Implementing Deep Neural Networks with Non Volatile Memories

Olivier Bichler¹ (olivier.bichler@cea.fr)
Daniele Garbin²
Elisa Vianello²
Luca Perniola²
Barbara DeSalvo²
Christian Gamrat¹

¹ CEA, LIST, Laboratory for Enhancing Reliability of Embedded Systems
² CEA, LETI
Summary

- Context

- **Opportunity** – Deep Neural Networks

- **Challenge** – The Memory Bottleneck

- **Paradigm Shift** – Spiking, NVM-based Networks

- Related Developments

- Perspectives
Internet of (Smart?) Things
ImageNet classification (Hinton’s team, hired by Google) [1]
- 1.2 million high res images, 1,000 different classes
- Top-5 17% error rate (huge improvement)

Facebook’s ‘DeepFace’ Program (labs head: Y. LeCun) [2]
- 4 million images, 4,000 identities
- 97.25% accuracy, vs. 97.53% human performance
# State-of-the-art in Recognition

<table>
<thead>
<tr>
<th>Database</th>
<th># Images</th>
<th># Classes</th>
<th>Best score</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNSIT</td>
<td>60,000 + 10,000</td>
<td>10</td>
<td>99.79% [3]</td>
</tr>
<tr>
<td>Handwritten digits</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTSRB</td>
<td>~ 50,000</td>
<td>43</td>
<td>99.46% [4]</td>
</tr>
<tr>
<td>Traffic sign</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>50,000 + 10,000</td>
<td>10</td>
<td>91.2% [5]</td>
</tr>
<tr>
<td>airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caltech-101</td>
<td>~ 50,000</td>
<td>101</td>
<td>86.5% [6]</td>
</tr>
<tr>
<td>ImageNet</td>
<td>~ 1,000,000</td>
<td>1,000</td>
<td>Top-5 83% [1]</td>
</tr>
<tr>
<td>DeepFace</td>
<td>~ 4,000,000</td>
<td>4,000</td>
<td>97.25% [2]</td>
</tr>
</tbody>
</table>

- State-of-the-art are Deep Neural Networks *every time*
Main Actors at International Level

**ACADEMICS**

- Stanford University
  - Deep learning
  - Andrew Ng
  - NVM-based architectures
  - H.-S. P. Wong

- University of Toronto
  - Deep learning
  - G. Hinton
  - A. Krizhevsky

- NYU
  - Deep learning
  - Overfeat, Torch
  - Y. LeCun

- UCSB
  - RRAM-based architectures
  - D. Strukov

- KAIST
  - Deep learning
  - J. Schmidhuber

- Michigan
  - Specialized architectures
  - H.-J. Yoo

- Leti & List
  - NVM-based architectures
  - Wei Lu

**INDUSTRIALS**

- Google
  - Deep learning
  - G. Hinton
  - O. Temam

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
  - Speech Recognition
  - DBN, RNN
  - R. Sarikaya
  - G. E. Dahl
  - Project Adam

- Qualcomm
  - Zeroth chip
  - nn-X
  - FPGA / GPU
  - Cloud
  - Y. LeCun
  - C. Farabet

- Synaptics
  - Memristor / RRAM
  - R. S. Williams

- LABS
  - RRAM-based architectures
  - S. Park

- Intel
  - NVM-based architectures
  - Wei Lu

- Samsung
  - Specialized architectures
  - H.-J. Yoo

- Synaptics
  - RRAM-based architectures
  - S. Park

- Madbits
  - Deep learning
  - C. Farabet

- Brain Corporation
  - E. M. Izhikevich
  - BrainOS

- GlobalSensing Technologies
  - NeuroDSP chip
  - Cognimem chip

- IDEAS
  - NeuroDSP chip
  - Cognimem chip

- SPIKENET
  - NeuroDSP chip
  - Cognimem chip

- Synaptics
  - RRAM-based architectures
  - S. Park

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
  - Speech Recognition
  - DBN, RNN
  - R. Sarikaya
  - G. E. Dahl
  - Project Adam

- Qualcomm
  - Zeroth chip
  - nn-X
  - FPGA / GPU
  - Cloud
  - Y. LeCun
  - C. Farabet

- Synaptics
  - Memristor / RRAM
  - R. S. Williams

- LABS
  - RRAM-based architectures
  - S. Park

- Intel
  - NVM-based architectures
  - Wei Lu

- Samsung
  - Specialized architectures
  - H.-J. Yoo

- Synaptics
  - RRAM-based architectures
  - S. Park

- Madbits
  - Deep learning
  - C. Farabet

- Brain Corporation
  - E. M. Izhikevich
  - BrainOS

- GlobalSensing Technologies
  - NeuroDSP chip
  - Cognimem chip

- IDEAS
  - NeuroDSP chip
  - Cognimem chip

- SPIKENET
  - NeuroDSP chip
  - Cognimem chip

- Synaptics
  - RRAM-based architectures
  - S. Park

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
  - Speech Recognition
  - DBN, RNN
  - R. Sarikaya
  - G. E. Dahl
  - Project Adam

- Qualcomm
  - Zeroth chip
  - nn-X
  - FPGA / GPU
  - Cloud
  - Y. LeCun
  - C. Farabet

- Synaptics
  - Memristor / RRAM
  - R. S. Williams

- LABS
  - RRAM-based architectures
  - S. Park

- Intel
  - NVM-based architectures
  - Wei Lu

- Samsung
  - Specialized architectures
  - H.-J. Yoo

- Synaptics
  - RRAM-based architectures
  - S. Park

- Madbits
  - Deep learning
  - C. Farabet

- Brain Corporation
  - E. M. Izhikevich
  - BrainOS

- GlobalSensing Technologies
  - NeuroDSP chip
  - Cognimem chip

- IDEAS
  - NeuroDSP chip
  - Cognimem chip

- SPIKENET
  - NeuroDSP chip
  - Cognimem chip

- Synaptics
  - RRAM-based architectures
  - S. Park

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
  - Speech Recognition
  - DBN, RNN
  - R. Sarikaya
  - G. E. Dahl
  - Project Adam

- Qualcomm
  - Zeroth chip
  - nn-X
  - FPGA / GPU
  - Cloud
  - Y. LeCun
  - C. Farabet

- Synaptics
  - Memristor / RRAM
  - R. S. Williams

- LABS
  - RRAM-based architectures
  - S. Park

- Intel
  - NVM-based architectures
  - Wei Lu

- Samsung
  - Specialized architectures
  - H.-J. Yoo

- Synaptics
  - RRAM-based architectures
  - S. Park

- Madbits
  - Deep learning
  - C. Farabet

- Brain Corporation
  - E. M. Izhikevich
  - BrainOS

- GlobalSensing Technologies
  - NeuroDSP chip
  - Cognimem chip

- IDEAS
  - NeuroDSP chip
  - Cognimem chip

- SPIKENET
  - NeuroDSP chip
  - Cognimem chip

- Synaptics
  - RRAM-based architectures
  - S. Park

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
  - Speech Recognition
  - DBN, RNN
  - R. Sarikaya
  - G. E. Dahl
  - Project Adam

- Qualcomm
  - Zeroth chip
  - nn-X
  - FPGA / GPU
  - Cloud
  - Y. LeCun
  - C. Farabet

- Synaptics
  - Memristor / RRAM
  - R. S. Williams

- LABS
  - RRAM-based architectures
  - S. Park

- Intel
  - NVM-based architectures
  - Wei Lu

- Samsung
  - Specialized architectures
  - H.-J. Yoo

- Synaptics
  - RRAM-based architectures
  - S. Park

- Madbits
  - Deep learning
  - C. Farabet

- Brain Corporation
  - E. M. Izhikevich
  - BrainOS

- GlobalSensing Technologies
  - NeuroDSP chip
  - Cognimem chip

- IDEAS
  - NeuroDSP chip
  - Cognimem chip

- SPIKENET
  - NeuroDSP chip
  - Cognimem chip

- Synaptics
  - RRAM-based architectures
  - S. Park

- DARPA
  - TrueNorth chip
  - PCM-based architectures

- IBM
  - Deep learning
  - Y. LeCun
  - DeepFace

- Facebook
  - Deep learning
  - Y. LeCun

- Baidu
  - Deep learning
  - Andrew Ng

- Microsoft Research
# Deep Convolutional Networks

**CIFAR-10**

who is the best in CIFAR-10?

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>91.2%</td>
<td>Network In Network</td>
<td>ICLR 2014</td>
</tr>
<tr>
<td>90.68%</td>
<td>Regularization of Neural Networks using DropConnect</td>
<td>ICML 2013</td>
</tr>
<tr>
<td>90.65%</td>
<td>Maxout Networks</td>
<td>ICML 2013</td>
</tr>
<tr>
<td>90.61%</td>
<td>Improving Deep Neural Networks with Probabilistic Maxout Units</td>
<td>ICLR 2014</td>
</tr>
<tr>
<td>90.5%</td>
<td>Practical Bayesian Optimization of Machine Learning Algorithms</td>
<td>NIPS 2012</td>
</tr>
<tr>
<td>89%</td>
<td>ImageNet Classification with Deep Convolutional Neural Networks</td>
<td>NIPS 2012</td>
</tr>
<tr>
<td>88.79%</td>
<td>Multi-Column Deep Neural Networks for Image Classification</td>
<td>CVPR 2012</td>
</tr>
<tr>
<td>84.87%</td>
<td>Stochastic Pooling for Regularization of Deep Convolutional Neural Networks</td>
<td>arXiv 2013</td>
</tr>
<tr>
<td>84.4%</td>
<td>Improving neural networks by preventing co-adaptation of feature detectors</td>
<td>arXiv 2012</td>
</tr>
<tr>
<td>83.96%</td>
<td>Discriminative Learning of Sum-Product Networks</td>
<td>NIPS 2012</td>
</tr>
<tr>
<td>80.02%</td>
<td>Learning Smooth Pooling Regions for Visual Recognition</td>
<td>BMVC 2013</td>
</tr>
<tr>
<td>79.6%</td>
<td>An Analysis of Single-Layer Networks in Unsupervised Feature Learning</td>
<td>AISTATS 2011</td>
</tr>
</tbody>
</table>

Convolutional Neural Network (CNN) or similar topology

Source: Rodrigo Benenson github page [http://rodrigob.github.io/are_we_there_yet/build/](http://rodrigob.github.io/are_we_there_yet/build/)
Convolutional Layer

Convolution operation:

\[ O_{i,j} = \tanh \left( \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} I_{i+k,j+l} \cdot K_{k,l} \right) \]

Input map
\[ (I_{i,j} \text{ matrix}) \]

Output feature map
\[ (O_{i,j} \text{ matrix}) \]

Each kernel generates ≠ output feature maps

Kernels are learned with gradient-descent algorithms (classical back-propagation is very efficient!)
CNNs Organization

Deep = number of layers >> 1
The German Traffic Sign Recognition Benchmark (GTSRB)

43 traffic sign types
> 50,000 images

Neurons: 287,843
Synapses: 1,388,800
  – Total memory: 1.5MB (with 8 bits synapses)
Connections: 124,121,800


Near human recognition (> 98%) [3]
The Memory Bottleneck

$n \times n$ cycles per kernel + non-linearity computation
$\times$ output matrix size
$\times$ number of kernels in output feature map
$\times$ number of output feature maps in layer
$\times$ number of layers
The Memory Bottleneck: Solutions?

- **data level parallelism**
  - SIMD instructions: $\times 2 - \times 32$ acceleration
  - But... limited by the size of the memory bus

- **number of processing cores**
  - $\times$ (number of cores) acceleration (assuming distributed memory)
  - High-end GPU: $\times 100$ acceleration over CPU! (@ 250W power consumption...)

**Back to our example:**

- ~ 125MM MAC operations (for 48x48 pixels inputs)
  - 128 bit memory bus (SIMD x16)
  - 16 processing cores (distributed memory)
  - 500K cycles @ 200 Mhz = 2.5 ms / input

- ROIs extraction @ 30 frames/s
  - Time to process 12 ROIs / frame...

Highly specialized architectures required to envision embeddable systems
Is a Paradigm Shift Possible?

- Fully distributed, fully parallel? ➔ Compute in memory!
- MAC computation in memory?

- Input signal coding?
  - Voltage level: digital to analog converter
  - Pulse duration: pulse width modulation

- Non-linearity computation?
  - Analog computation
  - Look-up table

Voltage level: digital to analog converter
Pulse duration: pulse width modulation
Non-linearity computation?
Spike-based Neural Networks

- **Input signal: rate-based coding**
  - From 1 pulse to N pulses / input time slot
  - N \(\leftrightarrow\) precision of the input signal discretization
  - Tunability: energy consumption \(\propto\) N, applicative performances \(\propto\) N

- **Non-linearity: refractory period**
  - Approximates \(\tanh()\) with a piece-wise linear function [7]
  - Easy to implement, no applicative performance penalty!

- **Direct interface to bio-inspired sensors [8]:**


Spike-based Coding & Propagation

Pixel brightness

Rate-based input coding

Spiking frequency

29x29 pixels
841 addresses

Layer 1
Layer 2
Layer 3
Layer 4

Correct Output

Address

Time

© CEA. All rights reserved
Our Simulation Tools: Xnet

Example on the MNIST database

Deep network description file: network.ini

```
; Environment
[env]
SizeX=29
SizeY=29
ConfigSection=env.config

[env.config]
ImageScale=0

; First layer (convolutional)
[conv1]
Input=env
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=6
Stride=2
ConfigSection=common.config

; Second layer (convolutional)
[conv2]
Input=conv1
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=12
Stride=2
ConfigSection=common.config

; Third layer (fully connected)
[fc1]
Input=conv2
Type=Fc
NbOutputs=100
ConfigSection=common.config

; Output layer (fully connected)
[fc2]
Input=fc1
Type=Fc
NbOutputs=10
ConfigSection=common.config

; Common config for static model
[common.config]
NoBias=1
WeightsLearningRate=0.0005
Threshold=1.0
NoClamping=1
```

MNIST database (60000 images)

```
3 9 5 8 4
9 2 4 3 8
```

```
xnet_convnet network.ini mnist
-learn 6000000 -log 10000
```
Back-propagation Offline Learning

Simulated network topology for MNIST (auto-generated)

- **env**: 29x29
- **conv1**: 6 (13x13)
- **conv2**: 12 (5x5)
- **fc1**: 100
- **fc2**: 10

Learning and test performances

**Learning**
- Recogn. rate: 99.7%

**Test**
- Recogn. rate: 98.7%

Learned kernels for conv1 layer
Spike-based Read-only Network

Spiking propagation of one pattern

Spike-based test performances

0% performance drop vs. static network! 😊

Spike-based network statistics

<table>
<thead>
<tr>
<th>Layer</th>
<th>Synapses (shared)</th>
<th>Connections</th>
<th>Events/frames</th>
<th>Events/connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>150</td>
<td>25,350</td>
<td>36,666</td>
<td>1.45</td>
</tr>
<tr>
<td>conv2</td>
<td>1,800</td>
<td>45,000</td>
<td>173,278</td>
<td>3.85</td>
</tr>
<tr>
<td>fc1</td>
<td>30,000</td>
<td>30,000</td>
<td>226,859</td>
<td>7.56</td>
</tr>
<tr>
<td>fc2</td>
<td>1,000</td>
<td>1,000</td>
<td>8,037</td>
<td>8.04</td>
</tr>
</tbody>
</table>

Recogn. rate: 98.7%
Spike-based Networks with NVMs

**PCM**

- Crystallization/Amorphization

**CBRAM**

- Forming/Dissolution of conductive filament

---

From spiking pre-synaptic neurons (inputs) to spiking post-synaptic neuron (output)

<table>
<thead>
<tr>
<th>Equivalent 2-PCM synapse</th>
</tr>
</thead>
</table>

**Unsupervised cars trajectories extraction**

**Unsupervised MNIST handwritten digits classification with stochastic learning**

---


[10] M. Suri et al., “CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: Auditory (cochlea) and visual (retina) cognitive processing applications”, IEDM, 2012
Implementation with NVM Devices

- Spike-based computing principle

- Convolution kernel
  - Output neurons
  - Input neurons
  - Signal propagation
  - Input spike
  - Synaptic weighting of the spike (multi-level or binary RRAM device(s))

- Other convolution kernel(s)...

- CMOS dynamic interconnect
Application Benchmarking in Xnet (1)

- **env**
  - SizeX=48
  - SizeY=48
  - ConfigSection=env.config

- **env.config**
  - ImageScale=1

- **conv1_7x7**
  - Input=env
  - Type=Conv
  - KernelWidth=7
  - KernelHeight=7
  - NbChannels=4
  - Stride=1
  - Kernel=Gabor
  - Kernel.Gamma=0.3
  - Kernel[0].Theta=0.0
  - Kernel[0][1].Theta=45.0
  - Kernel[0][2].Theta=90.0
  - Kernel[0][3].Theta=135.0
  - ConfigSection=common_fixed.config

- **conv1_9x9**
  - Input=env
  - Type=Conv
  - KernelWidth=9
  - KernelHeight=9
  - NbChannels=4
  - Stride=1
  - Kernel=Gabor
  - Kernel.Gamma=0.3
  - Kernel[0][0].Theta=0.0
  - Kernel[0][1].Theta=45.0
  - Kernel[0][2].Theta=90.0
  - Kernel[0][3].Theta=135.0
  - ConfigSection=common_fixed.config

- **pool1**
  - Input=conv1_7x7,conv1_9x9
  - Type=Pool
  - PoolWidth=8
  - PoolHeight=8
  - NbChannels=8
  - Stride=4
  - Pooling=Max
  - Mapping.Size=1
  - Mapping.NbIterations=4

- **fc1**
  - Input=pool1
  - Type=Fc
  - NbOutputs=20
  - ConfigSection=common.config

- **fc2**
  - Input=fc1
  - Type=Fc
  - NbOutputs=2
  - ConfigSection=common.config

- **[common_fixed.config]**
  - NoBias=1
  - WeightsLearningRate=0.0
  - BiasLearningRate=0.0
  - NoClamping=1

**Simplified “HMAX”-like:**

- 8,560 weights to learn
- 925,320 shared weights
Caltech 101 subset: 2 categories

- Faces_easy (435 images) – 200 learning / 200 testing

- BACKGROUND_Google (468 images) – 200 learning / 200 testing
20 output neurons

- Fast learning

![Learning (20,000 steps)](image)

- Weights discretization

<table>
<thead>
<tr>
<th>Precision (number of levels)</th>
<th>Ideal</th>
<th>256</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>98.25</td>
<td>99</td>
<td>98</td>
<td>97.75</td>
<td>97.75</td>
<td>98.5</td>
<td>89.75</td>
<td>55.5</td>
</tr>
</tbody>
</table>

- tanh() approximated with simple saturation ➔ identical performances

Final: 100% (fast 10000)

Final: 98.25%
Towards Hardware Synthesis

1) Deep network builder

```
; Environment
[env]
SizeX=8
SizeY=8
ConfigSection=env.config

[env.config]
ImageScale=0

; First layer (convolution)
[conv1]
Input=env
Type=Conv
KernelWidth=3
KernelHeight=3
NbChannels=32
Stride=1

; Second layer (pooling)
[pool1]
Input=conv1
Type=Pool
PoolWidth=2
PoolHeight=2
NbChannels=32
Stride=2
```

2) Defects learning

xnet network.ini database -learn

3) Performances analysis

Estimated defects visualization

4) C Export and RTL synthesis
Towards Fully CNNs

State-of-the-art in image segmentation

- Take arbitrary input size
- Trained end-to-end, pixels-to-pixels
- Eliminate redundant calculations inherent to “patch” segmentation
- Spike-coding compatible!

Towards even more bio-inspired systems!

- Unsupervised online learning (Spike-Timing-Dependent Plasticity)
- Learning directly from bio-inspired sensors (artificial retina, cochlea, ...)

Input activity (128x128)

Output feature map activity (factor 2 subsampling = 57x57)
Conclusion

Deep Neural Networks are...

- ... at the edge of today’s recognition systems
- ... deployed in large-scale commercial products (Facebook, Google, ...)
- ... hard to integrate into embedded products, even with ASICs

Spiking NVM-based deep networks are promising:

- Computing capabilities identical to conventional networks
- Provide the high memory density required
- True computing in memory, eliminate the memory bottleneck
- Simple and efficient performance tunability capabilities
- Direct interface to bio-inspired sensors (retina, cochlea...)
- Large potential for advanced bio-inspired learning systems
Thank you!

Questions?
Unsupervised Features Extraction

Learning rule

Network topology

Input stimuli

Neuron model

Leaky Integrate & Fire:

\[ u(t) = u_0 \cdot e^{-\frac{t_{\text{spike}} - t_{\text{last spike}}}{\tau_{\text{leak}}} + w } \]

Synaptic model

Neurons activity

Synaptic weights

Neuron membrane potential

O. Bichler et al. “Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity.” Neural Networks, 2012