</tr>
<tr>
<td>Big letters</td>
<td>1.91</td>
</tr>
<tr>
<td>Case insensitive</td>
<td>7.58</td>
</tr>
<tr>
<td>Merged letters (37 classes)</td>
<td>8.21</td>
</tr>
</tbody>
</table>
Small NORB

- 48600 96x96 stereo images
- 5 classes with 10 instances
- 5 instances for training and 5 for testing
- bad/challenging dataset, only 5 instances/class, some instances from test set are completely different than the one from training set
- IP maps (Mexican hat) are needed only for this data set
- previous state of the art: Behnke et al. 2.87%
- 40% of the errors are cars erroneously classified as trucks

<table>
<thead>
<tr>
<th>translation [%]</th>
<th>IP maps</th>
<th>test [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no</td>
<td>7.86 ± 0.55</td>
</tr>
<tr>
<td>5</td>
<td>no</td>
<td>4.71 ± 0.57</td>
</tr>
<tr>
<td>0</td>
<td>yes</td>
<td>3.94 ± 0.48</td>
</tr>
<tr>
<td>5</td>
<td>yes</td>
<td><strong>2.53 ± 0.40</strong></td>
</tr>
</tbody>
</table>

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
CIFAR10

- small, 32x32 pixels color images
- complex backgrounds
- 10 classes
- 50000 training images
- 10000 test images

<table>
<thead>
<tr>
<th>trans. [M]</th>
<th>IP maps</th>
<th>TfbV [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0; 100M</td>
<td>no</td>
<td>28.87 ± 0.37</td>
</tr>
<tr>
<td>0; 100M</td>
<td>edge</td>
<td>29.11 ± 0.36</td>
</tr>
<tr>
<td>5; 100M</td>
<td>no</td>
<td>20.26 ± 0.21</td>
</tr>
<tr>
<td>5; 100M</td>
<td>edge</td>
<td>21.87 ± 0.57</td>
</tr>
<tr>
<td>5; 100M</td>
<td>hat</td>
<td>21.44 ± 0.44</td>
</tr>
<tr>
<td>5; 200M</td>
<td>no</td>
<td>19.90 ± 0.16</td>
</tr>
<tr>
<td><strong>5; 300M</strong></td>
<td>no</td>
<td><strong>19.51 ± 0.18</strong></td>
</tr>
<tr>
<td>5; 400M</td>
<td>no</td>
<td>19.54 ± 0.16</td>
</tr>
</tbody>
</table>

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
Committees

- Extremely simple idea
- Easy to compute
- Averaging the corresponding outputs of many nets
- Decrease the error with 20-80%
- Work better with preprocessing in case of handwritten characters
Chinese Handwriting Recognition Competition ICDAR 2011

- Offline Chinese Character Recognition (Task 1)
- 3GB of data
- 3755 classes
- >1M characters
- 48x48 pixels grayscale images
- 270 samples / class
- 9 teams
- No knowledge of Chinese
- First place at ICDAR 2011 competition
  - 92.18% first prediction is correct
  - 99.29% CR10

Flexible, High Performance Convolutional Neural Networks for Image Classification; D. Ciresan et al.
German Traffic Sign Recognition Competition

- Initial phase:
  - More than 10 teams
  - We won 1st place with 98.98%

- 40000 color images 15x15 to 250x250

- 43 different signs: 26640 training images, 12569 testing images

Input:
- Extract Region of Interest (ROI)
- Enhance contrast with four different methods
- Resize up/down to 48x48 pixels
- Great variation in details, contrast and illumination

http://benchmark.ini.rub.de/

- Final phase: last Wednesday at IJCNN 2011
New normalizations

A Committee of Neural Networks for Traffic Sign Classification; D. Ciresan et al.
Normalized images

- original
- imadjust
- histeq
- clahe
- contrast
Distortions

- At the beginning of every epoch, each image in the training set is distorted
- The original training set is used only for validation
- Type:
  - Translation (random for both axes, maximum 10%)
  - Rotation (random, maximum 10 degrees)
  - Scaling (random for both axes, maximum 10%)
- Improves generalization and recognition rate on test set
- Non-uniform backgrounds => border effects

A Committee of Neural Networks for Traffic Sign Classification; D. Ciresan et al.
CNN trained on preprocessed color images

- Original + 4 preprocessed datasets
- Train 5 nets for each
- Build a 25-net committee
- First place with 99.43% recognition rate
- The only team with better than human (98.84%) results

A Committee of Neural Networks for Traffic Sign Classification; D. Ciresan et al.
## Summary

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best result by others (error [%])</th>
<th>Our result (error [%])</th>
<th>Decrease [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>0.39</td>
<td>0.23</td>
<td>41%</td>
</tr>
<tr>
<td>NIST SD 19</td>
<td>all tasks</td>
<td>all tasks</td>
<td>20-80%</td>
</tr>
<tr>
<td>Chinese</td>
<td>10.01</td>
<td>6.5</td>
<td>35%</td>
</tr>
<tr>
<td>NORB</td>
<td>2.87</td>
<td>2.01</td>
<td>30%</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>18.81</td>
<td>11.75</td>
<td>37%</td>
</tr>
<tr>
<td>Traffic signs</td>
<td>1.69</td>
<td>0.54</td>
<td>72%</td>
</tr>
</tbody>
</table>
Image segmentation

Red – cubes
Green – cylinders
White – background
Searching for objects – adding a negative class with partial cubes and cylinders
Detecting cell contours
Detecting cells
Conclusions

• Big deep nets combining CNN and other ideas are now state of the art for many image classification tasks.
• No need to extract handcrafted features.
• Supervised training with simple gradient descent training is best. No need for unsupervised pre-training (e.g. autoencoders) in case of sufficient training samples.
• Distorting the training set improves recognition rate on unseen data.
• CPUs are not enough anymore, use GPUs which are 2 orders of magnitude faster.
• Robust (smallest error rates) and fast enough ($10^3$-$10^4$ images/s) for immediate industrial application.
What is next?

- Test the CNNs on different datasets:
  - CALTECH 101 & 256, ImageNet, cluttered NORB
  - medical images
- Use CNNs for general scene understanding (segmentation).
- Robot vision applications (detect type, position and orientation of objects in a scene).
- Computer vision applications (de-blurring, segmentation, similarity check etc.).
- Try to reach human image classification performance on more datasets.
- Add unsupervised pre-training.
- More at www.idsia.ch/~ciresan
Publications

Conference papers:
- Convolutional Neural Network Committees For Handwritten Character Classification (D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, ICDAR 2011)
- Better digit recognition with a committee of simple Neural Nets (U. Meier, D. Ciresan, L. M. Gambardella, J. Schmidhuber, ICDAR 2011)

Journal papers:

Under review:
- Democratic Committee of Fast Deep MLPs Improves MNIST Recognition Rate (D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, Neural Computation)
\[
\begin{align*}
\max \left( \left\lfloor \frac{i - K_x + 1}{S_x + 1} \right\rfloor, 0 \right) & \leq x \leq \min \left( \left\lceil \frac{i - K_x}{S_x + 1} \right\rceil, M_x - 1 \right) \\
\max \left( \left\lfloor \frac{i - K_y + 1}{S_y + 1} \right\rfloor, 0 \right) & \leq y \leq \min \left( \left\lceil \frac{i - K_y}{S_y + 1} \right\rceil, M_y - 1 \right)
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>mammal</th>
<th>human</th>
<th>plane</th>
<th>truck</th>
<th>car</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>mammal</td>
<td>0</td>
<td>22</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>human</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>plane</td>
<td>88</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>37</td>
<td>185</td>
</tr>
<tr>
<td>truck</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>26</td>
</tr>
<tr>
<td>car</td>
<td>79</td>
<td>0</td>
<td>0</td>
<td>246</td>
<td>0</td>
<td>325</td>
</tr>
</tbody>
</table>