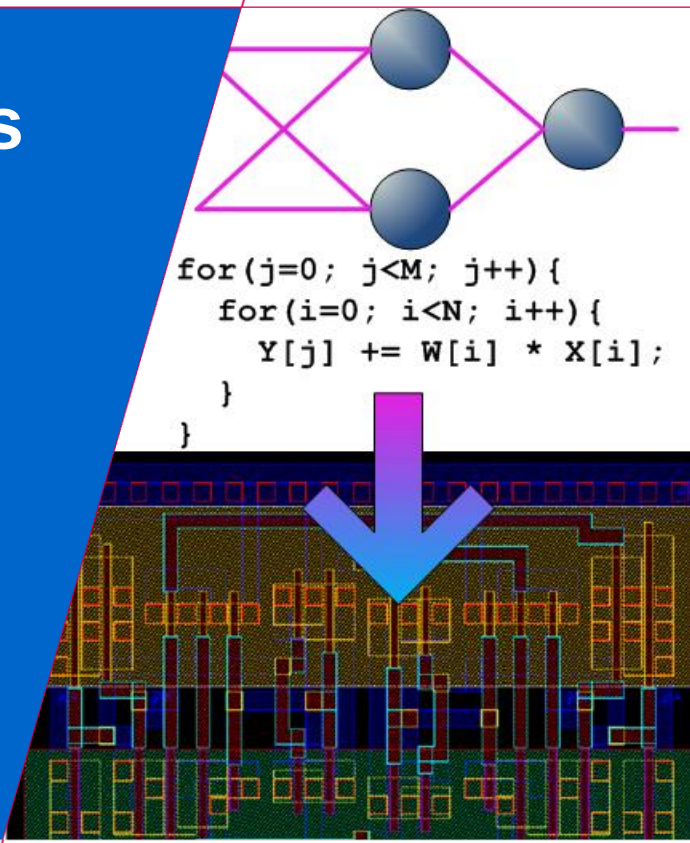
Electronic Systems

Neural Computer Architectures

Accelerating Deep Learning Applications



By: Maurice Peemen

Date: 31-5-2017

TU/e Technische Universiteit Eindhoven University of Technology

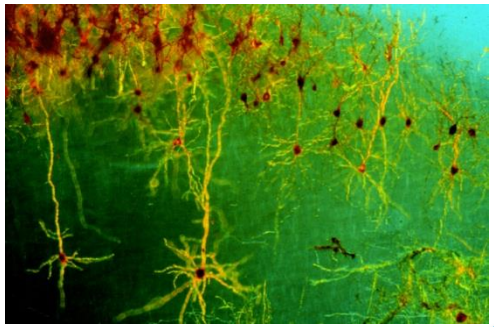
Where innovation starts

Background Maurice

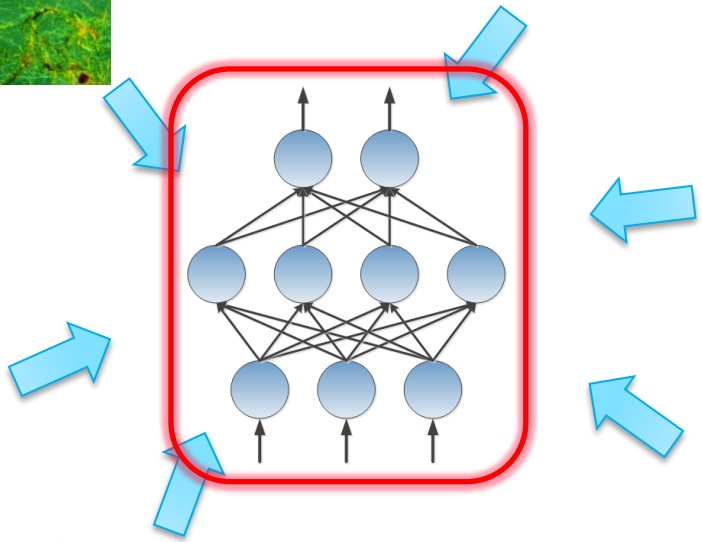
- Masters at TU/e
- PhD work at TU/e
- Thesis work with Henk Corporaal
  - Improving the Efficiency of Deep Convolutional Networks
  - Defense after summer
- Research Scientist at FEI Company
  - Electron Microscopy Imaging Challenges

Convergence of different domains

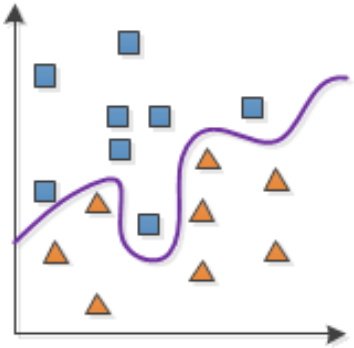
Neurobiology

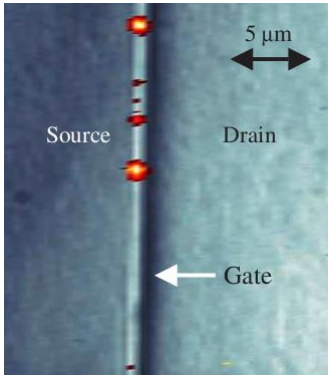


Applications



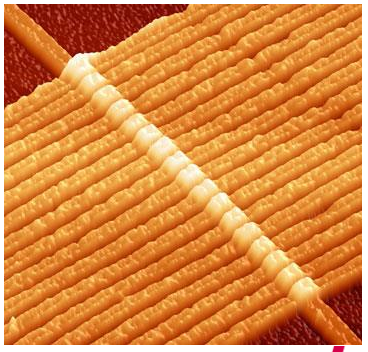
Machine Learning



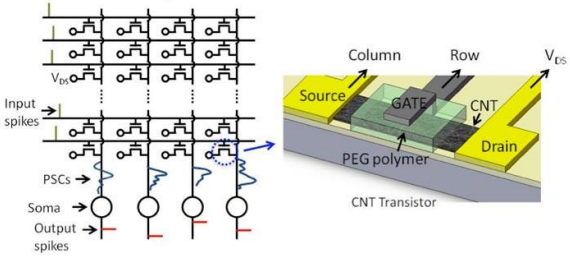


Constraints

Technology

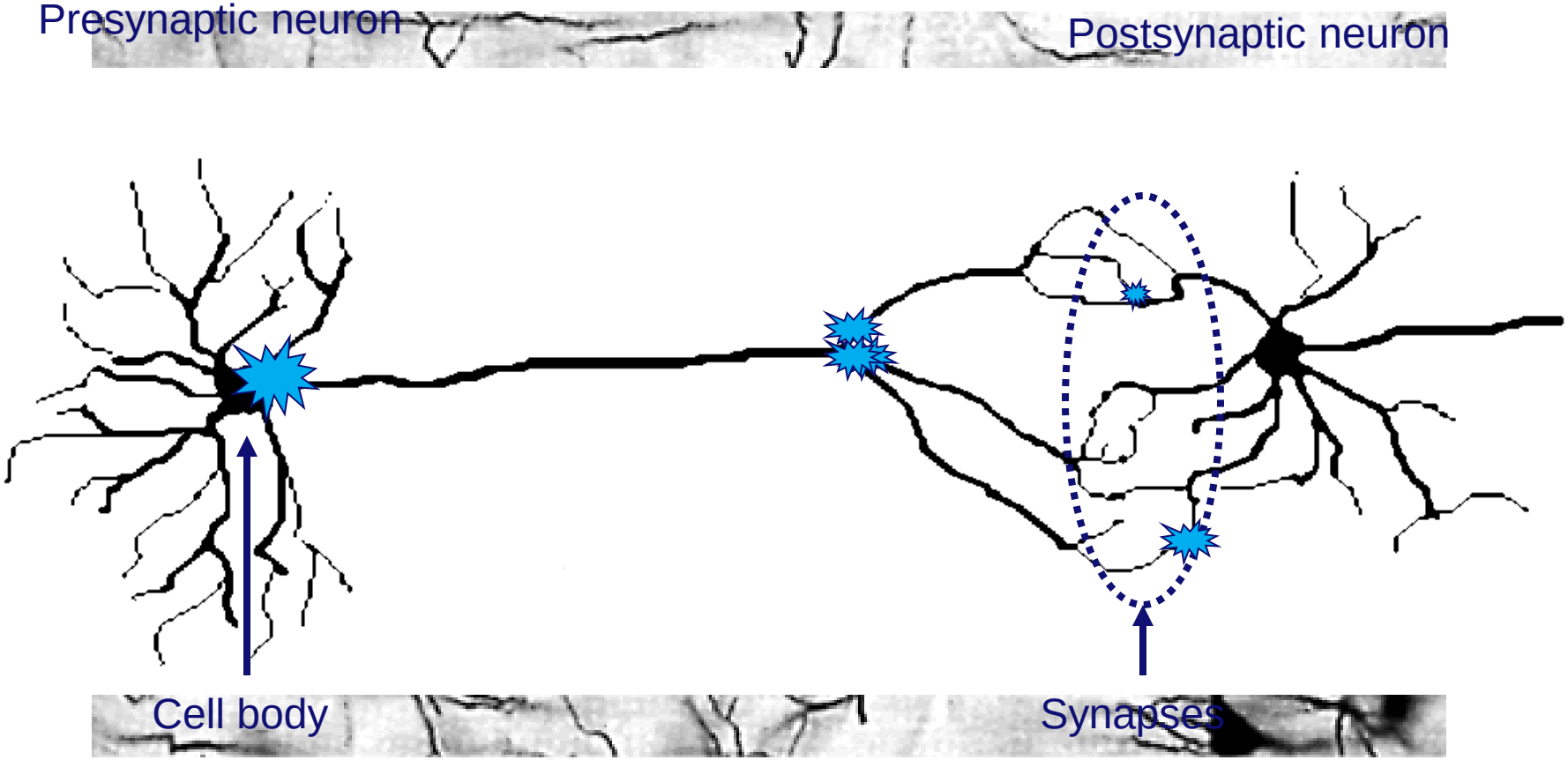


Innovations



Neuromorphic

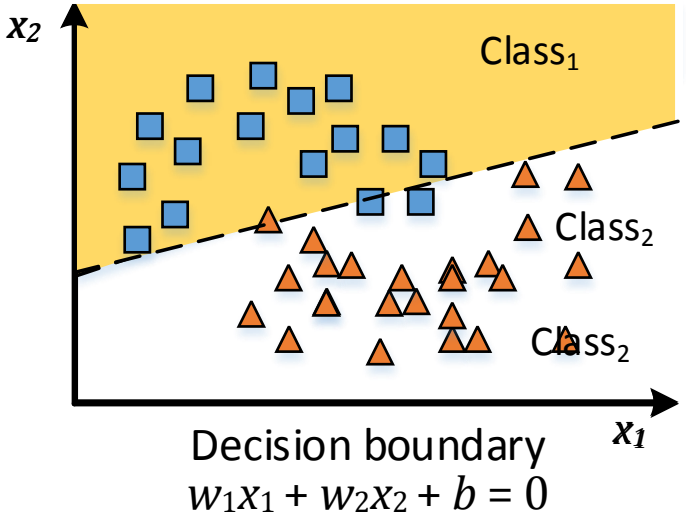
Biological Neural Networks

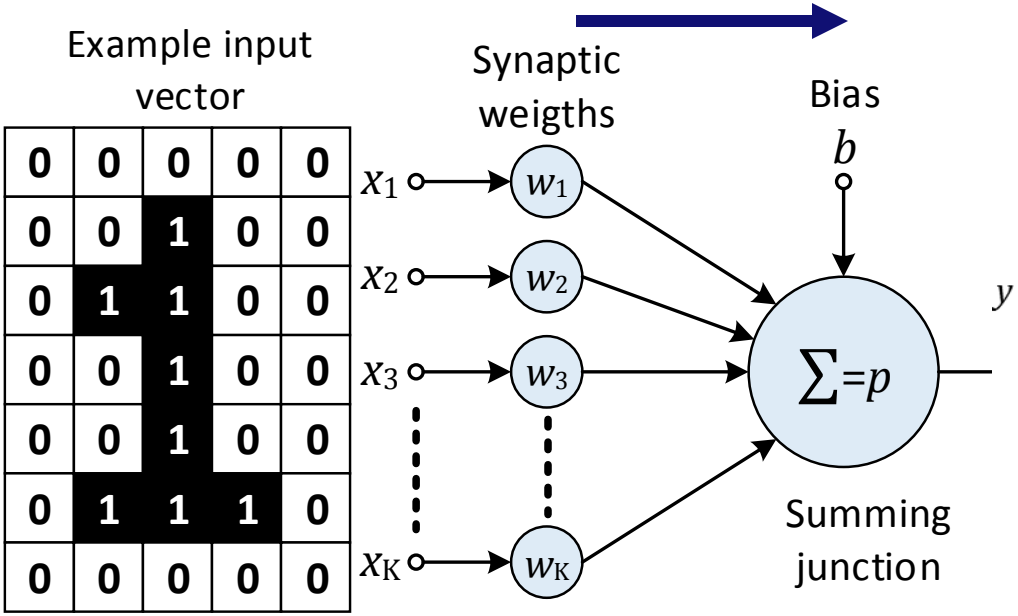


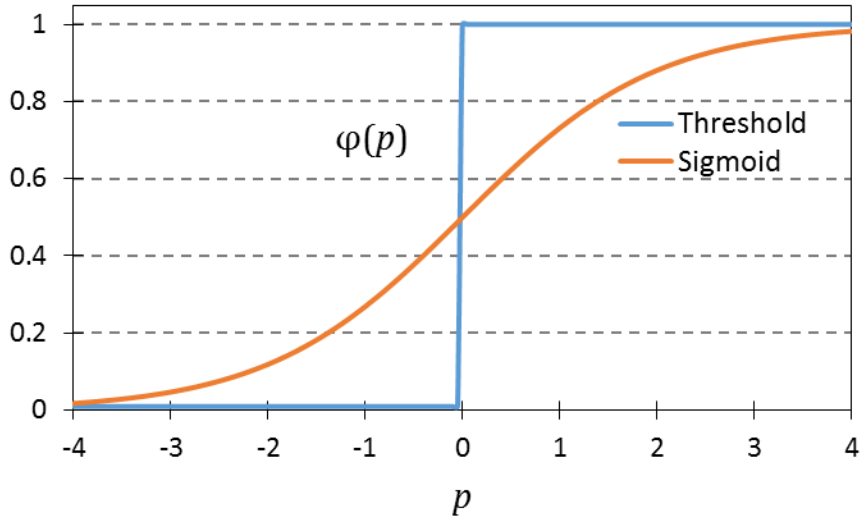
Perceptron Model (1957)

- **Feed forward processing**
- **Tuning the weights by learning**
- **Non-linear separability (1969)**

$$y = \varphi\left(b + \sum_i x_i \cdot w_i \right)$$

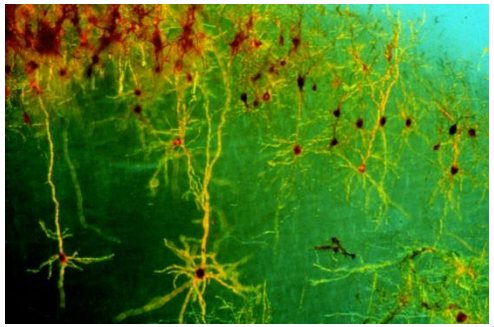




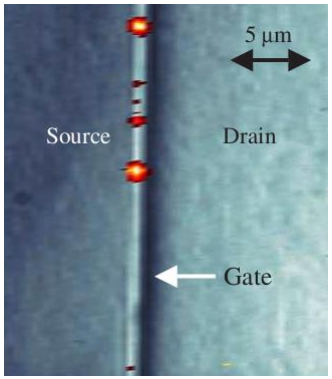


Convergence of different domains

Neurobiology

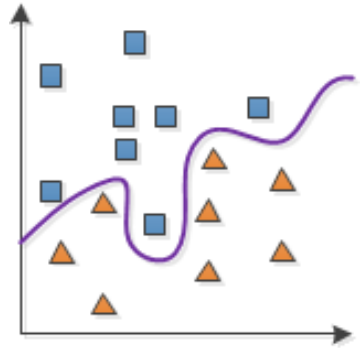


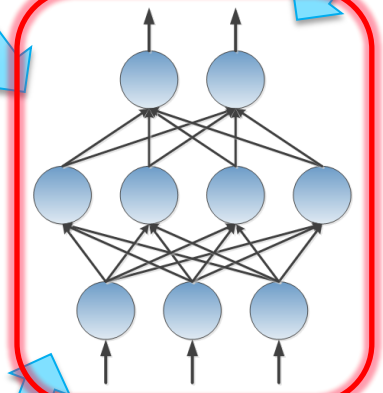
Applications



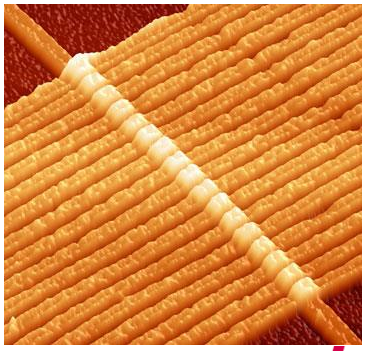
Constraints

Machine Learning

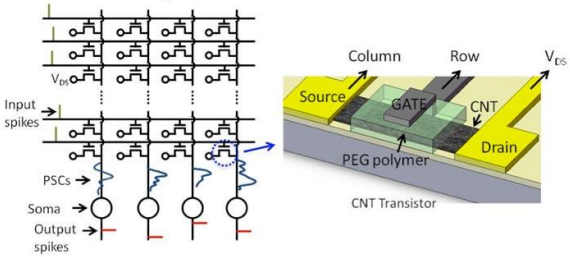




Technology



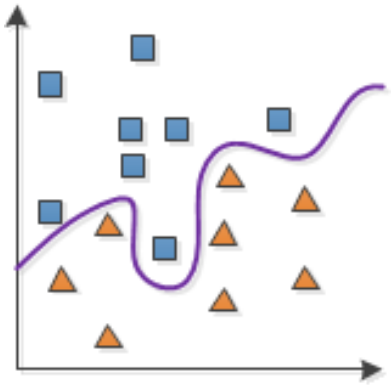
Innovations



Neuromorphic

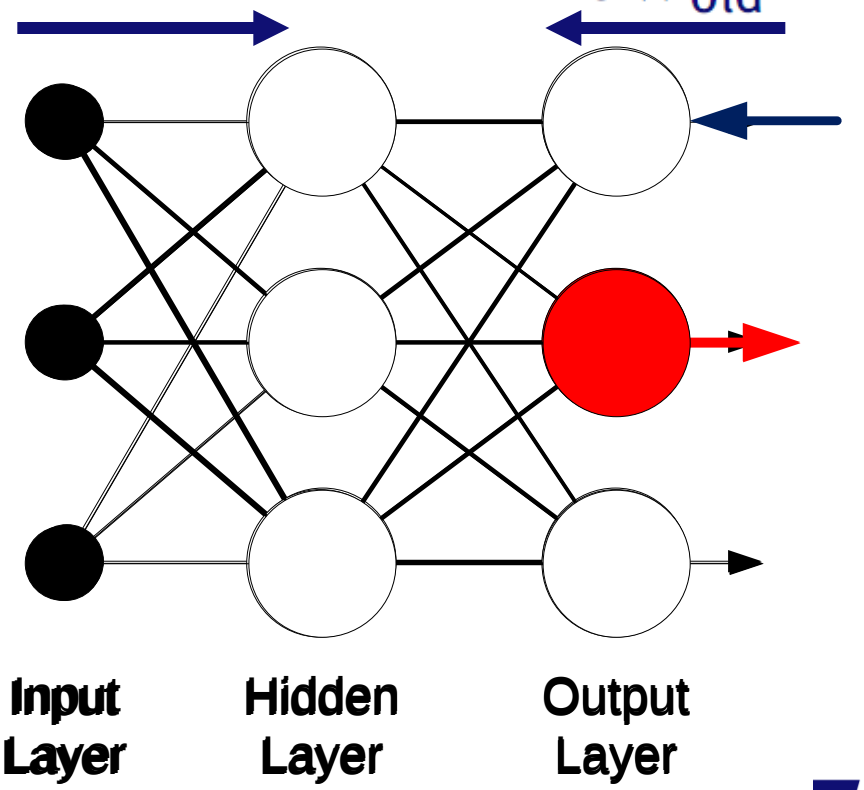
Multi Layer Perceptron (1979)

- **Training is done by error back-propagation**



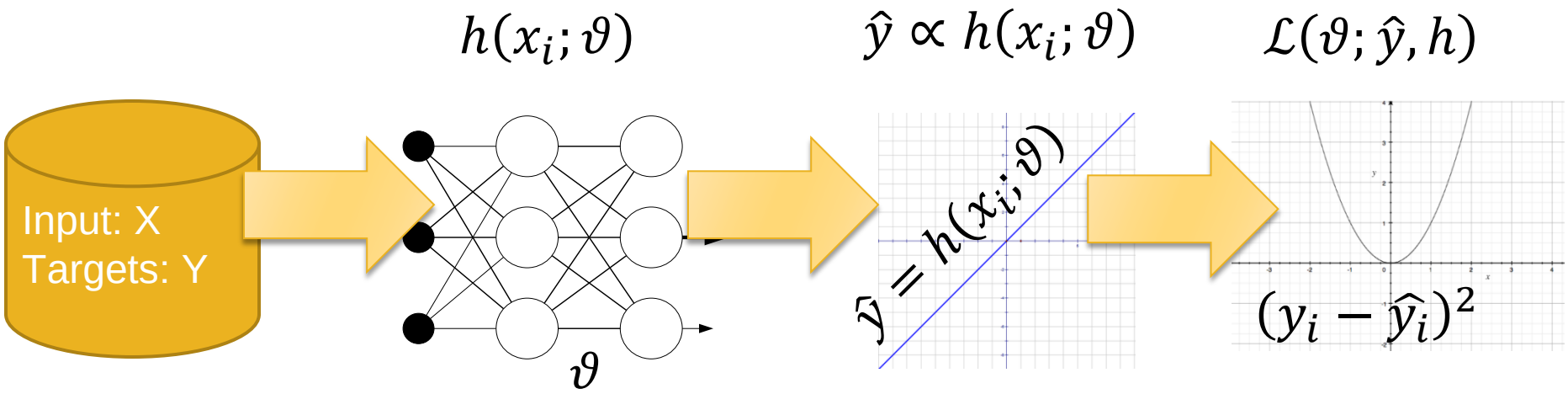
0	0	0	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	0	1	0	0
0	1	1	1	0
0	0	0	0	0

$$W_{new} = W_{old} - \eta \frac{\partial E_{CE}}{\partial W_{old}}$$



Optimization Through Gradient Descent (1)

- Collect annotated data
- Define model and initialize randomly
- Predict based on current model
  - In neural network jargon “forward propagation”
- Evaluate predictions



Optimization Through Gradient Descent (2)

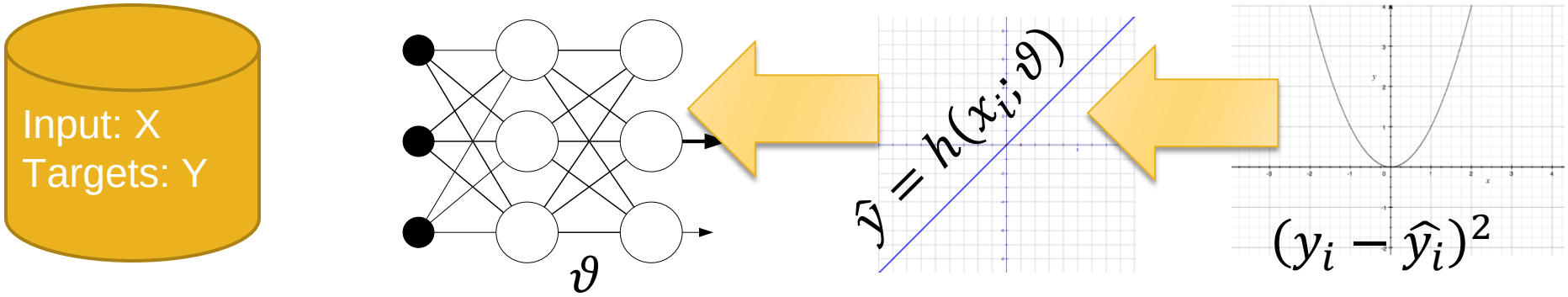
- **Use the derivative of the error to optimize:**

$$\theta^{t+1} = \theta^t - \eta_t \nabla_\theta \mathcal{L}$$

- **The most important component is the gradient**
- **Then update the parameters in the negative direction of the gradient**

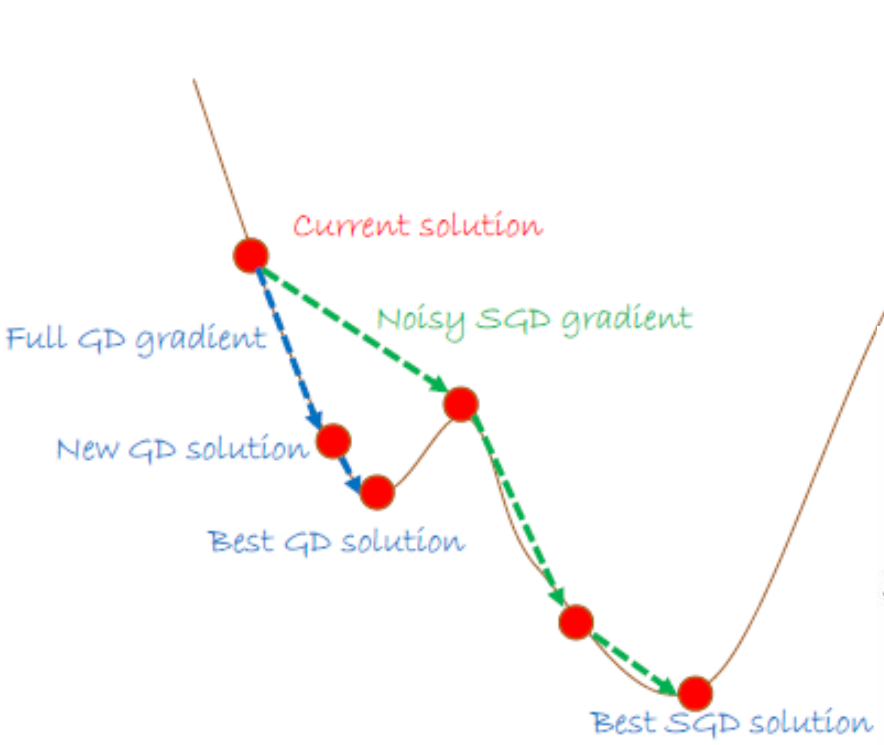
$$\frac{\partial h(x_i)}{\partial \theta} \qquad \frac{\partial \hat{y}}{\partial h} \qquad \frac{\partial \mathcal{L}(\vartheta; \hat{y}, h)}{\partial \hat{y}}$$

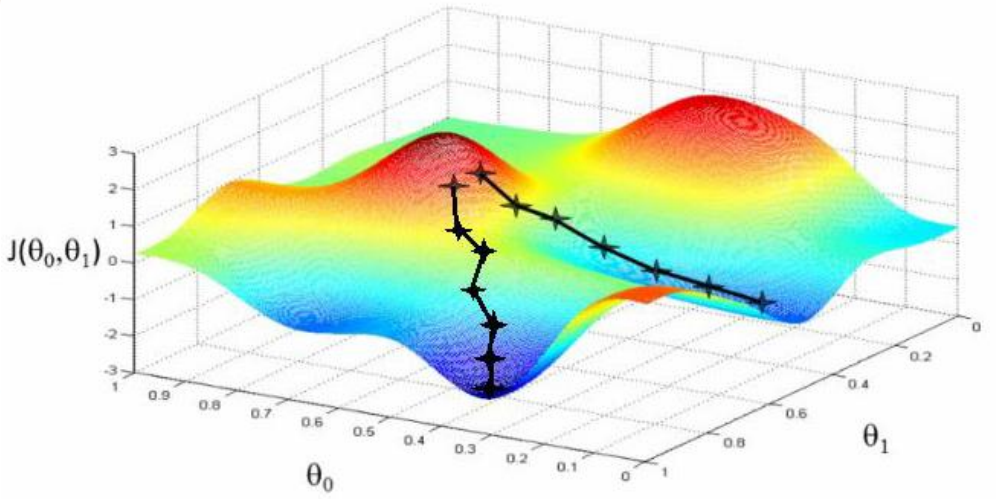
$$h(x_i; \vartheta) \qquad \hat{y} \propto h(x_i; \vartheta) \qquad \mathcal{L}(\vartheta; \hat{y}, h)$$



Optimization Through Gradient Descent (3)

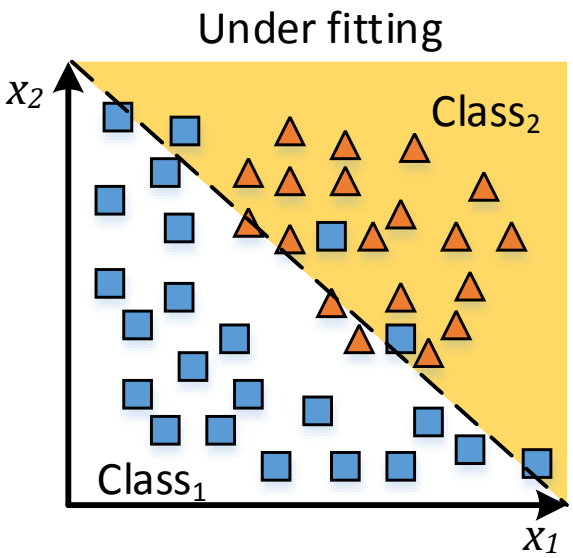
- Follow the path towards local minima in the parameter space
- Stochastic Gradient Descent
  - Every random sample update the parameter space

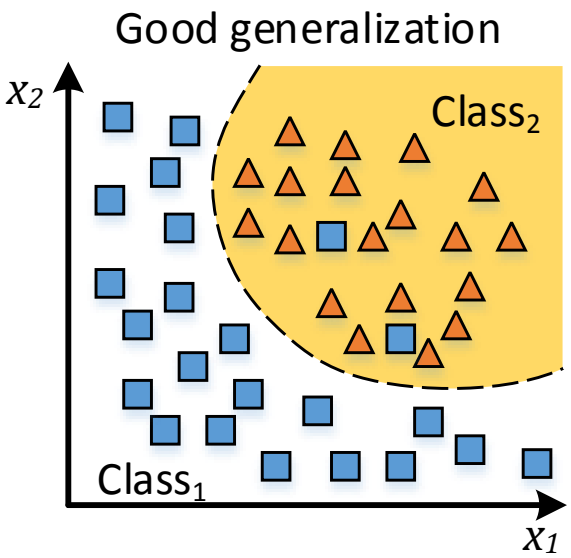


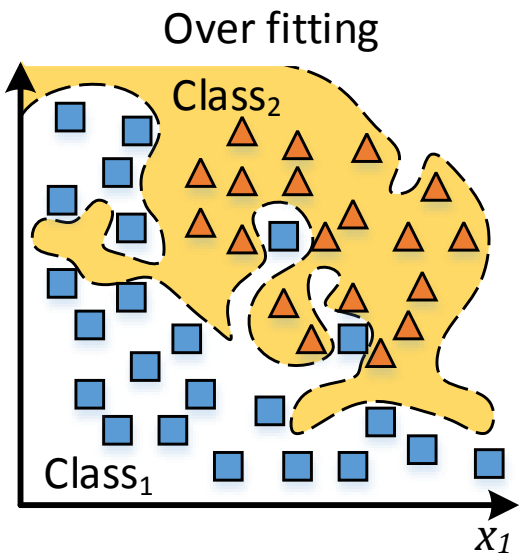


Generalization

- Take an abstract representation
- Add details
- Add to many details
- A common problem in Machine Learning problems

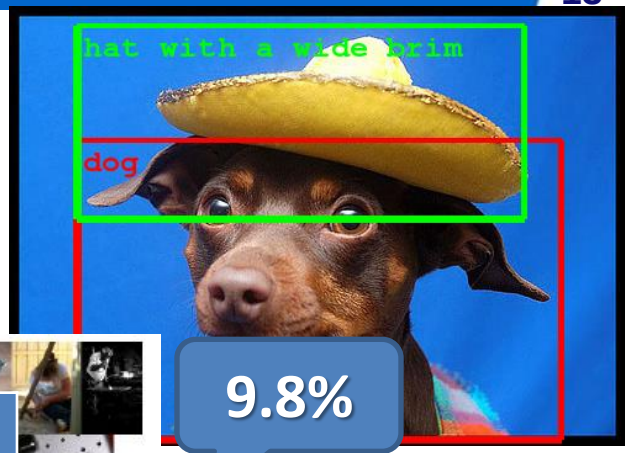






Data is Everywhere

Large Scale Visual Recognition Challenge



- IMAGENET 2012 Classification
  - Many training samples: 1,281,167
  - Large number of classes: 1000

Submission	Method	Error Rate
SuperVision	Convolutional Net	0.164
ISI	Other stuff… (SIFT, SVMs)	0.262
XRCE/INRIA		0.271
OXFORD_VGG		0.273

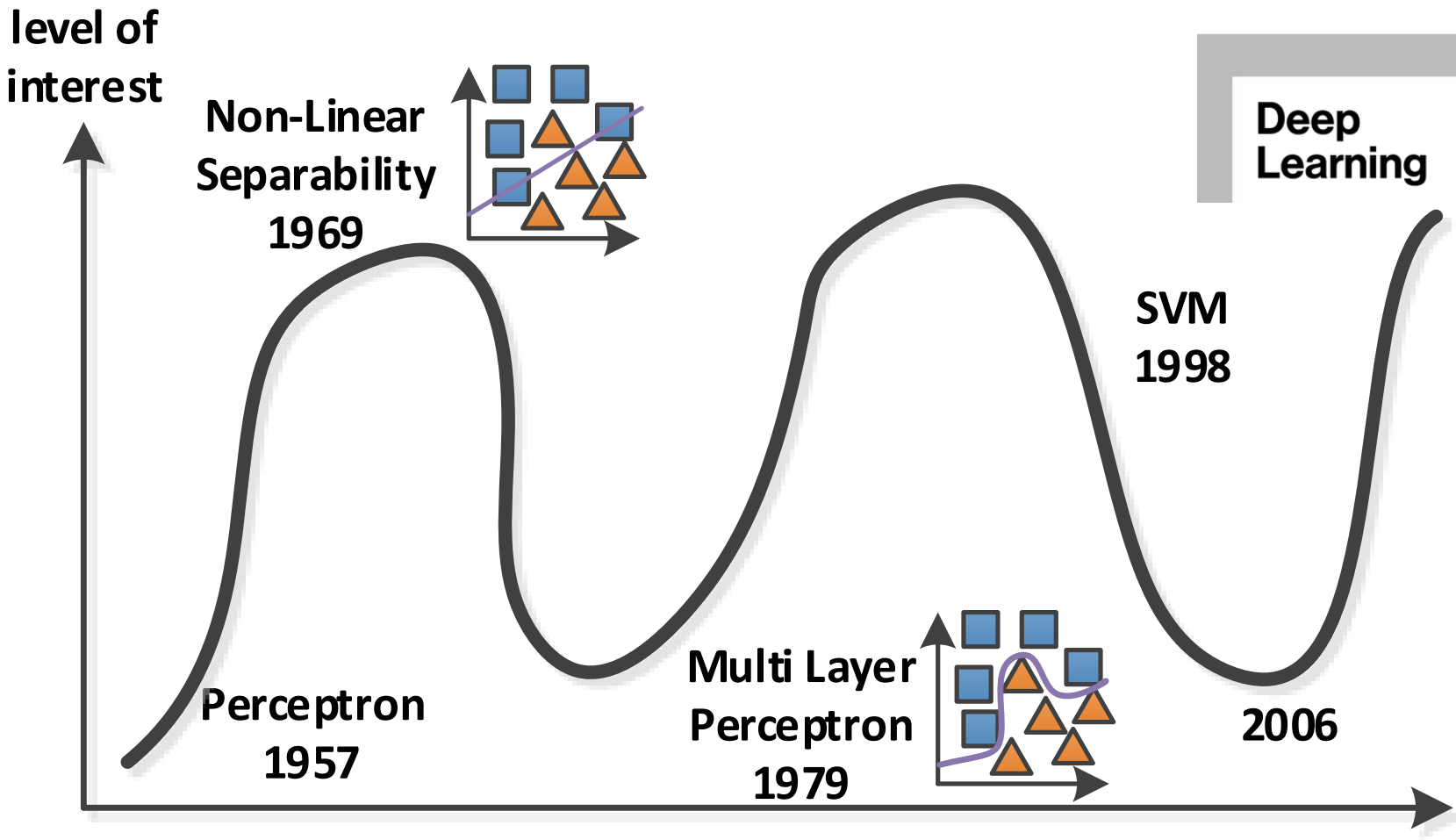
9.8%

1.2%

- ImageNet 2013
  - Complete top 10 used Deep Nets

The Hype Curve of Neural Networks

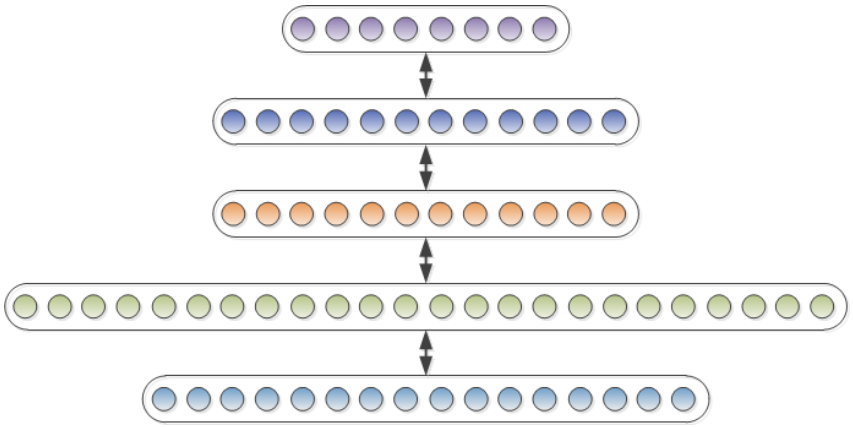
MIT Technology Review 10 BREAKTHROUGH TECHNOLOGIES 2013



Deep Big Neural Networks

- Deep Big Neural networks outperform SVM
- Complex models with over 10 billion parameters
- Unreasonably effective at classification tasks

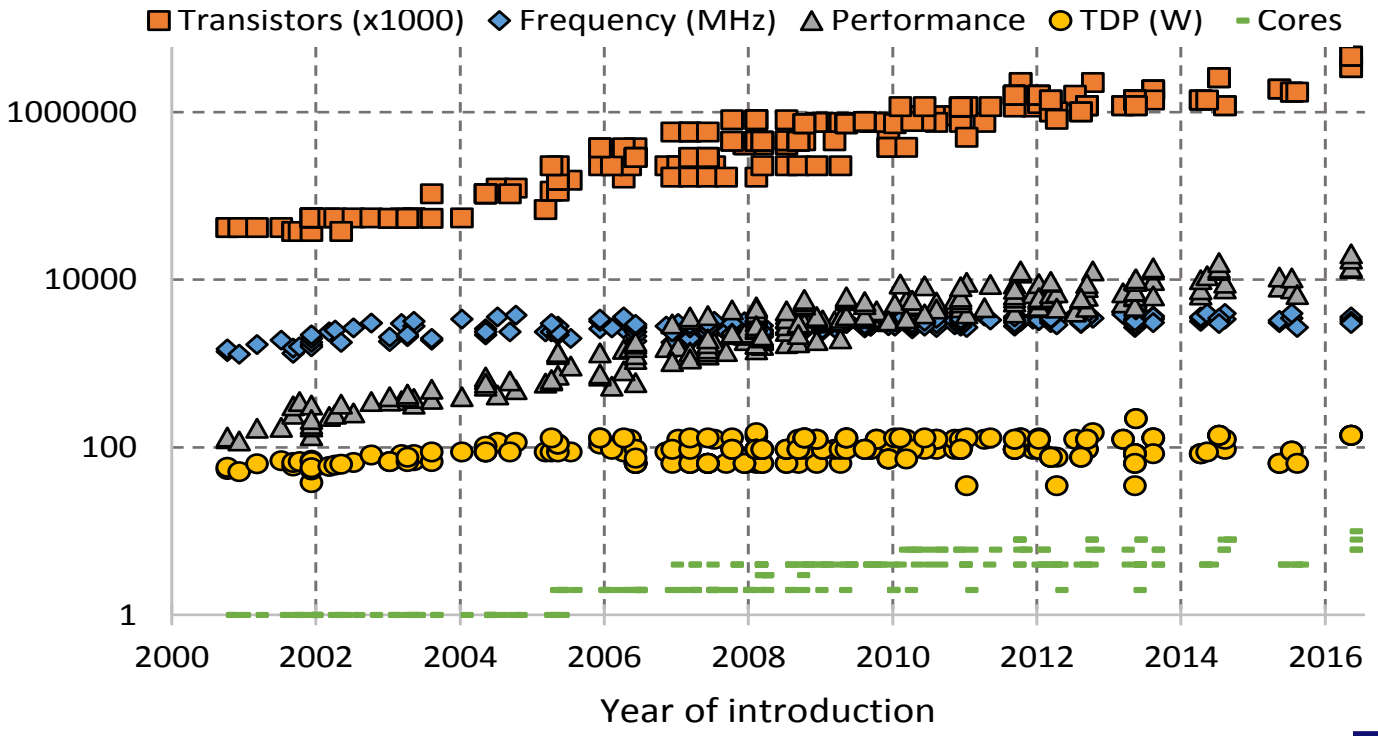
- ≥ 5 layers
- 1000s of nodes
- connection constraints



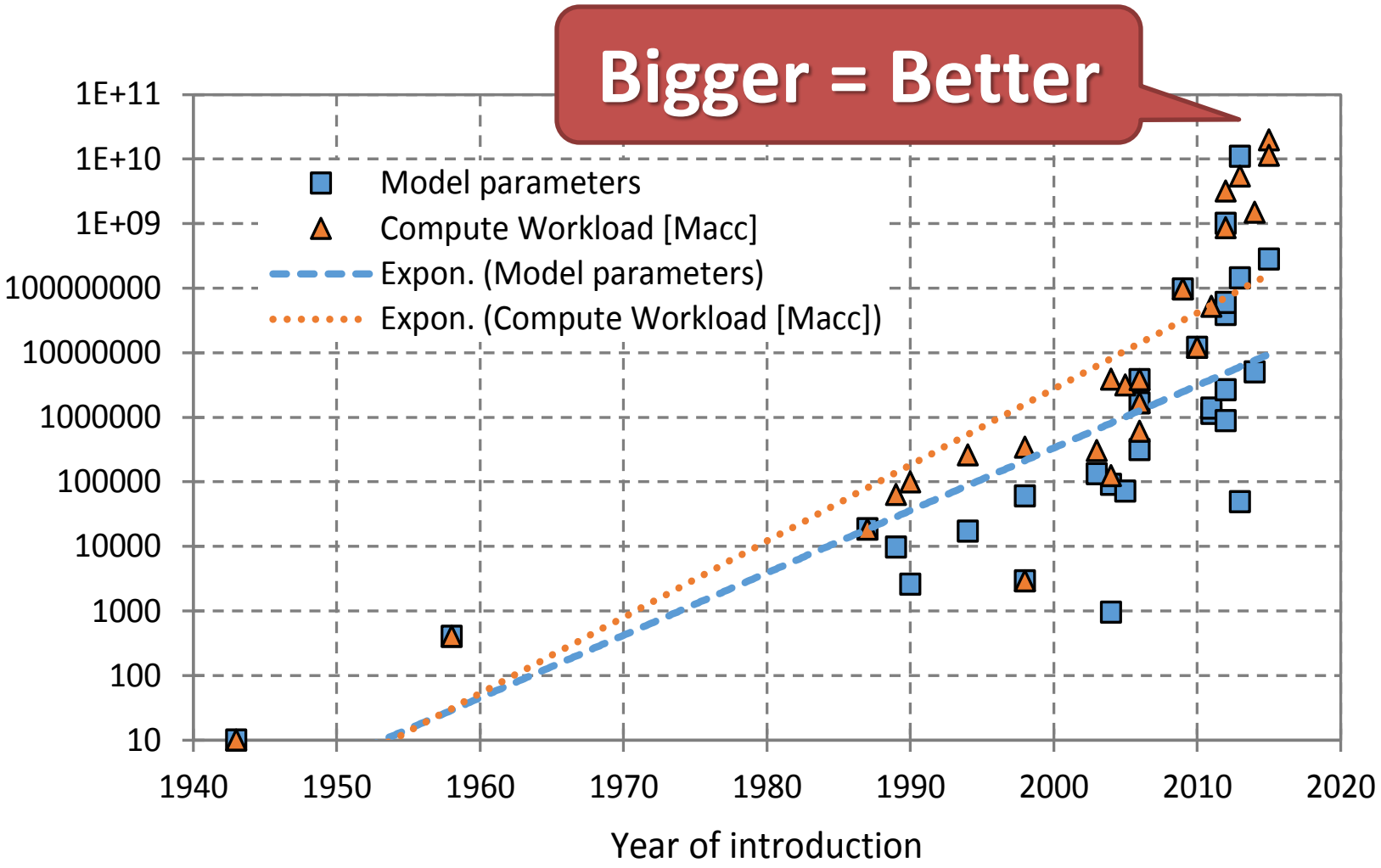
Big Deep Network

The unreasonable effectiveness of deep networks

- **Big data everywhere**
  - **Google, Facebook, Amazon**
  - **NSA, Government, Banks**
- **Moore's law**

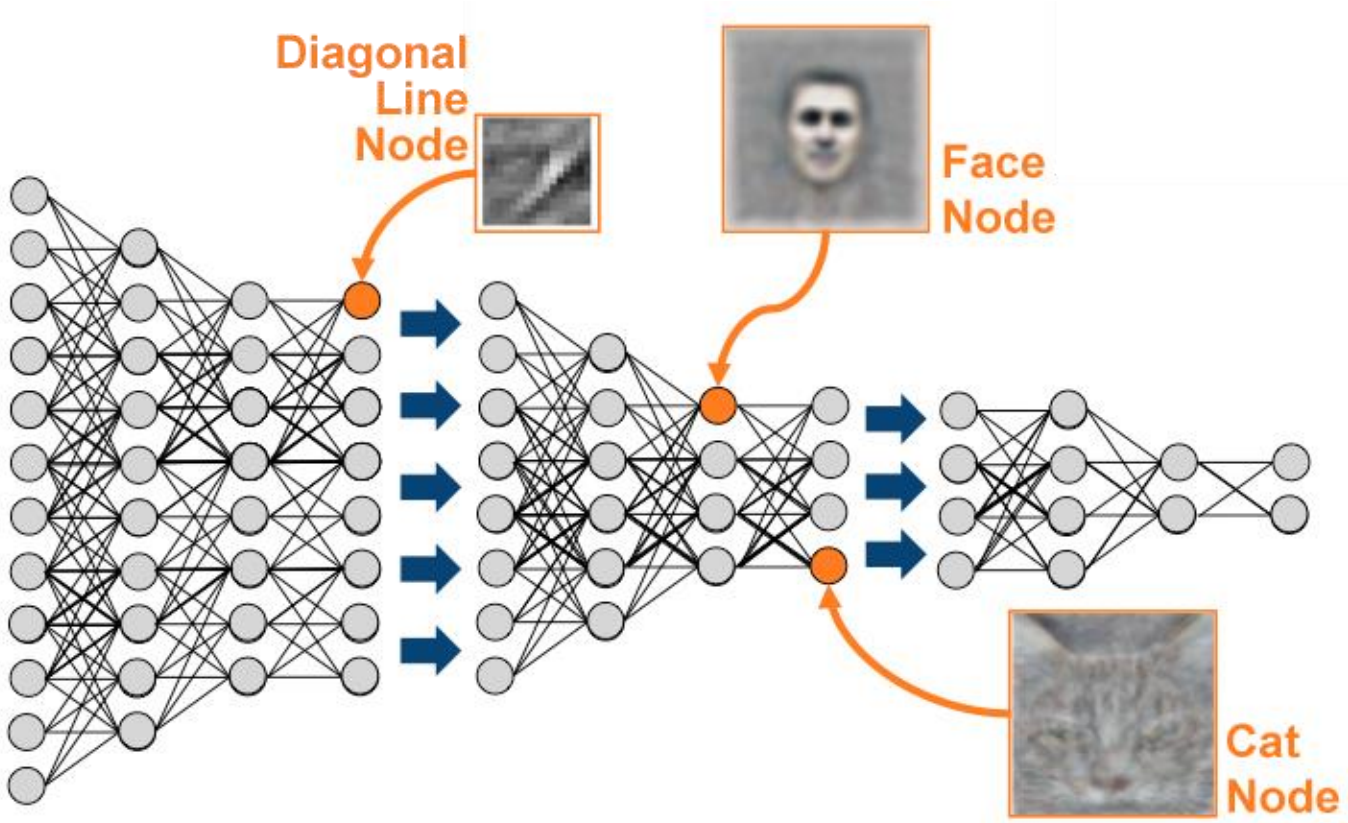


Trends in Deep Learning

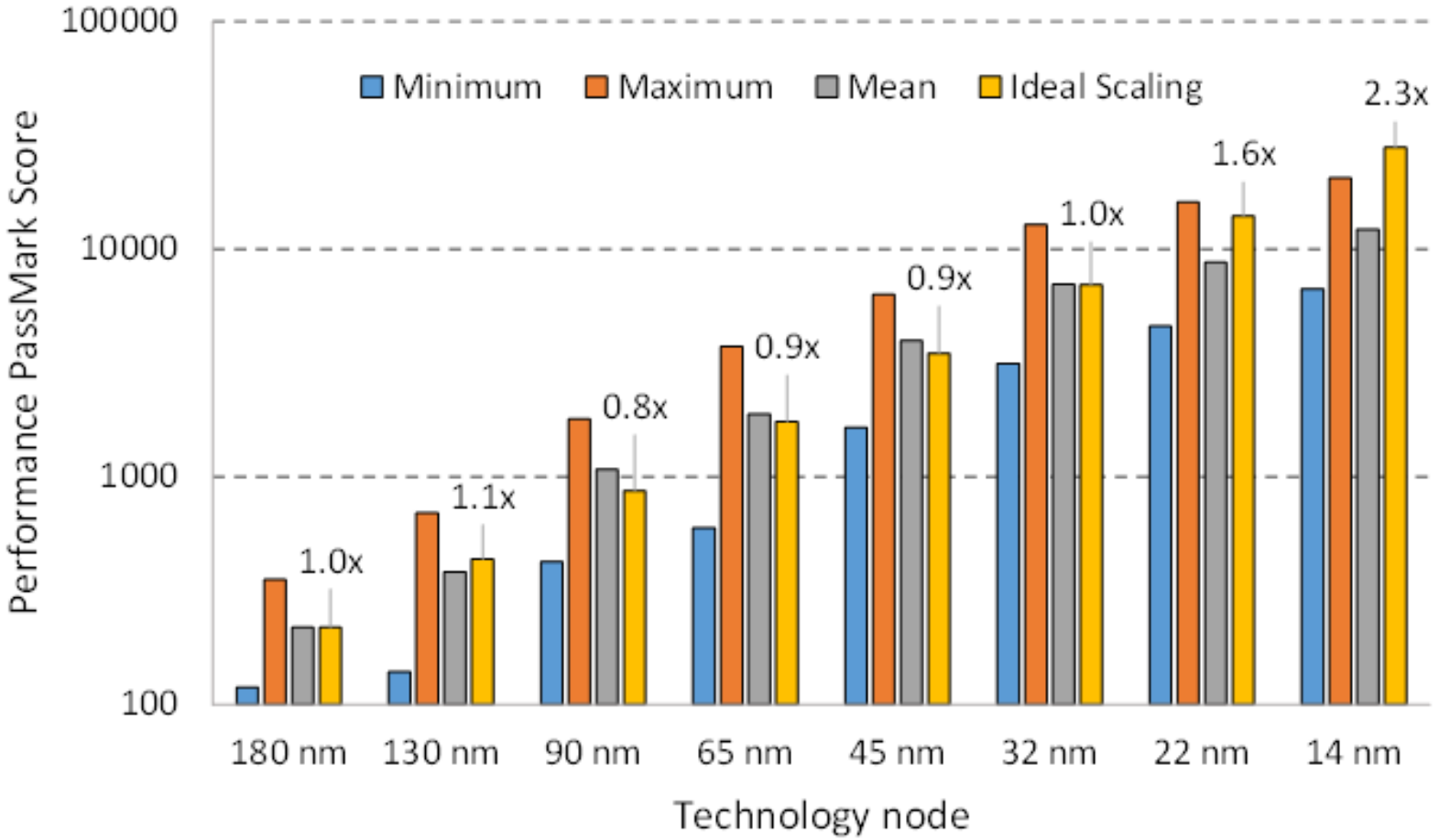


Example of a large network trained by Google

- **10 Billion free parameters**
- **Trained on many millions of YouTube videos**

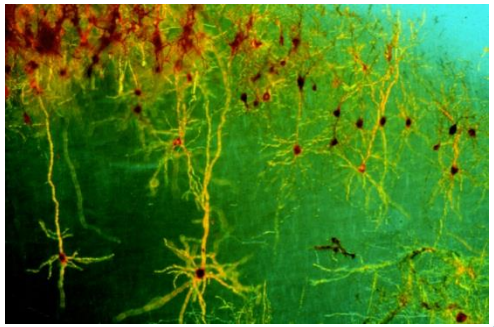


Performance scaling is slowing down

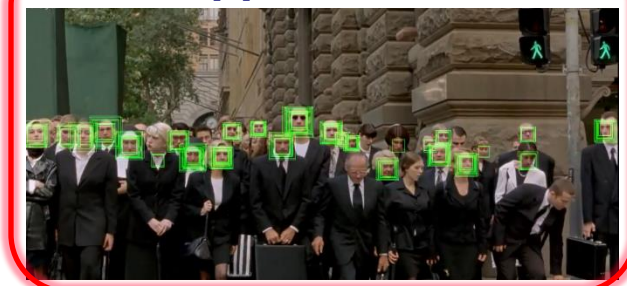


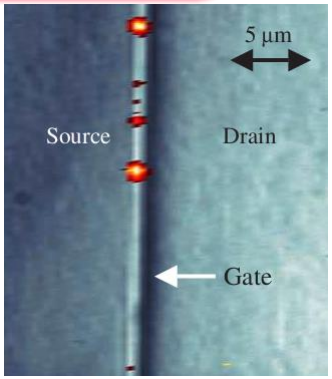
Convergence of different domains

Neurobiology



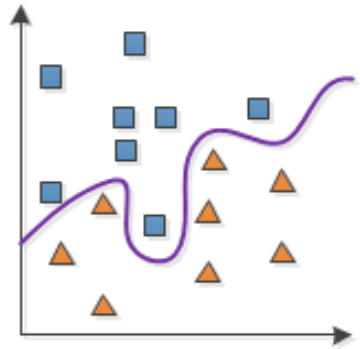
Applications

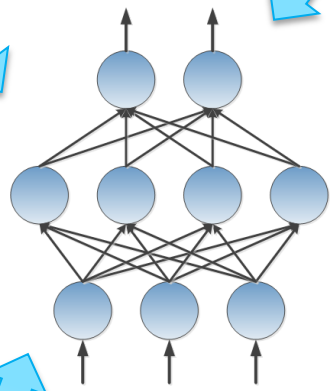


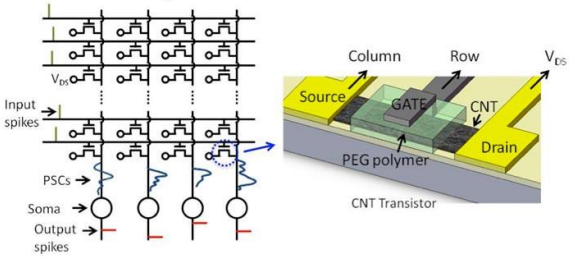


Constraints

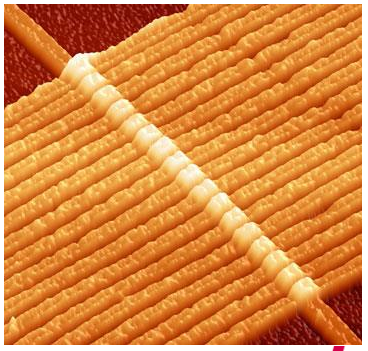
Machine Learning







Technology



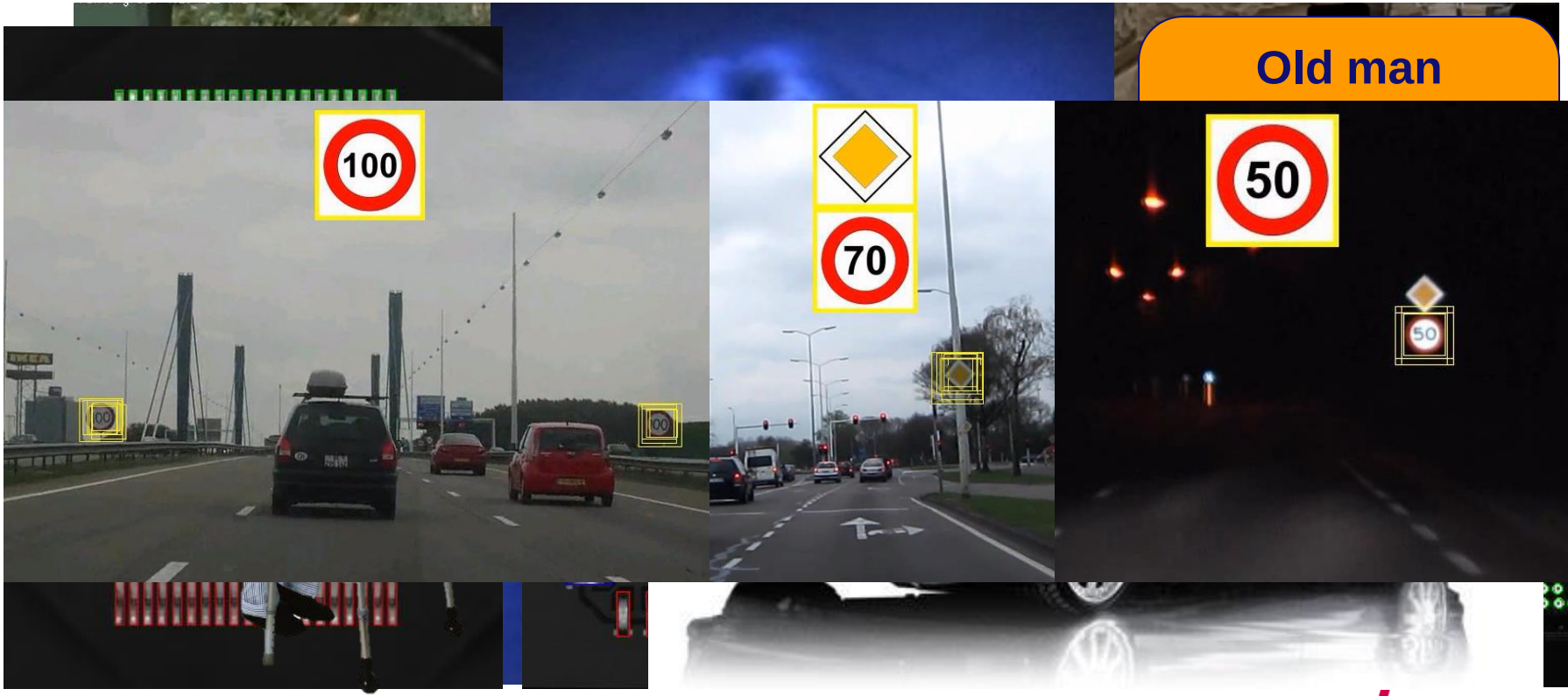
Innovations

Neuromorphic

Classification: Face detection

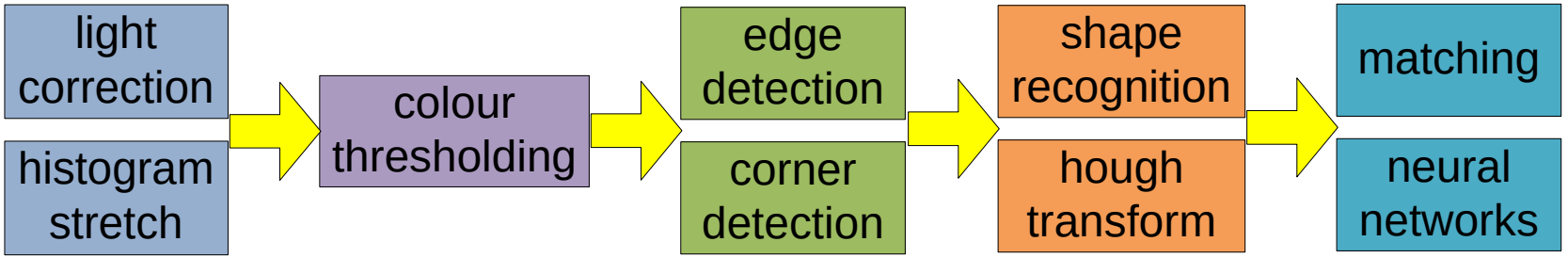
Intelligent Vision Applications

- Emerging field of research
- Applications in many domains
- Examples: Security, Industrial, Medical, Automotive



Classical recognition systems are stupid

- Design is based on knowledge of the task
- Carefully tuned pipeline of algorithms
- Really complex for real world problems
- Design must be redone if the task changes



Train a Neural Network for the task

- **Focus on data instead of algorithm complexity**
-
-

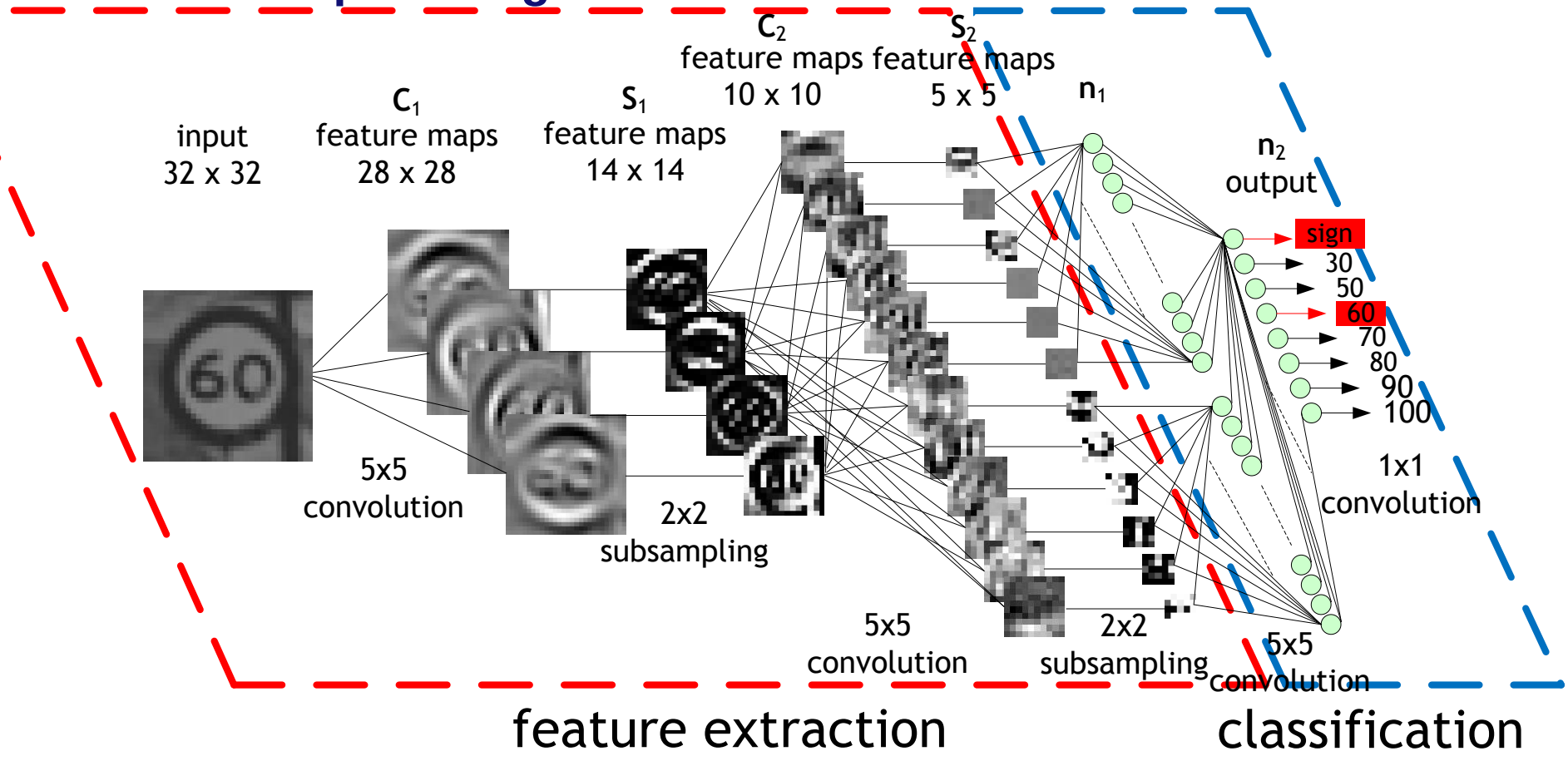
30 km/h | 50 km/h | 60 km/h | 70 km/h | 80 km/h | 90 km/h | 100 km/h

Background images hard to suppress

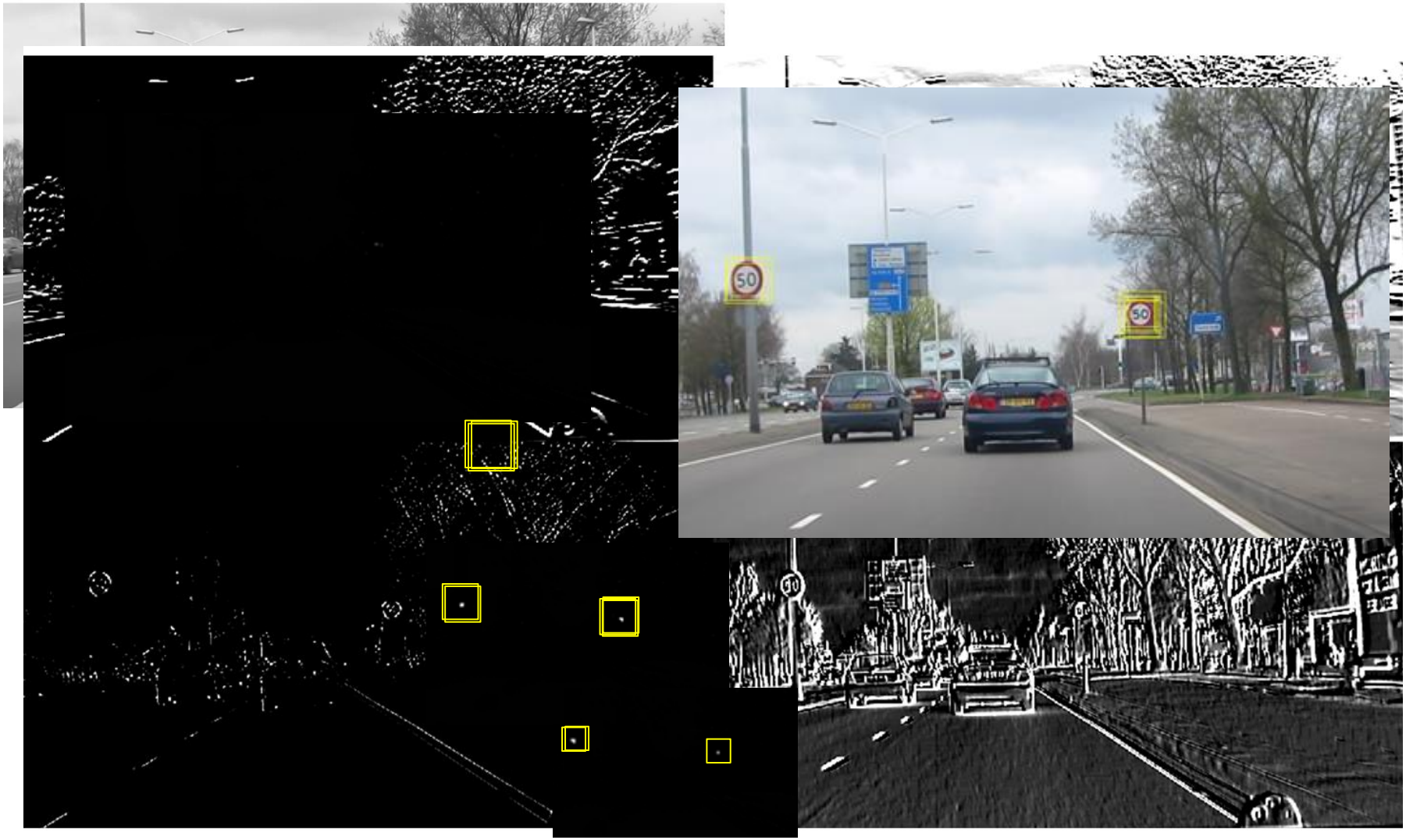
Random background image patches

Biologically inspired object recognition

- **Convolutional Neural Network**
  - **A deep and big neural network**

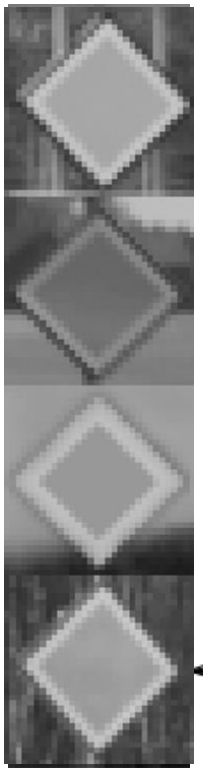


Detection and Recognition Application

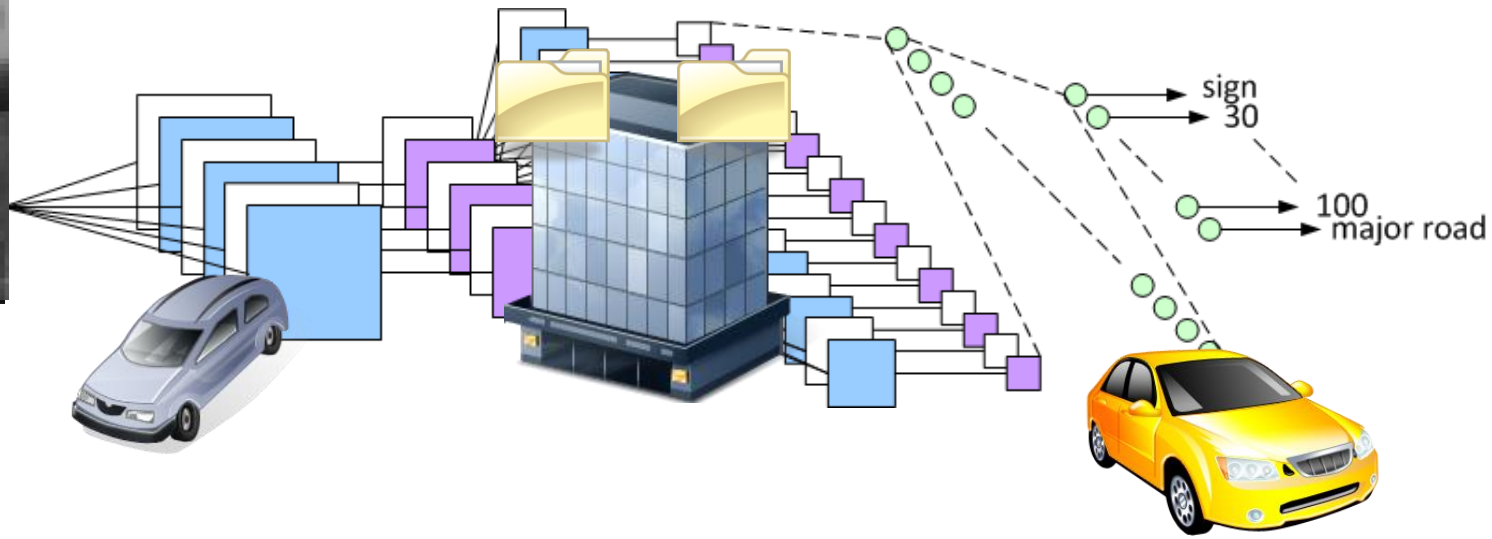


Speed Sign Detection and Recognition

Advantage of flexibility

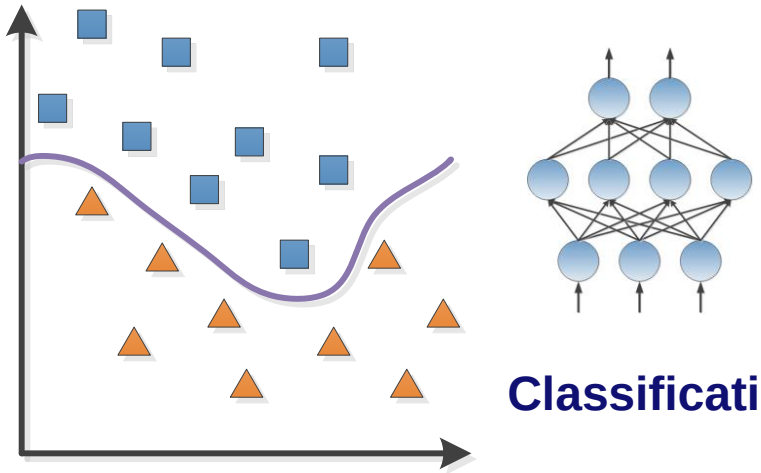


- Extend existing trained network
- Add new road signs and restart training
- New weight file is new functionality
- Send new weight file to users (100 KB)

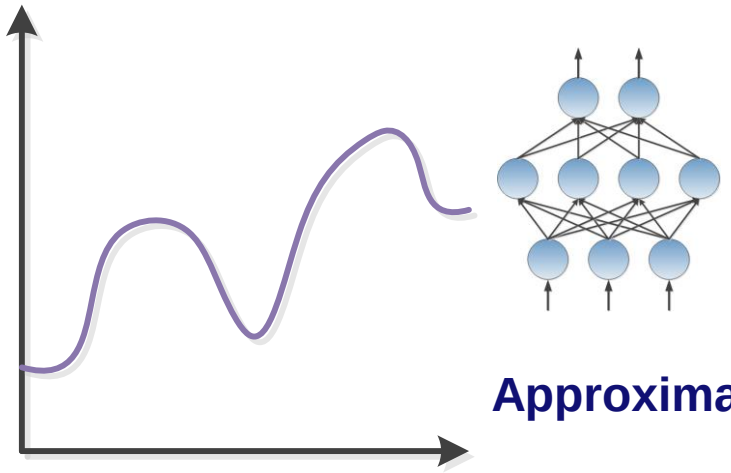


Major road detection

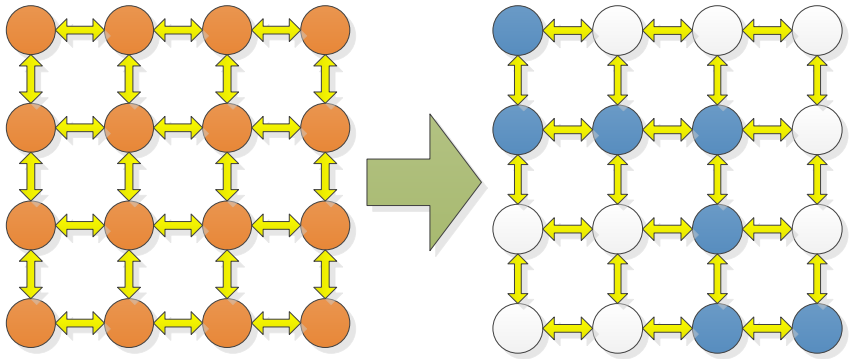
What can these NN further do



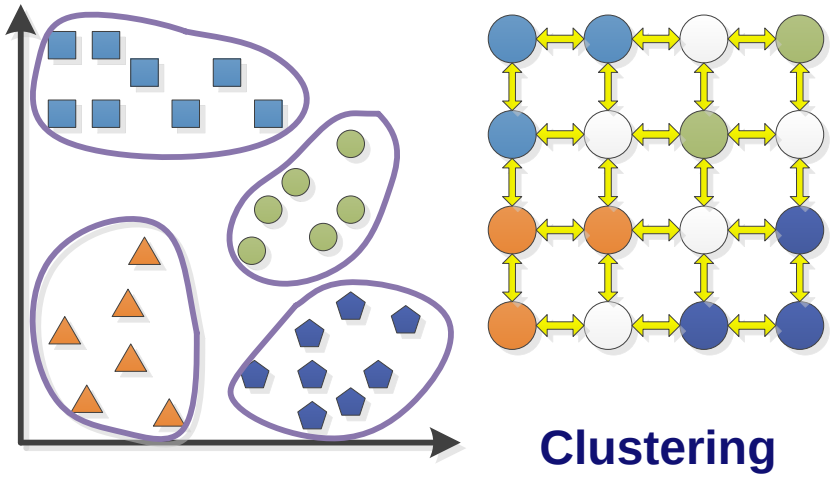
Classification



Approximation



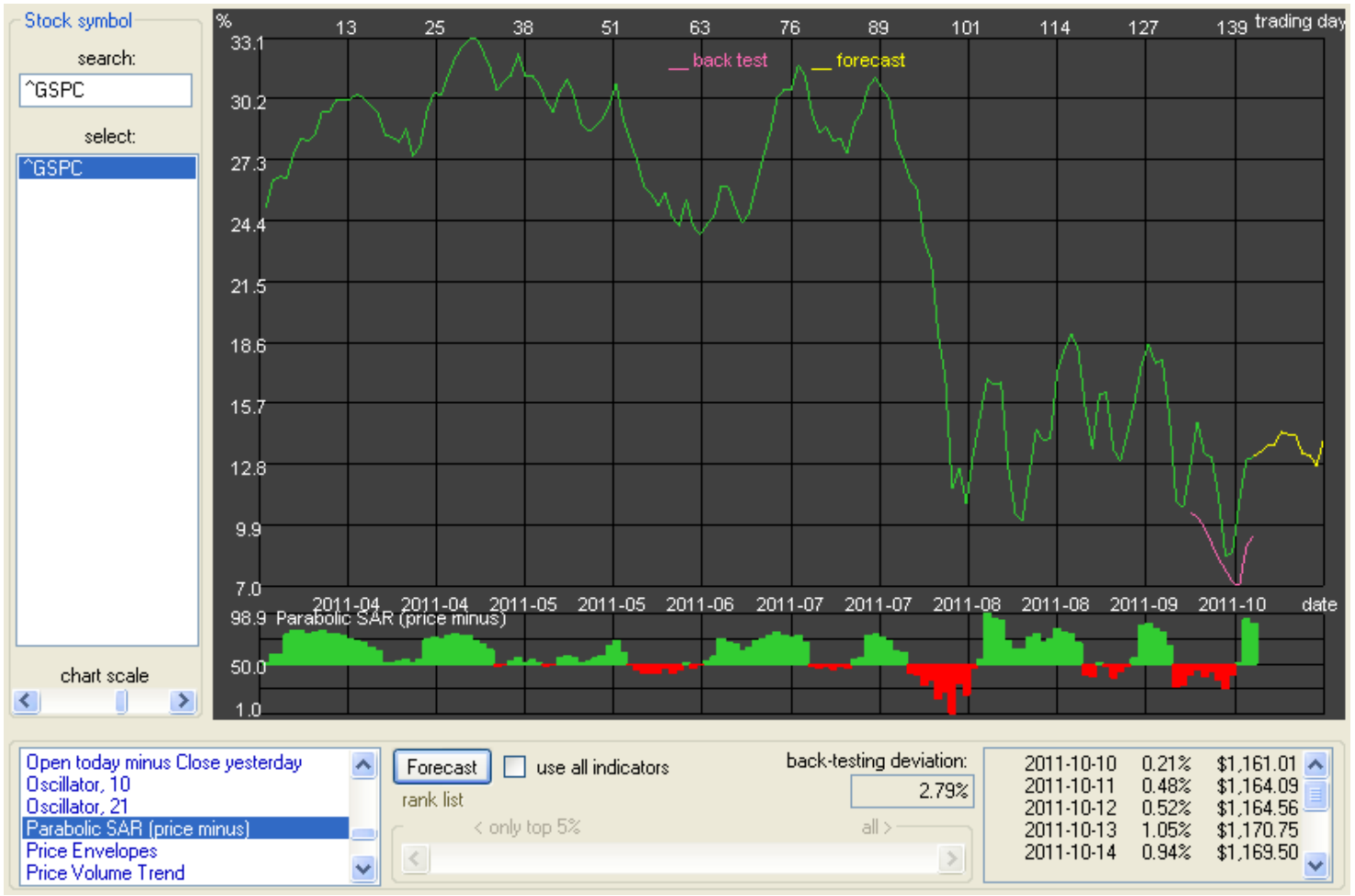
Optimization



Clustering

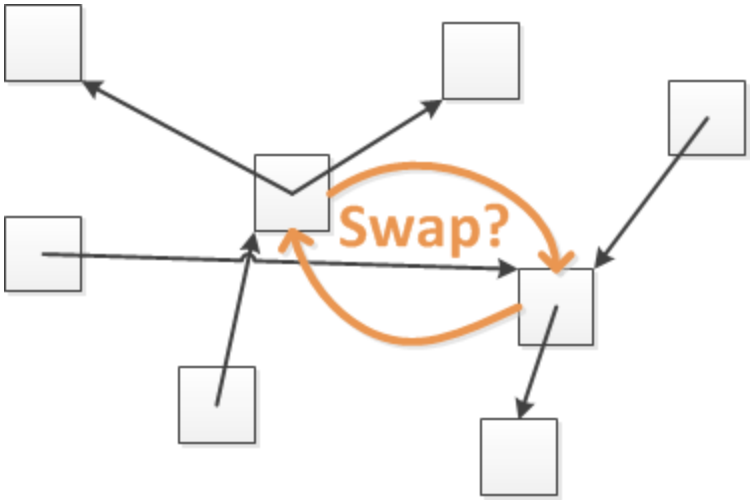
Function Approximation

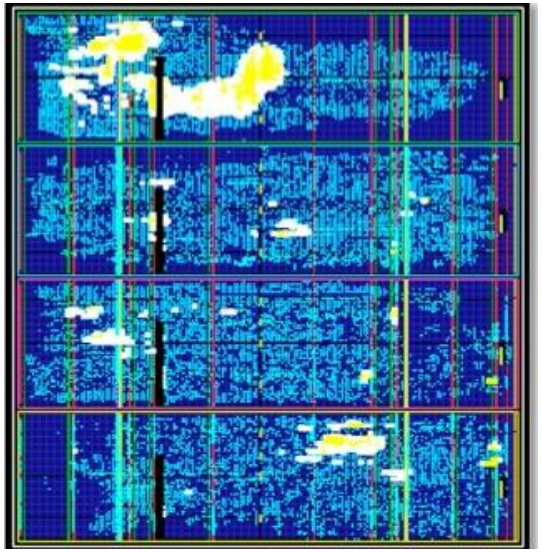
- **Stock market prediction: Black Scholes**

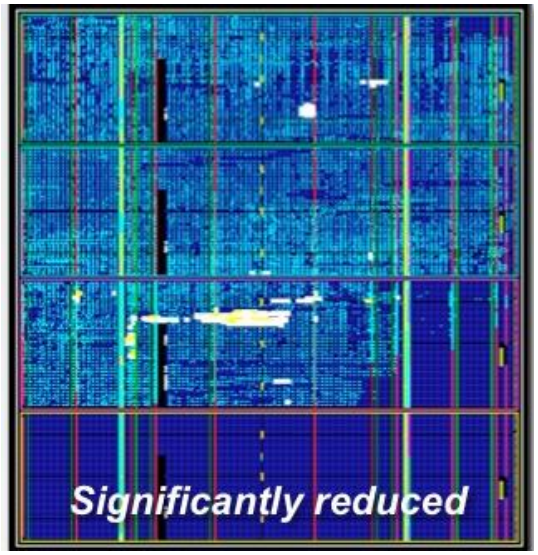


Placement Optimization

- Chip routing: Canneal
- Minimize wire length
- Hopfield Neural Network

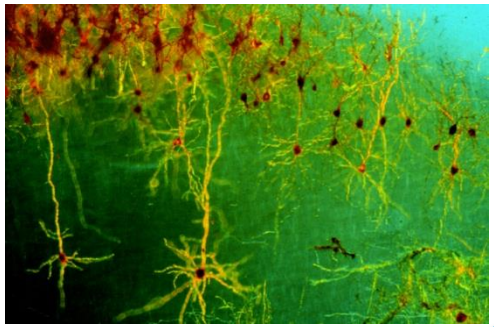




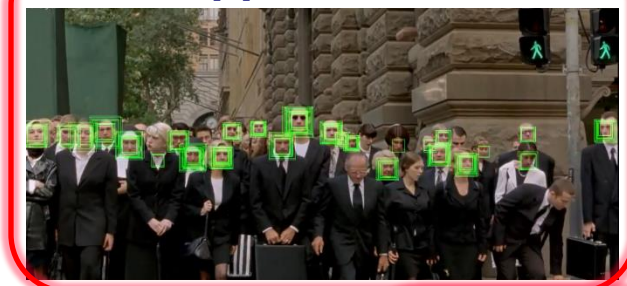


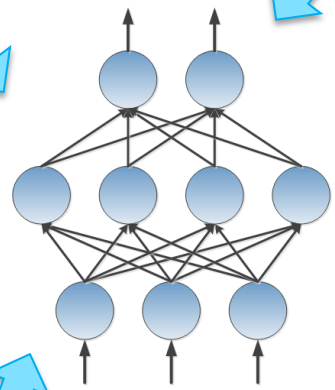
Convergence of different domains

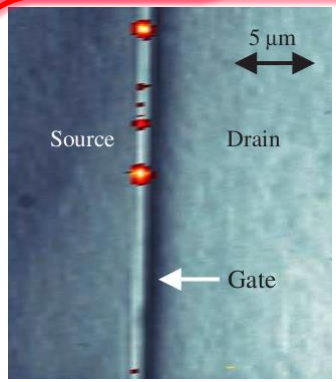
Neurobiology



Applications



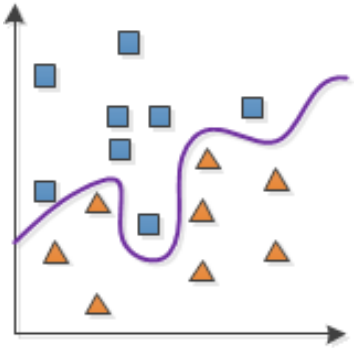


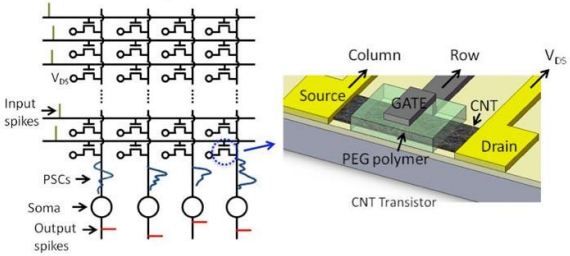


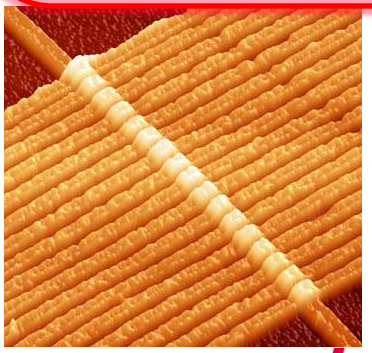
Constraints

Technology

Machine Learning







Innovations

Neuromorphic

Technology Constraints

- **Dark Silicon**
- **Defect tolerance**

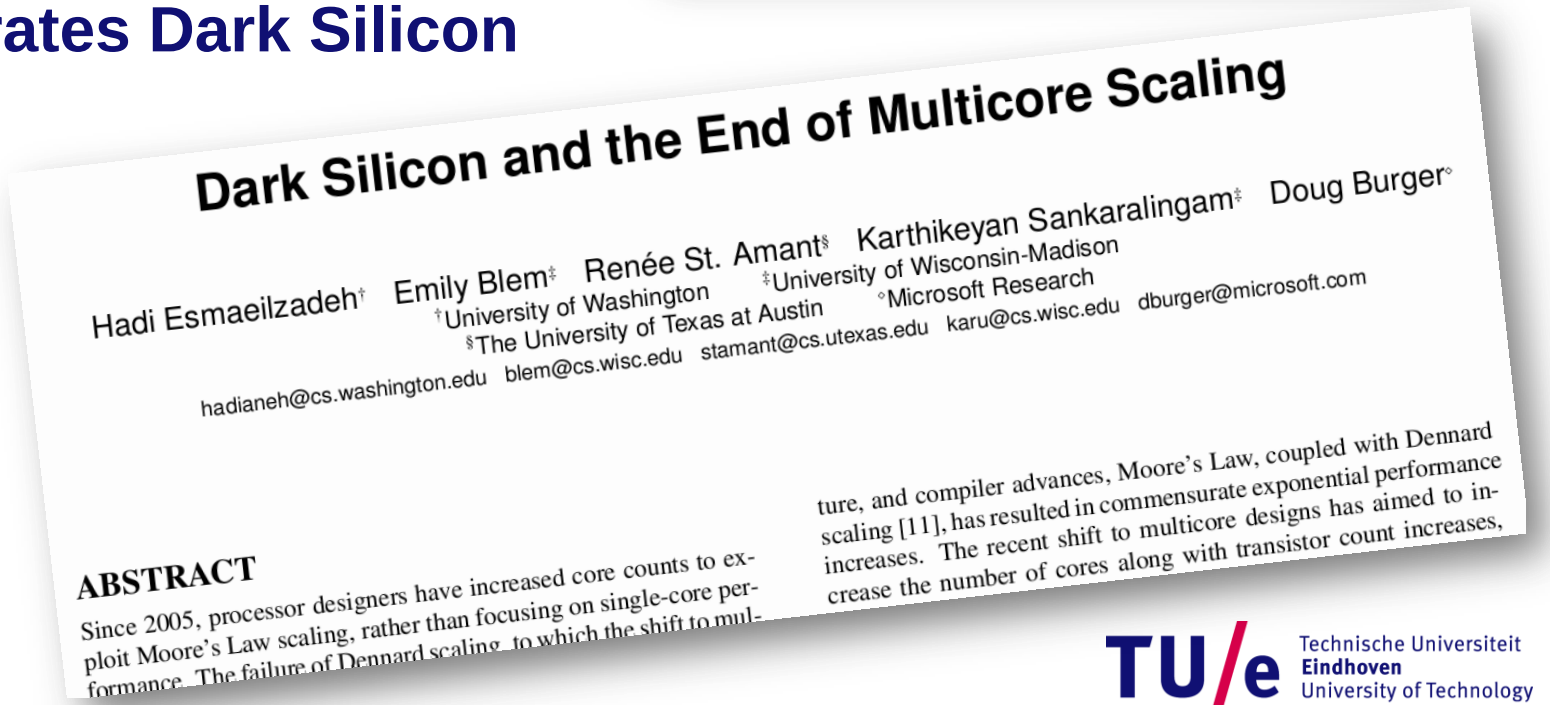
Dark Silicon

- What to do with chips that are too hot?
  - Reduce clock frequency
  - Go multi-core
- If chip is still too hot?
- Turn parts of the chip off!
- Generates Dark Silicon

EE Times News & Analysis

ARM CTO warns of dark silicon

Keynoter claims it may be impossible to utilize all available transistors on a chip



Dark Silicon and the End of Multicore Scaling

Hadi Esmaeilzadeh† Emily Blem‡ Renée St. Amant§ Karthikeyan Sankaralingam‡ Doug Burger◇

†University of Washington ‡University of Wisconsin-Madison

§The University of Texas at Austin ◇Microsoft Research

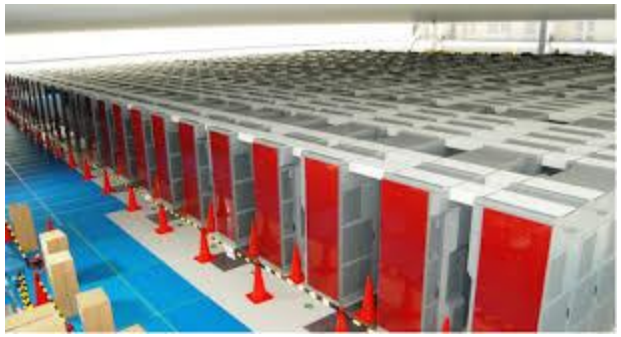
hadianeh@cs.washington.edu blem@cs.wisc.edu stamant@cs.utexas.edu karu@cs.wisc.edu dburger@microsoft.com

ABSTRACT

Since 2005, processor designers have increased core counts to exploit Moore's Law scaling, rather than focusing on single-core performance. The failure of Dennard scaling, to which the shift to multi-

ture, and compiler advances, Moore's Law, coupled with Dennard scaling [11], has resulted in commensurate exponential performance increases. The recent shift to multicore designs has aimed to increase the number of cores along with transistor count increases,

Energy Efficiency

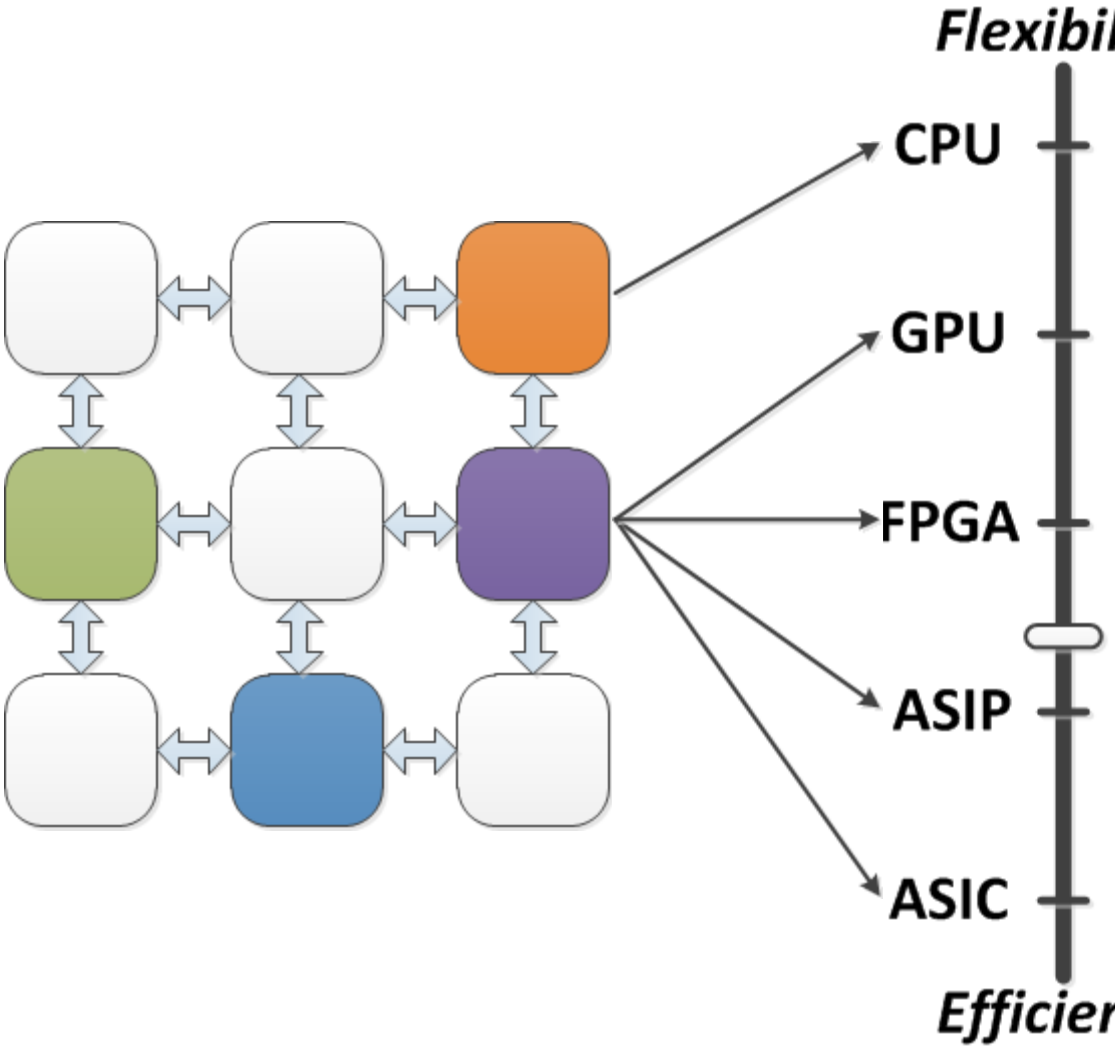


Super Computer (K computer, Fujitsu)
8.2 billion Megaflops => 9.9 million watts
~ 800 Megaflops / watt

Human Brain
2.2 billion Megaops => 20 watts
~ 110 Teraops / watt

iPad 2
170 Megaflops => 2.5 watts
~ 68 Megaflops / watt

Toward Heterogeneous Systems

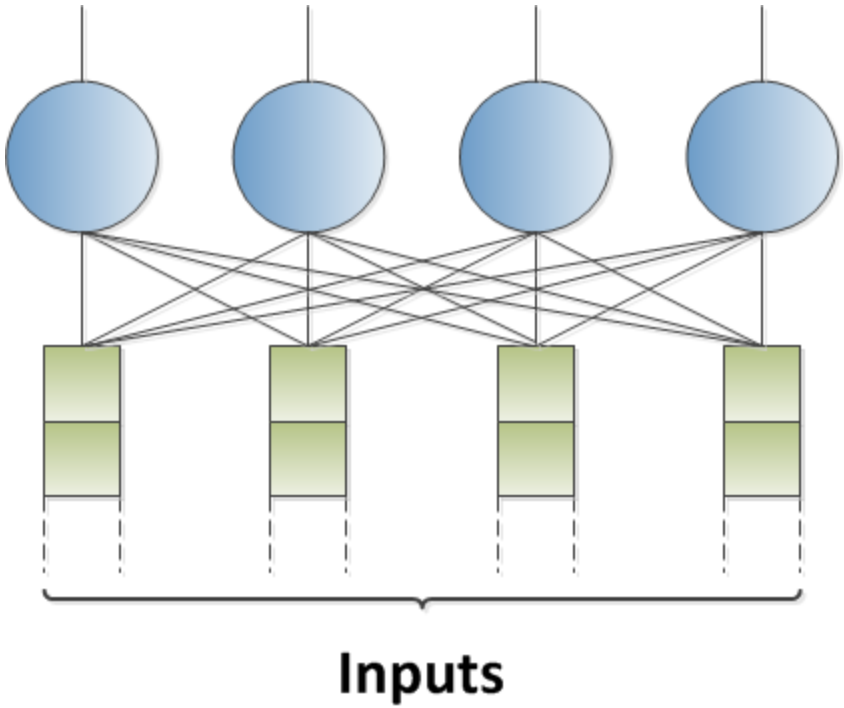


- Efficient accelerators
- Multi-purpose ASICs
- ANN is a candidate
  - Flexible functionality
  - State-of-the-art results
  - Parallelism

Developing ANN Accelerators


```
for i = 1:N
  Y[i] = Bias[i]
  for k = 1:K
    Y[i] += X[k] * W[i][k]
  Y[i] = Sigmoid(Y[i])
```

$$y_i = \varphi\left(b_i + \sum_k x_k \cdot w_{ik} \right)$$



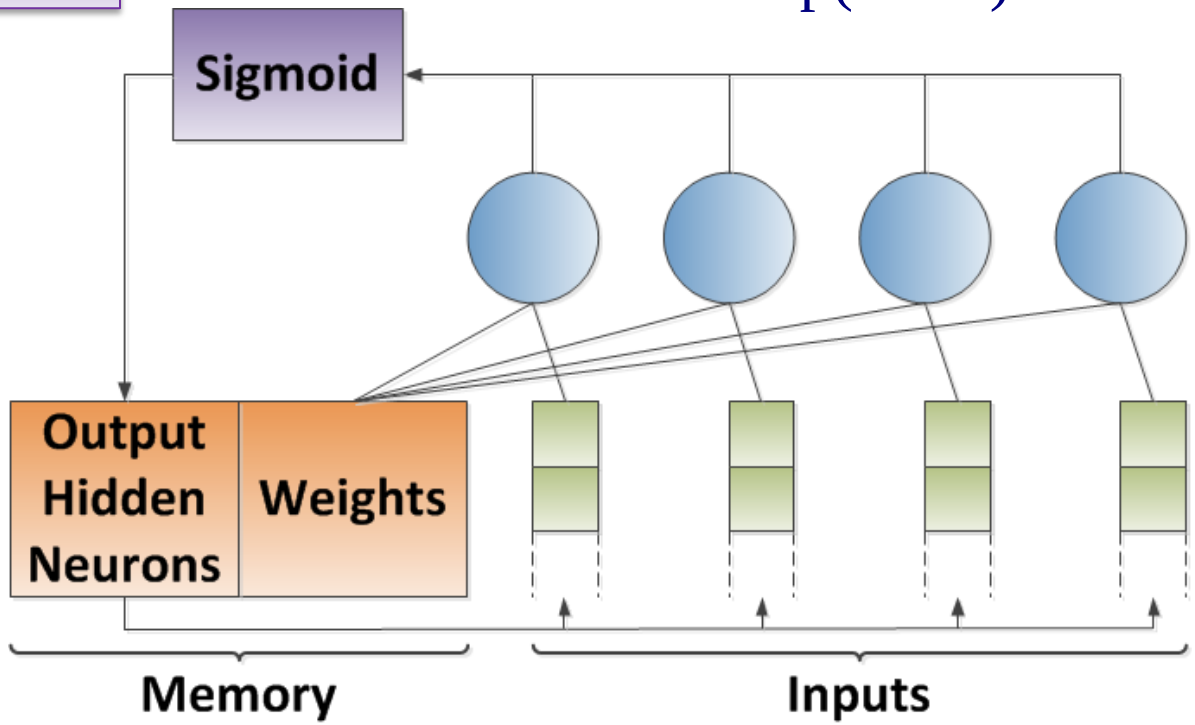
Time-Multiplexed Accelerator


```
for i = 1:N
  Y[i] = Bias[i]
  for k = 1:K
    Y[i] += X[k] * W[i][k]
  Y[i] = Sigmoid(Y[i])
```

$$y_i = \varphi\left(b_i + \sum_k x_k \cdot w_{ik} \right)$$

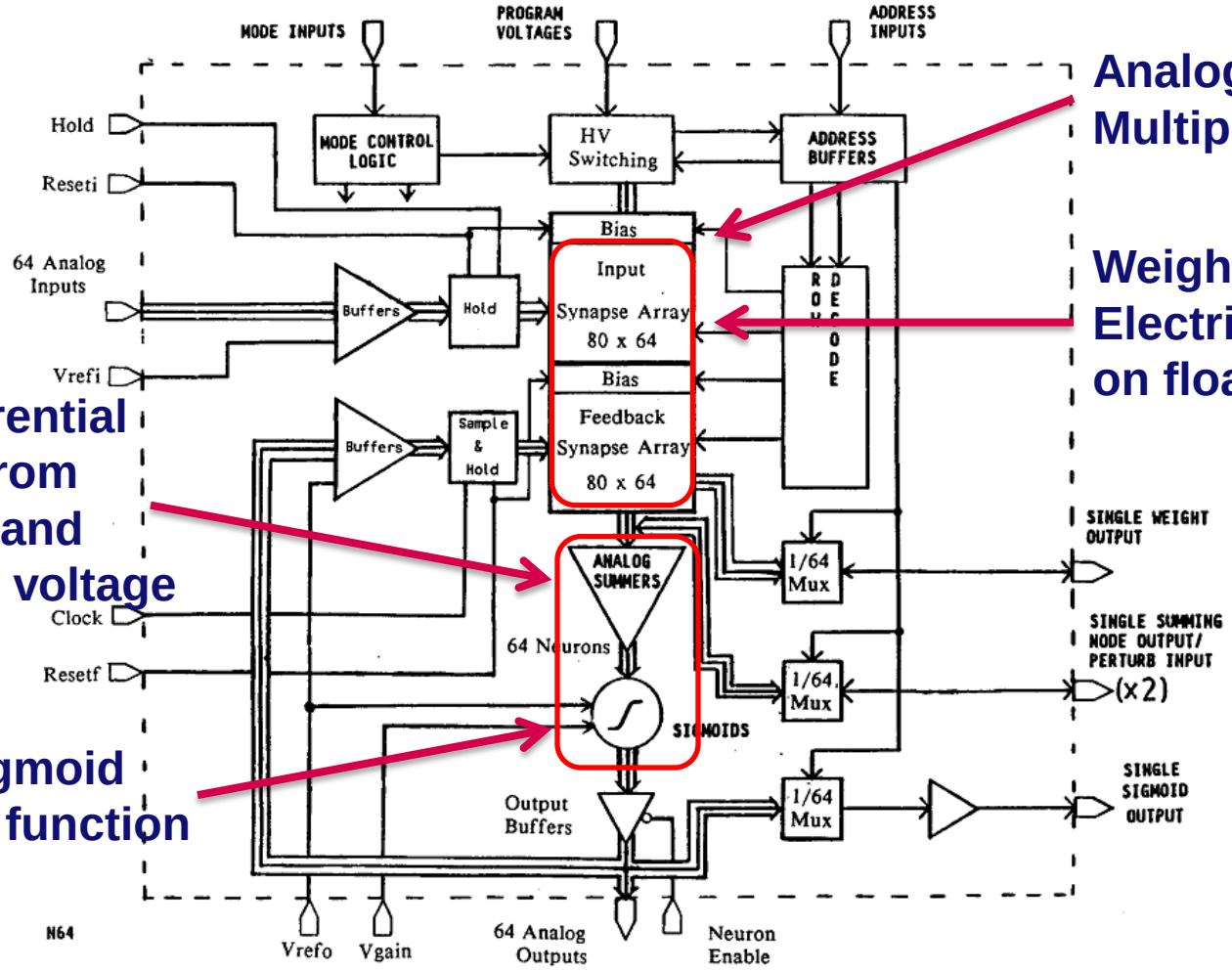
$$\varphi(v) = \frac{1}{1+\exp(-a \cdot x)}$$

- **Load Bias**
  - X[1:N] = 1
  - W[i][1] = Bias[i]
- **Perform MACC**
- **Sigmoid**
  - Approximate



Analog Intel ETANN 1990

- **Electrically Trainable Analog Neural Network**



Analog Gilbert-Multiplier Circuits

Weights stored as Electrical Charge on floating gates

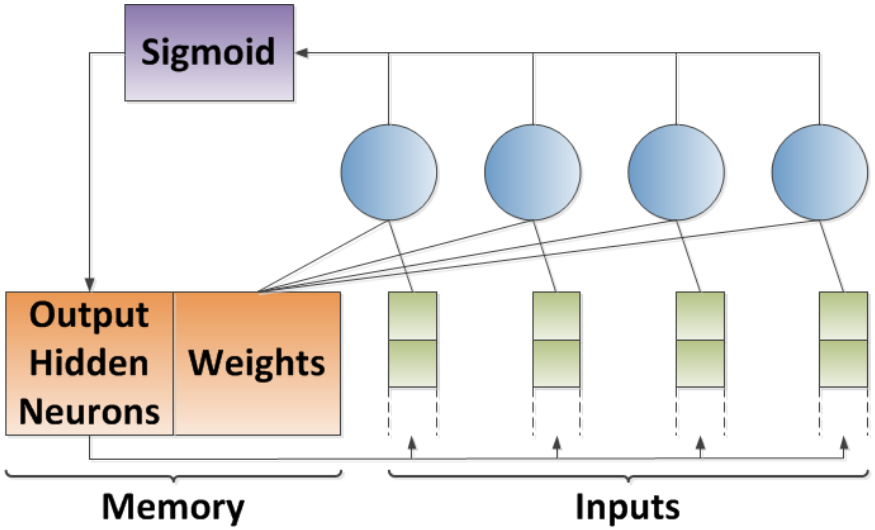
Sum differential currents from synapses and convert to voltage

Analog sigmoid activation function

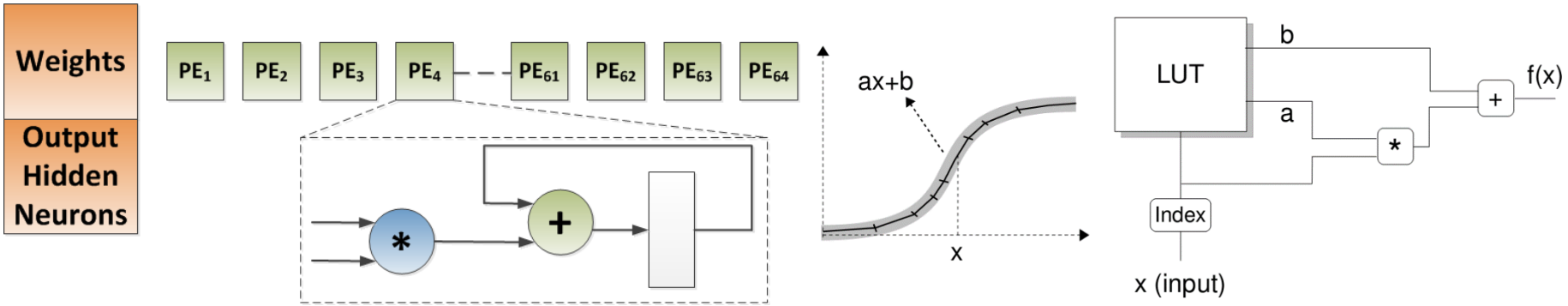
Digital Implementation

- **Sigmoid Function**
  - **Look Up Table**
  - **Use linear approximation**

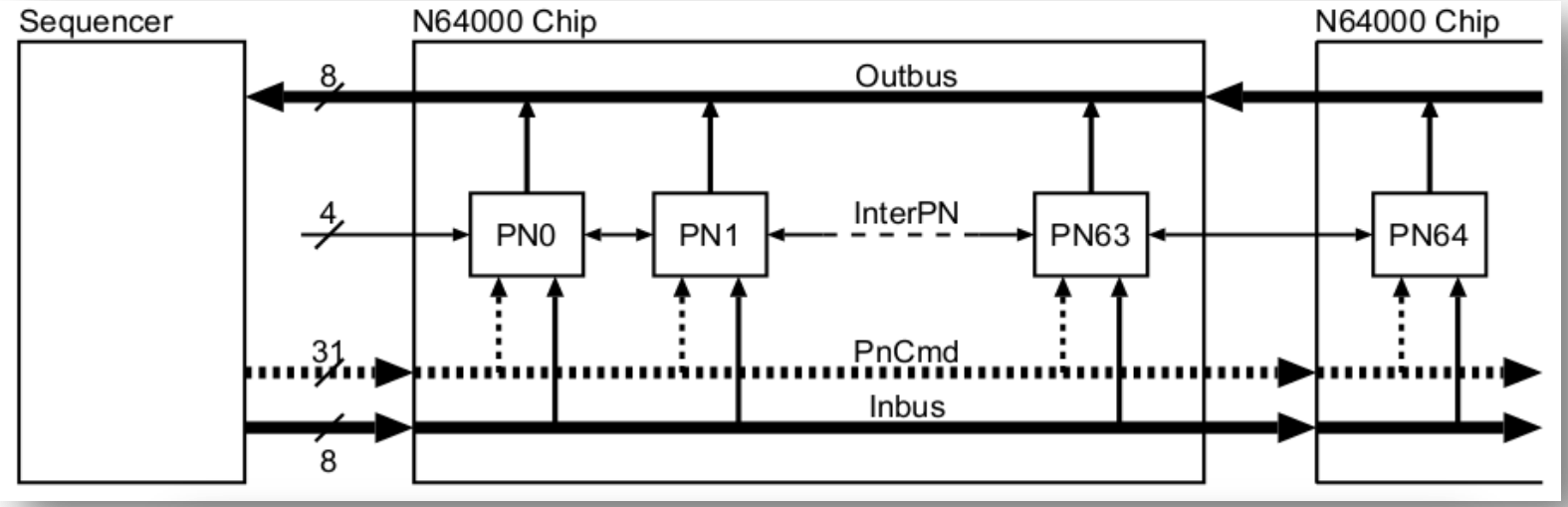
$$\varphi(x) \approx b_i + a_i \cdot x$$

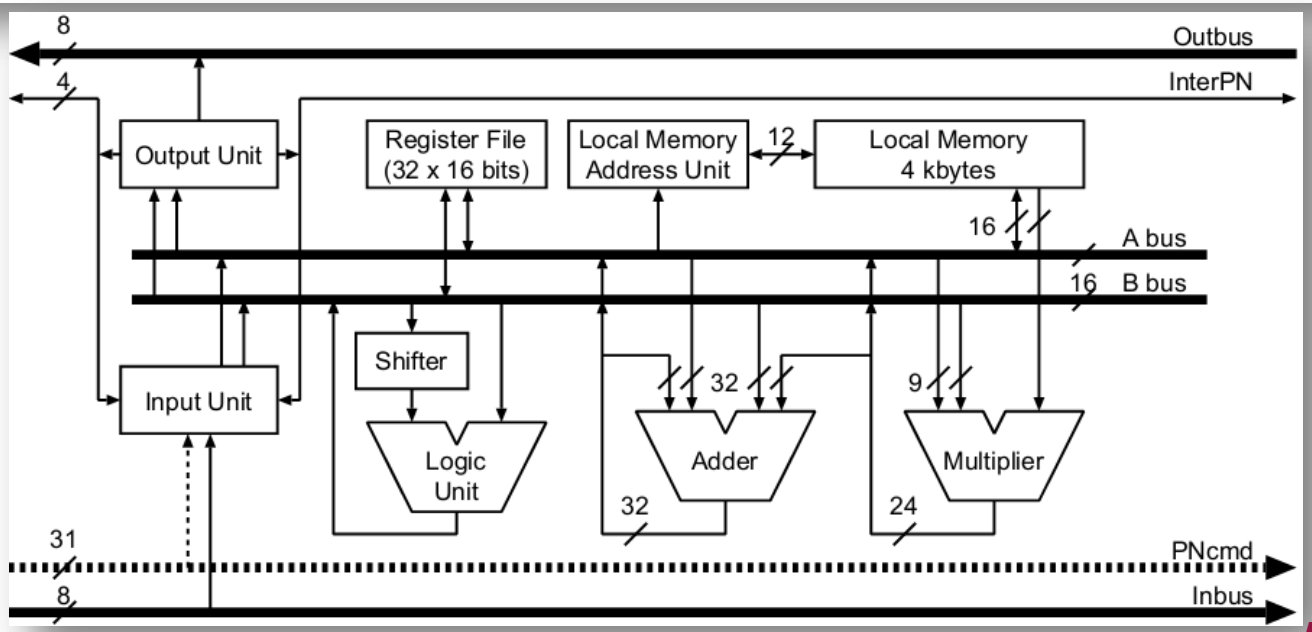


- **SIMD Multiply Accumulate**



SIMD design Adaptive Solutions N64000



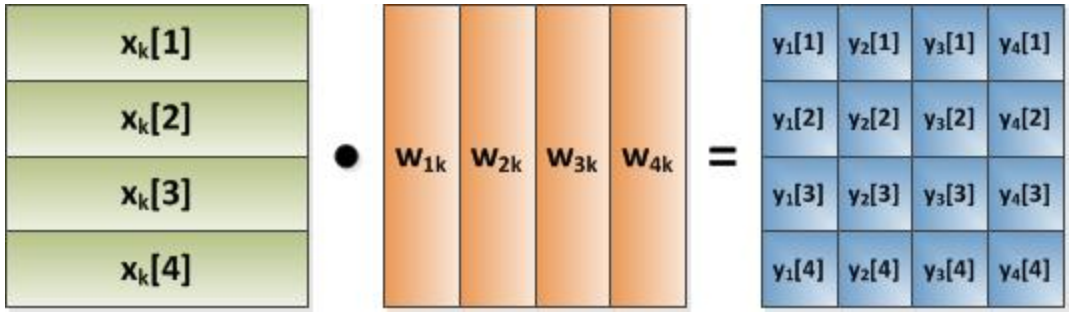


Conversion to vector operations

$$y_i[n] = \varphi\left(b_i + \sum_k x_k[n] \cdot w_{ik} \right)$$

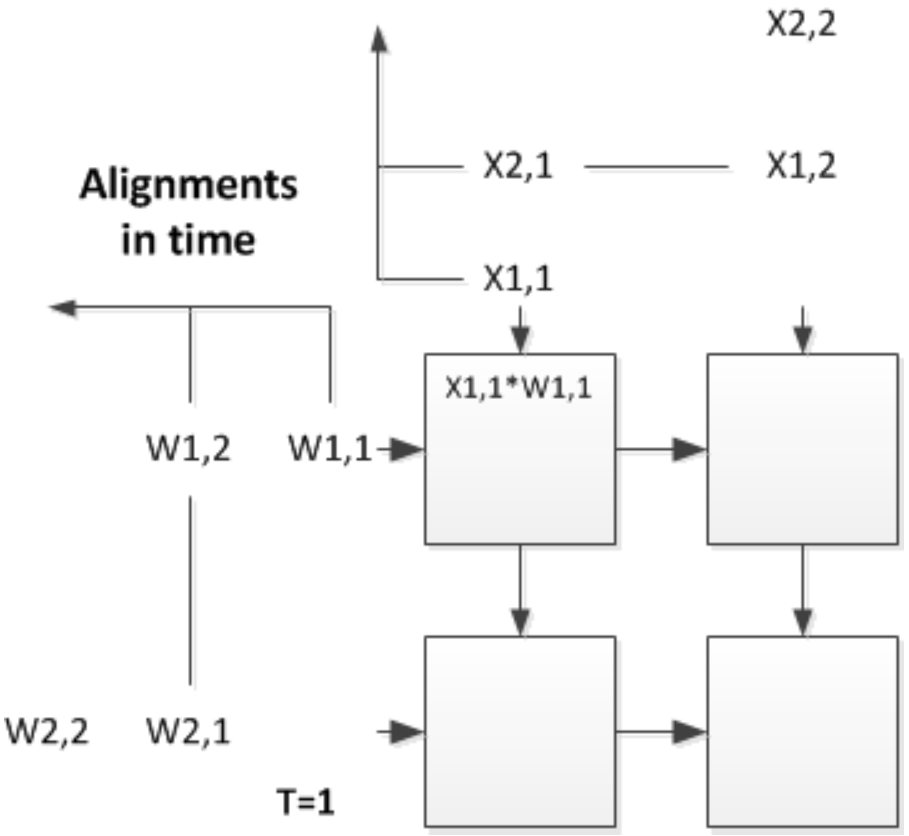
$$\mathbf{y}[n] = \varphi\left(\mathbf{b} + \mathbf{x}[n] \cdot \mathbf{W}\right)$$

$$\mathbf{Y} = \varphi\left(\mathbf{b} + \mathbf{X} \cdot \mathbf{W}\right)$$



Systolic Matrix Multiplication

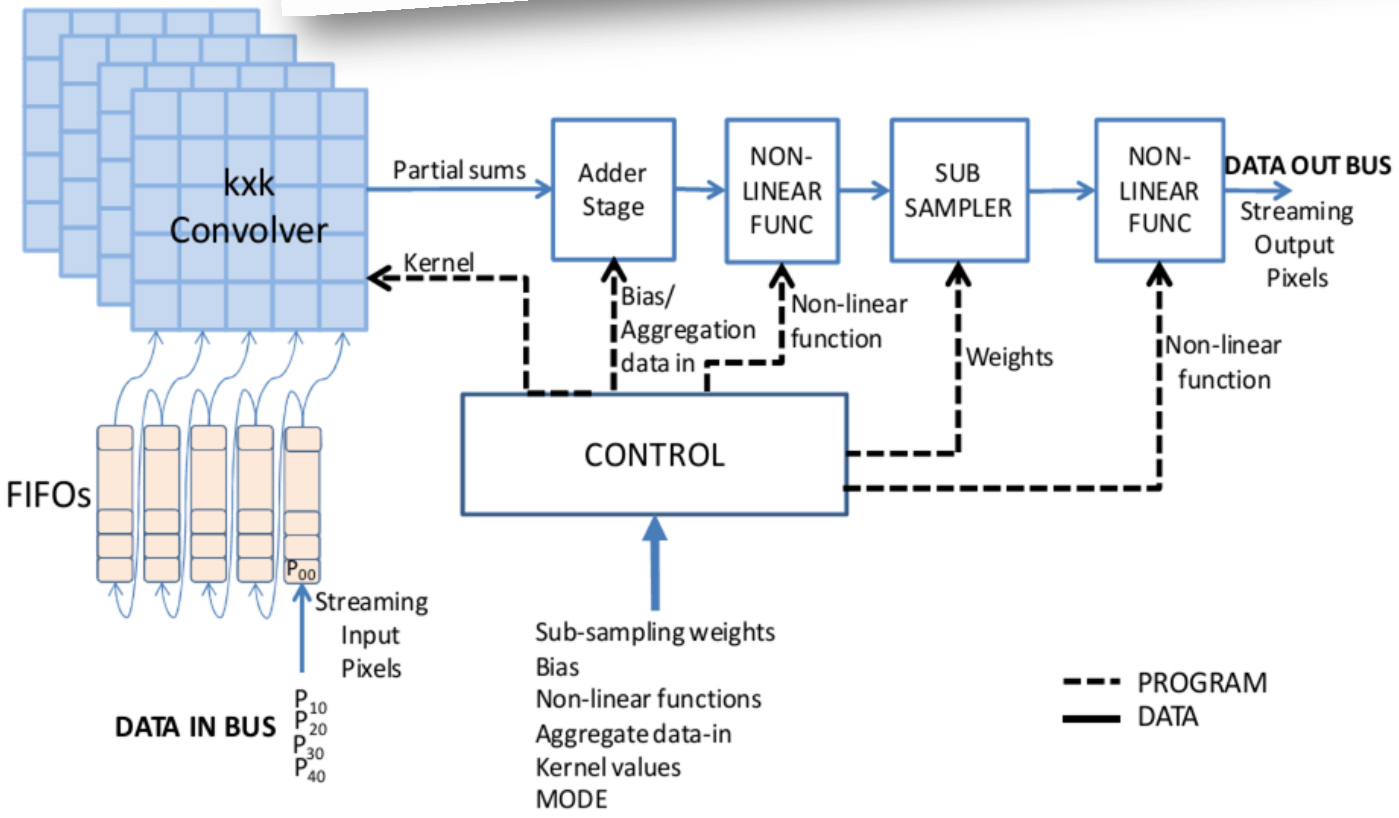
- **Siemens MA16**
  - **High efficiency**
  - **Low flexibility**



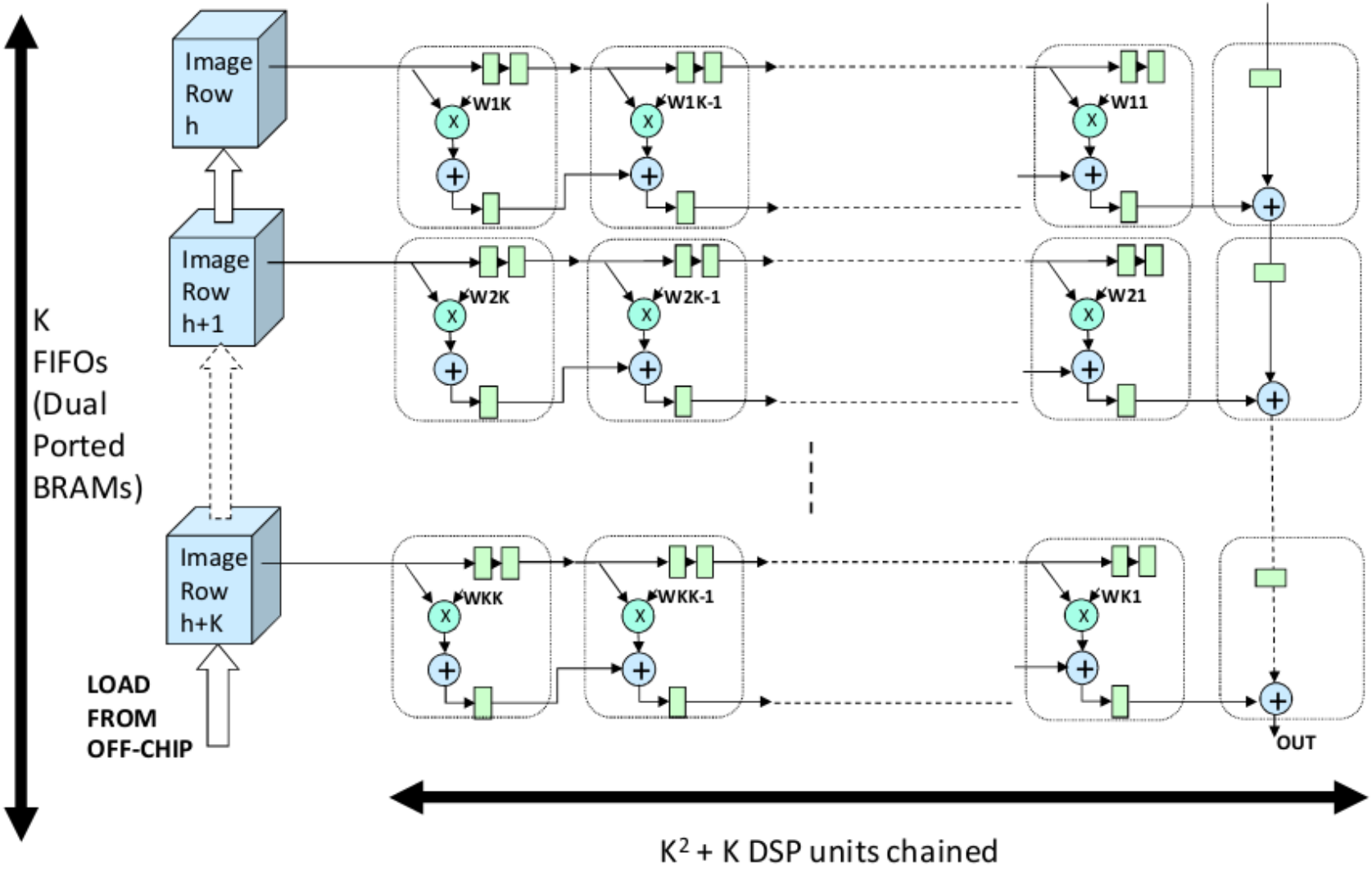
An example research accelerator

A Dynamically Configurable Coprocessor for Convolutional Neural Networks

Srimat Chakradhar, Murugan Sankaradas, Venkata Jakkula and Srihari Cadambi
NEC Laboratories America, Inc.
4 Independence Way, Princeton NJ 08540.
{chak, murugs, jakkula, cadambi}@nec-labs.com

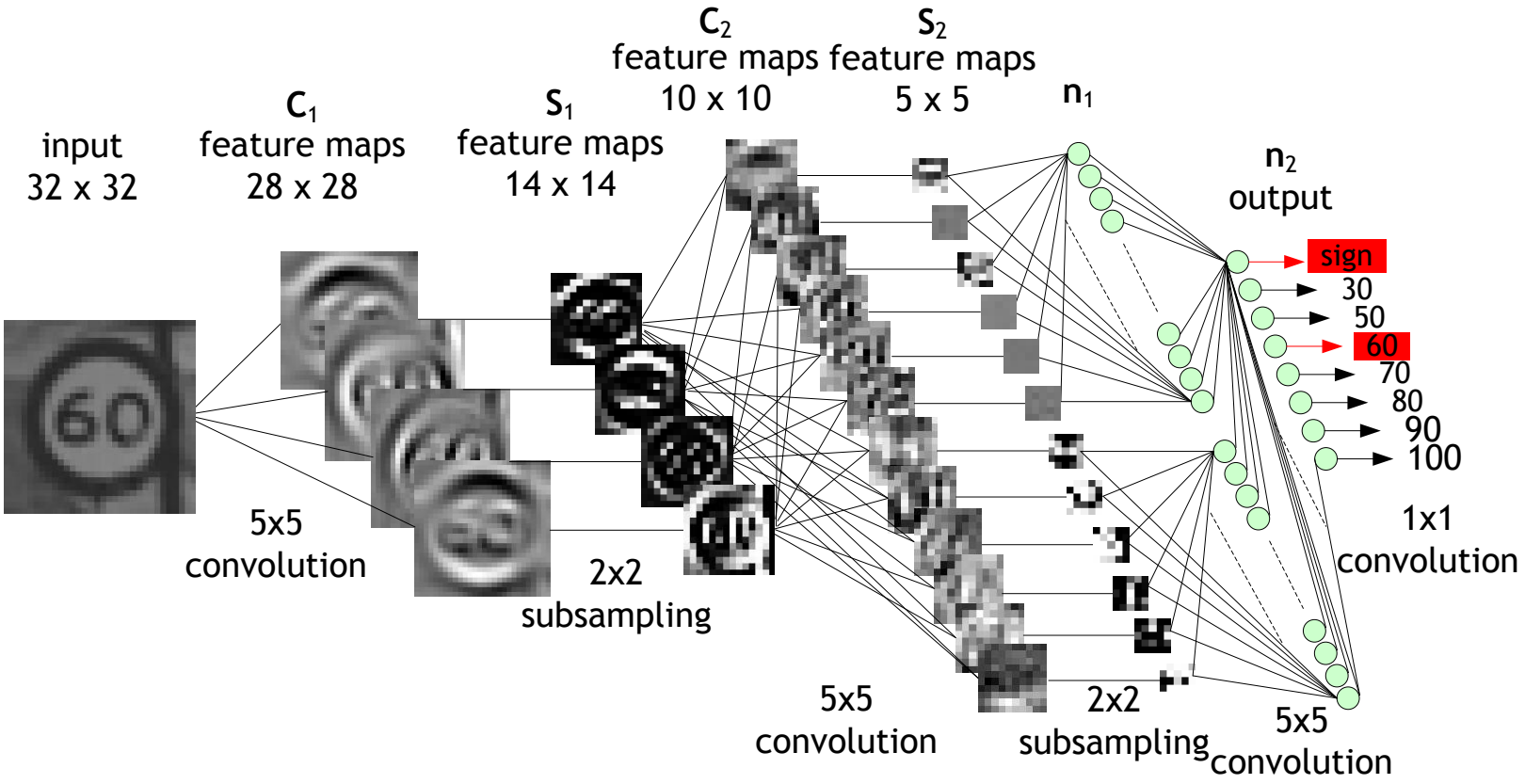


Systolic 2D Convolution



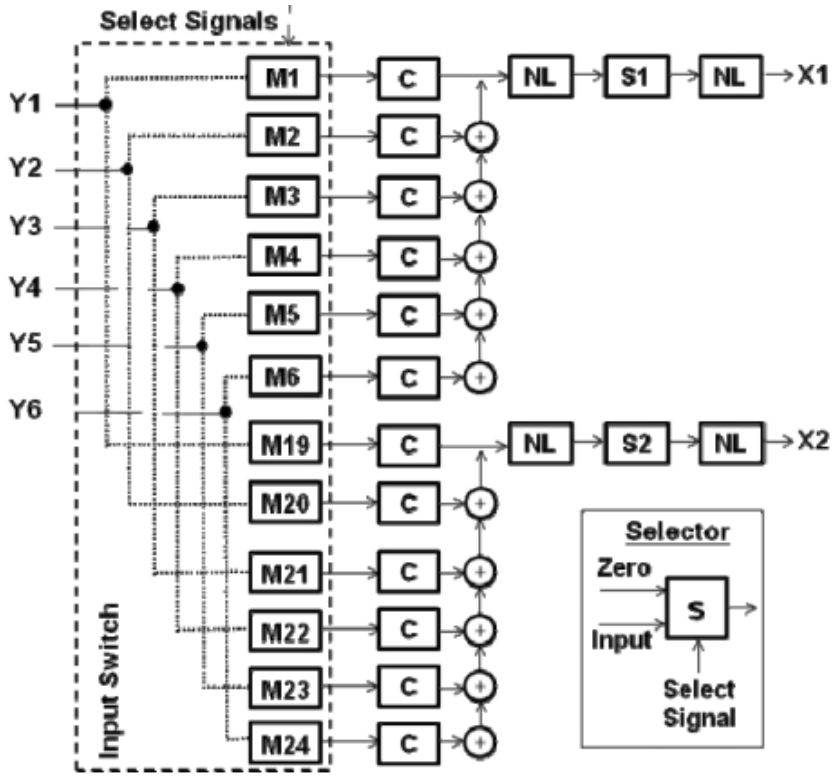
Convolutional Neural Network

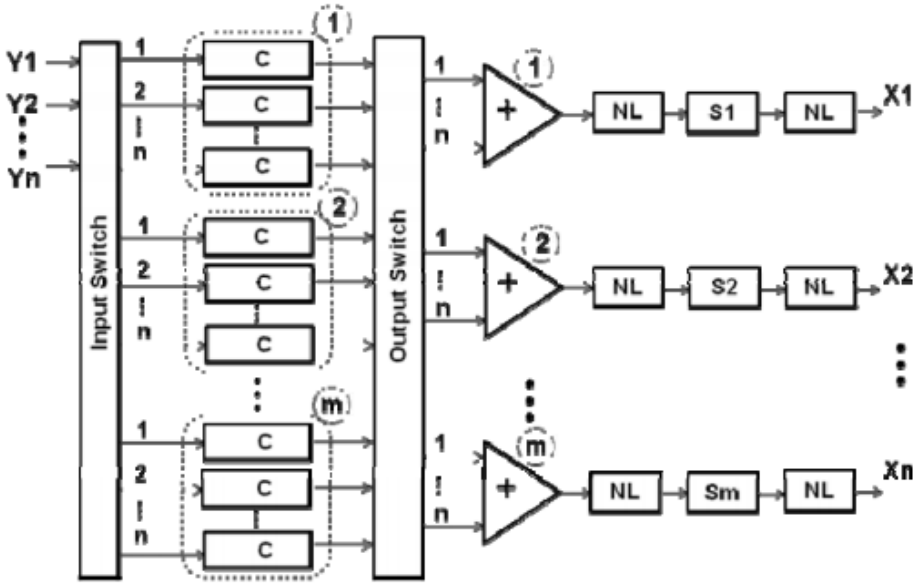
- **Data reuse**



Reduce Memory Accesses

- Configurable Number of Input Maps
- Configurable Number of Output Maps

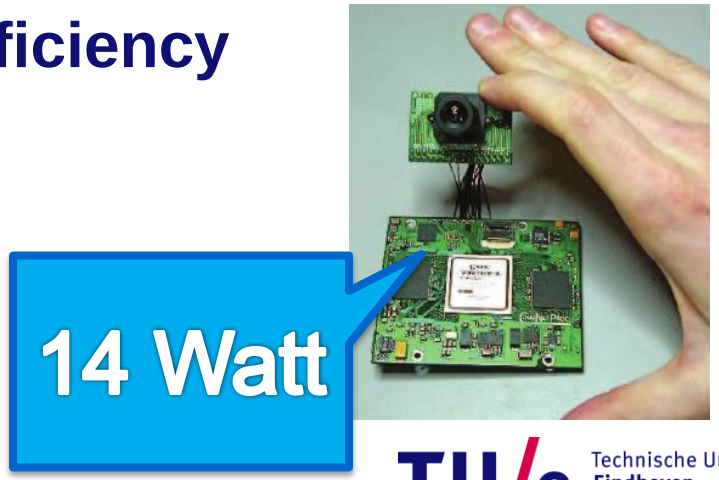




Is it worth the effort?

CNN (640 x 480 pixels input image)	**Multicore** (Xeon @ 2.33 Ghz, 8 Cores, 16 GB) BLAS	**GPU** (C870 @ 1.35 Ghz, 1.5 GB RAM) PCIe	**CNP** (FPGA @200 MHz)	**DC-CNN** @ 120 Mhz 20 conv., 128-bit port width, PCI		**Speedup** of DC-CNN		
				Compute time	Transfer time	Over 2.3 GHz, 8-core	Over 1.35 GHz, 128-core GPU	Over CNP
Automotive Safety	110 ms	85 ms	-	13 ms	11 ms	8.5x	6.5x	-
Video Surveillance	212 ms	163 ms	-	27 ms	34 ms	7.8x	6.0x	-
Face Recognition	217 ms	167 ms	-	42 ms	11 ms	5.2x	4.0x	-
Mobile Robot Vision	147 ms	114 ms	100 ms	21 ms	11 ms	7.0x	5.4x	4.8x
Face Detection	136 ms	105 ms	-	24 ms	11 ms	5.7x	4.4x	-

- **More important the energy efficiency**

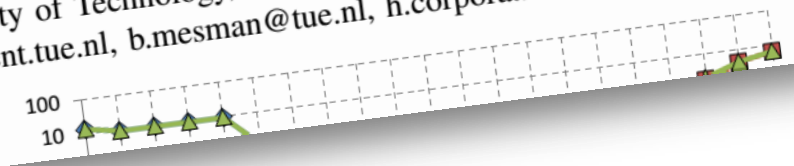


More Flexibility and Better Memory Behaviour?

Memory-Centric Accelerator Design for Convolutional Neural Networks

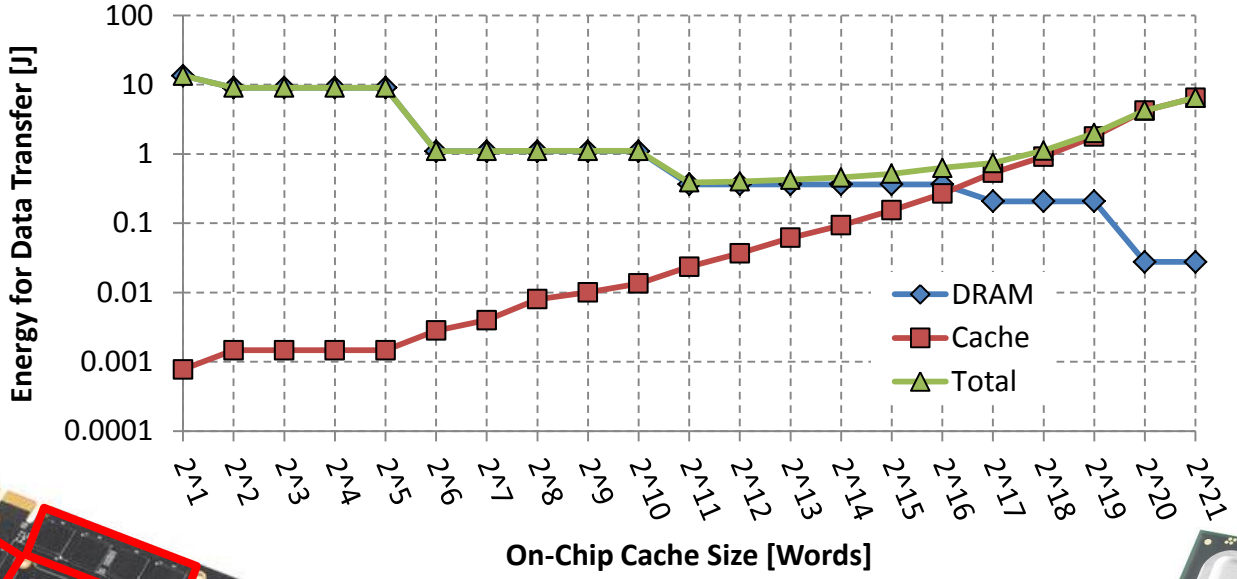
Maurice Peemen, Arnaud A. A. Setio, Bart Mesman and Henk Corporaal
Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands
Email: m.c.j.peemen@tue.nl, arnaud.arindra.adiyoso@student.tue.nl, b.mesman@tue.nl, h.corporaal@tue.nl

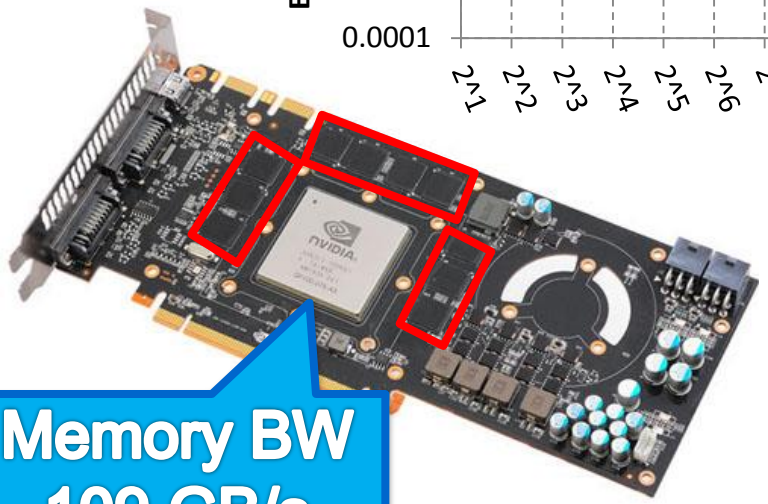
***Abstract*—In the near future, cameras will be used everywhere**
as flexible sensors for numerous applications. For mobility and



The performance bottleneck

- Huge data transfer requirements (3.4 billion per layer)
- Exploit data reuse with local memories



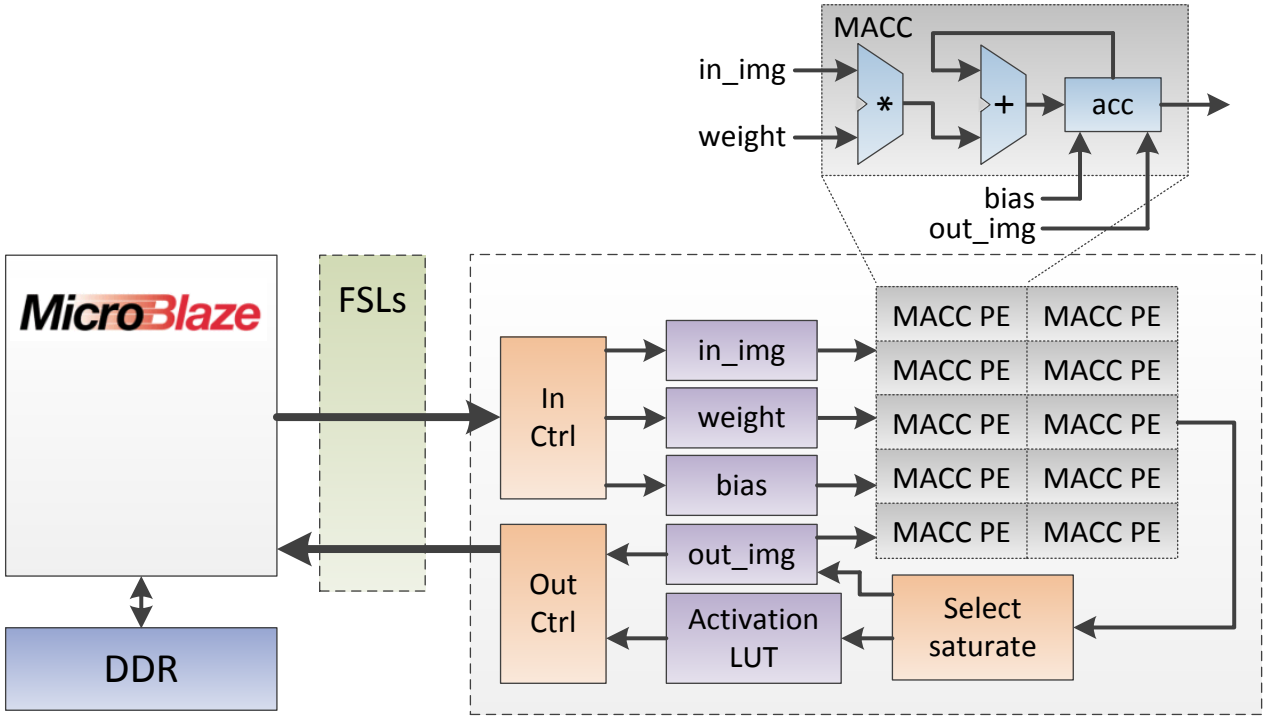


Memory BW
109 GB/s

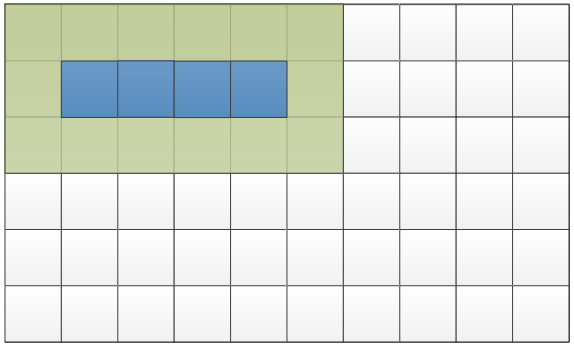
3 MB Cache

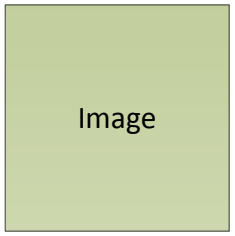
Accelerator Template

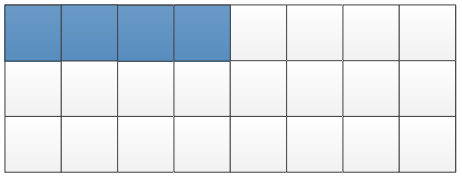
- FPGA prototyping platform: Xilinx Virtex 6
- Designed with Vivado High Level Synthesis (HLS)

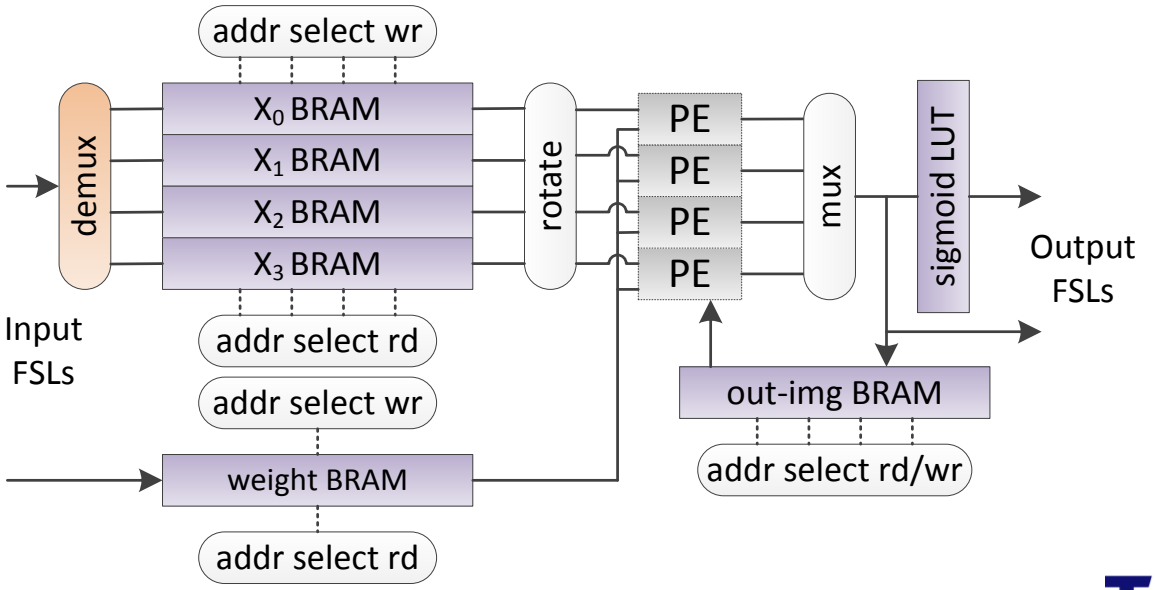


Programmable Buffers

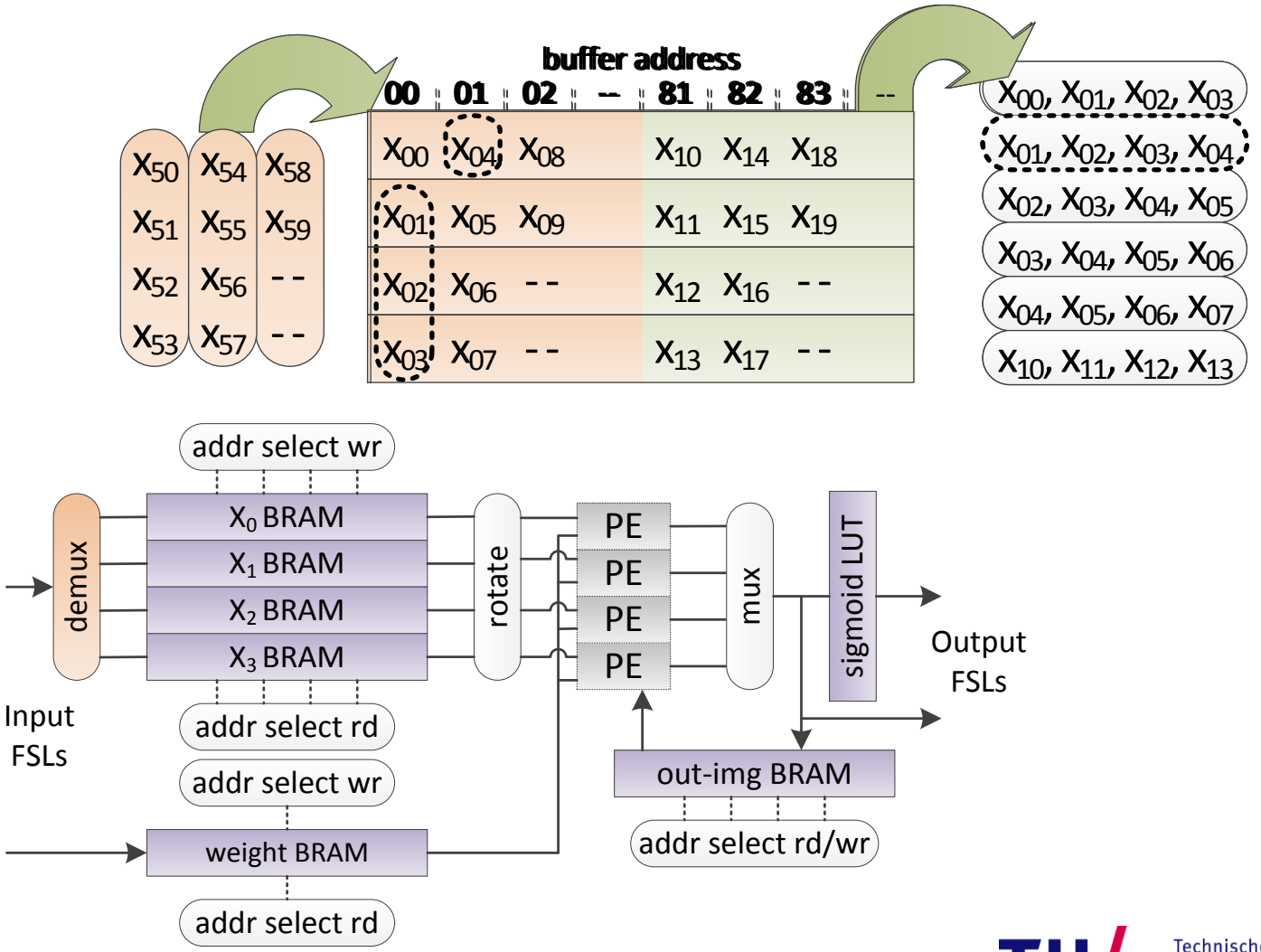








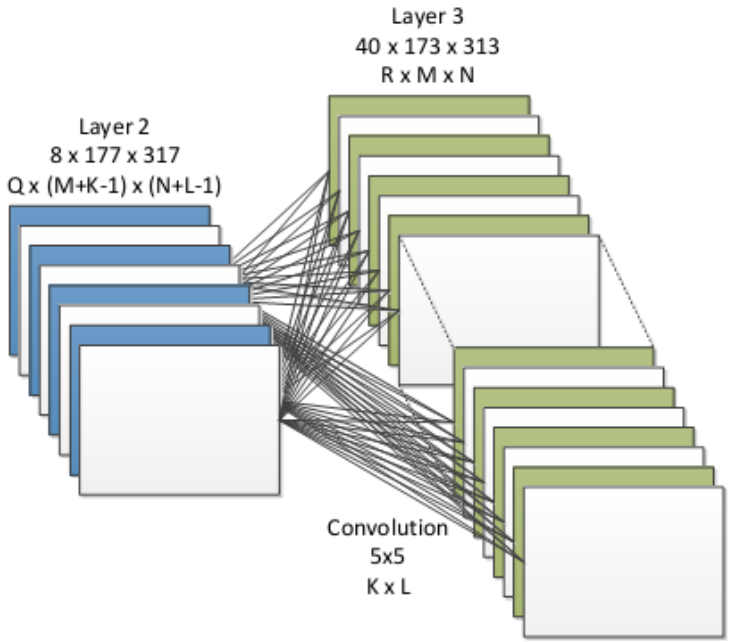
Programmable Buffers



What would be the best compute order?

- **Small memories have low energy per access**
  - **Area and Latency advantage**
- **Big memories can exploit more data reuse**

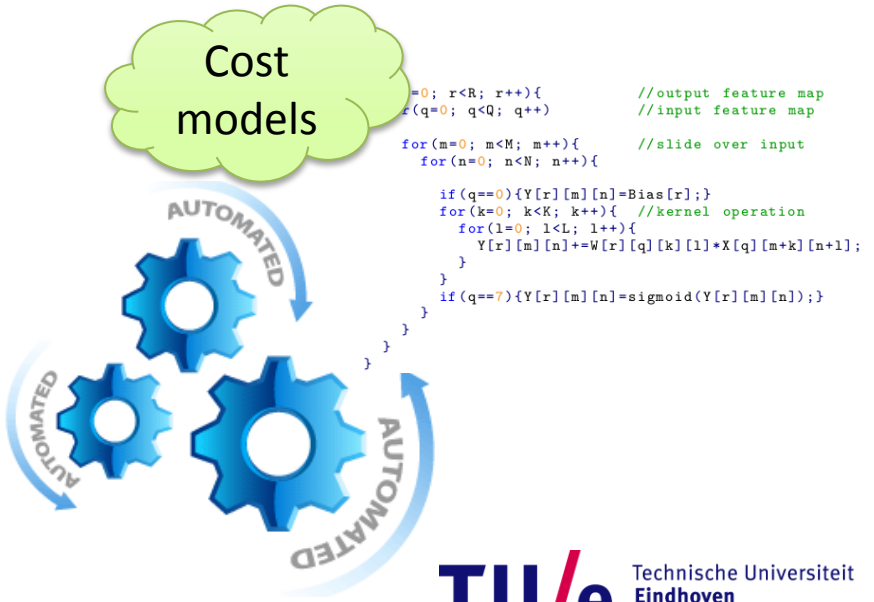
```
for(r=0; r<R; r++){                  //output feature map
  for(q=0; q<Q; q++)                 //input feature map

    for(m=0; m<M; m++){              //slide over input
      for(n=0; n<N; n++){

        if(q==0){Y[r][m][n]=Bias[r];}
        for(k=0; k<K; k++){  //kernel operation
          for(l=0; l<L; l++){
            Y[r][m][n]+=W[r][q][k][l]*X[q][m+k][n+l]
          }
        }
        if(q==7){Y[r][m][n]=sigmoid(Y[r][m][n]);}
      }
    }
  }
}
```



Improve by locality driven synthesis

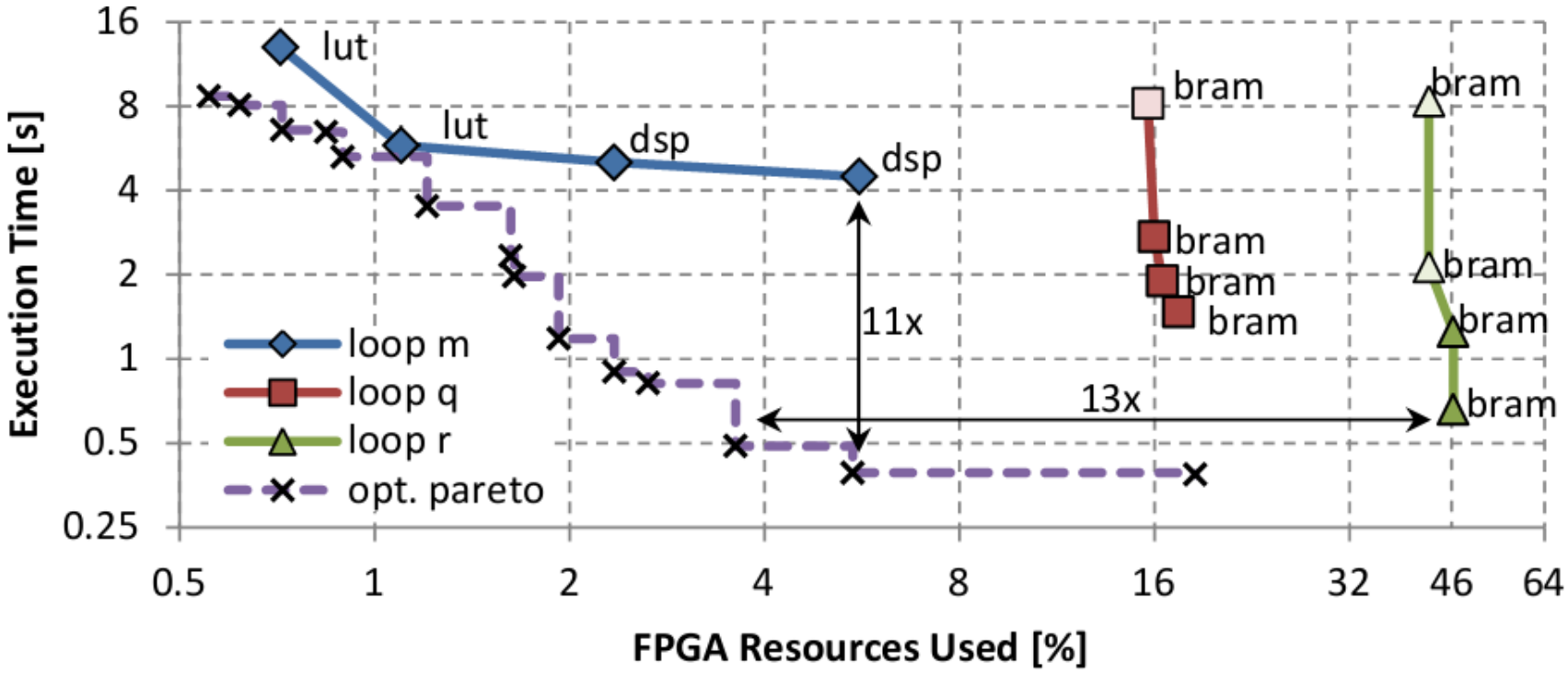
- **Loop Transformations**
  - **Interchange**
  - **Tiling**
- **Reduce reuse distance**
- **A huge design space!**
- **Use a framework with:**
  - **Reuse detection**
  - **Model utilized reuse**
  - **Model required buffer size**
  - **Optimize for buffer size**

```
for(r=0; r<R; r++){              //output feature map
  for(q=0; q<Q; q++)             //input feature map

    for(m=0; m<M; m++){          //slide over input
      for(n=0; n<N; n++){

        if(q==0){Y[r][m][n]=Bias[r];}
        for(k=0; k<K; k++){  //kernel operation
          for(l=0; l<L; l++){
            Y[r][m][n]+=W[r][q][k][l]*X[q][m+k][n+l];
          }
        }
        if(q==7){Y[r][m][n]=sigmoid(Y[r][m][n]);}
      }
    }
  }
}
```



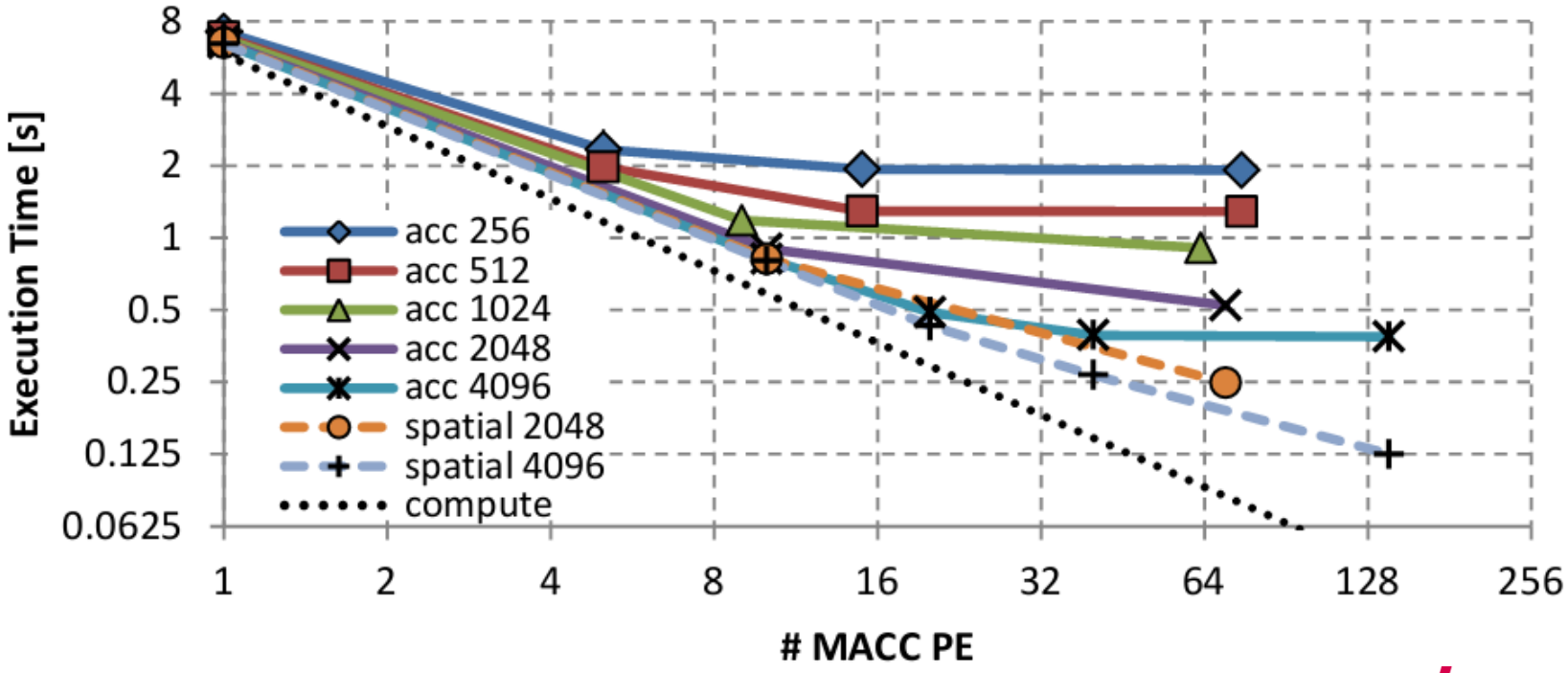
Compared to manually optimized order

- Up to 13x resource reduction
- Up to 11x performance increase



Memory bandwidth requirements?

- Data layout transformation
- Bandwidth up to 150 MB/s
- Better than an optimized Intel implementation



What do we achieve?

- Small but flexible accelerators
- Up to 13x smaller
- Up to 11x faster

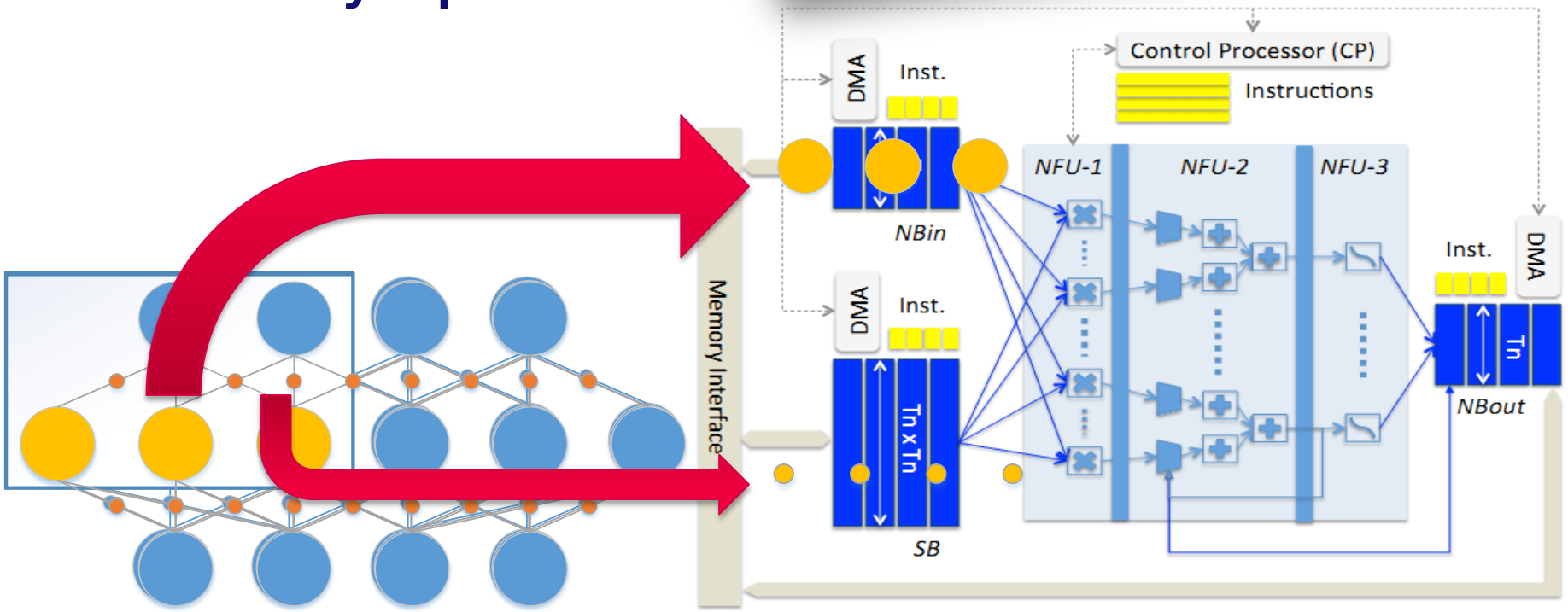
- XPower Analyzer 4.5 Watt
- External RAM 0.5 Watt

State-of-the-art accelerator

- Split buffers for BW
- Partial layer
- Fetch input neurons
- Fetch synapses

DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning

Tianshi Chen, SKLCA, ICT, China
Zidong Du, SKLCA, ICT, China
Ninghui Sun, SKLCA, ICT, China
Jia Wang, SKLCA, ICT, China
Chengyong Wu, SKLCA, ICT, China
Yunji Chen, SKLCA, ICT, China
Olivier Temam, Inria, France

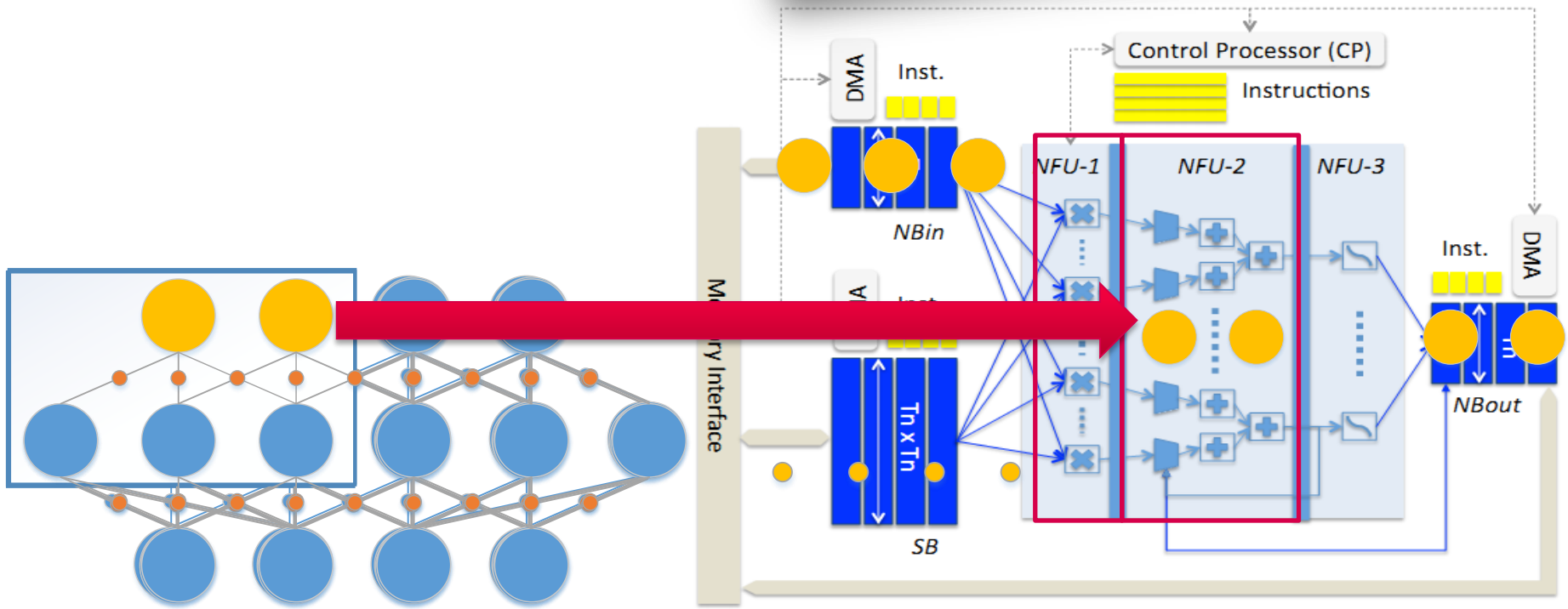


State-of-the-art accelerator

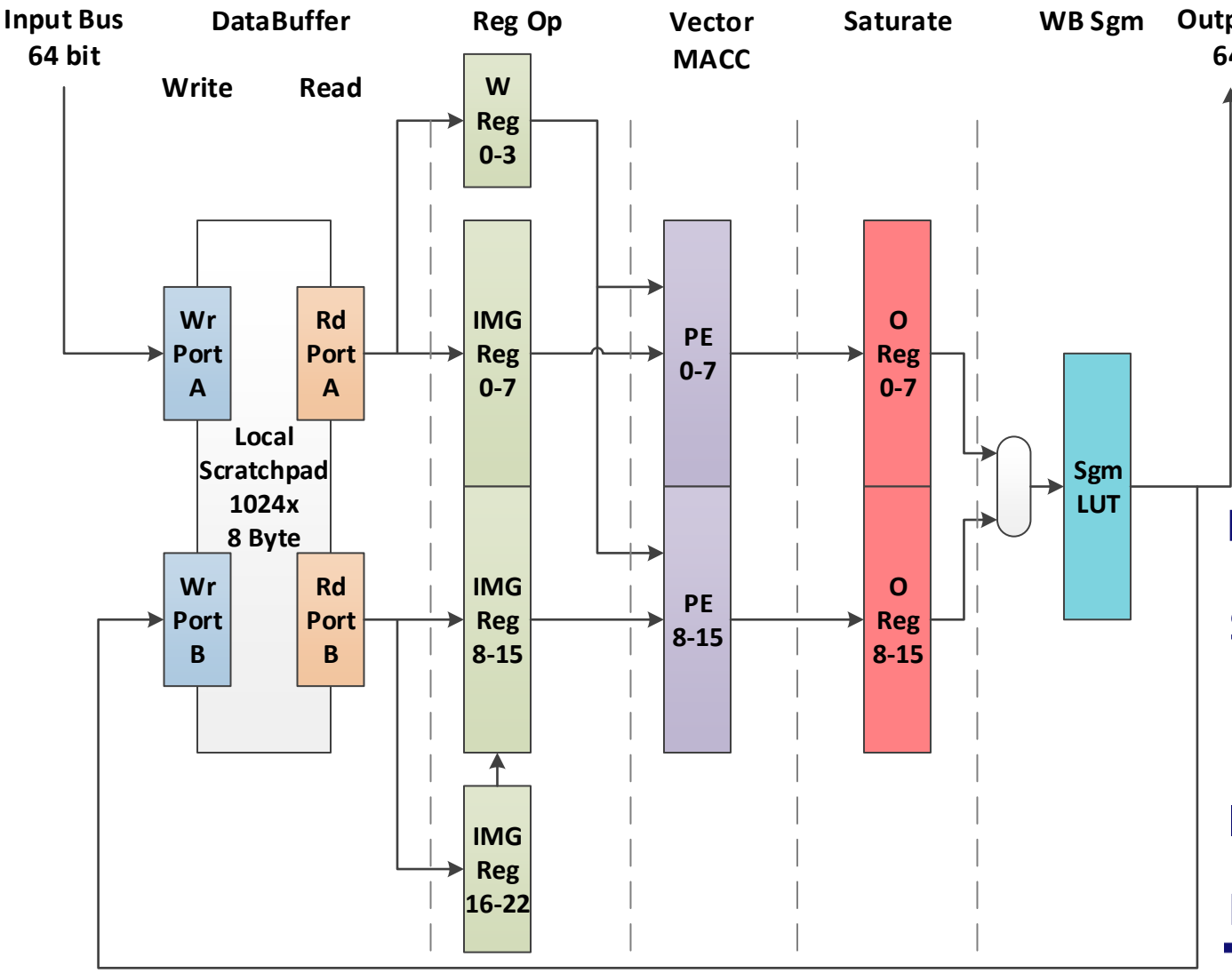
- Multiply
- Sum neurons inputs
- Backup partial sums

DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning

Tianshi Chen, SKLCA, ICT, China
Zidong Du, SKLCA, ICT, China
Ninghui Sun, SKLCA, ICT, China
Jia Wang, SKLCA, ICT, China
Chengyong Wu, SKLCA, ICT, China
Yunji Chen, SKLCA, ICT, China
Olivier Temam, Inria, France



The Neuro Vector Engine



Programmable

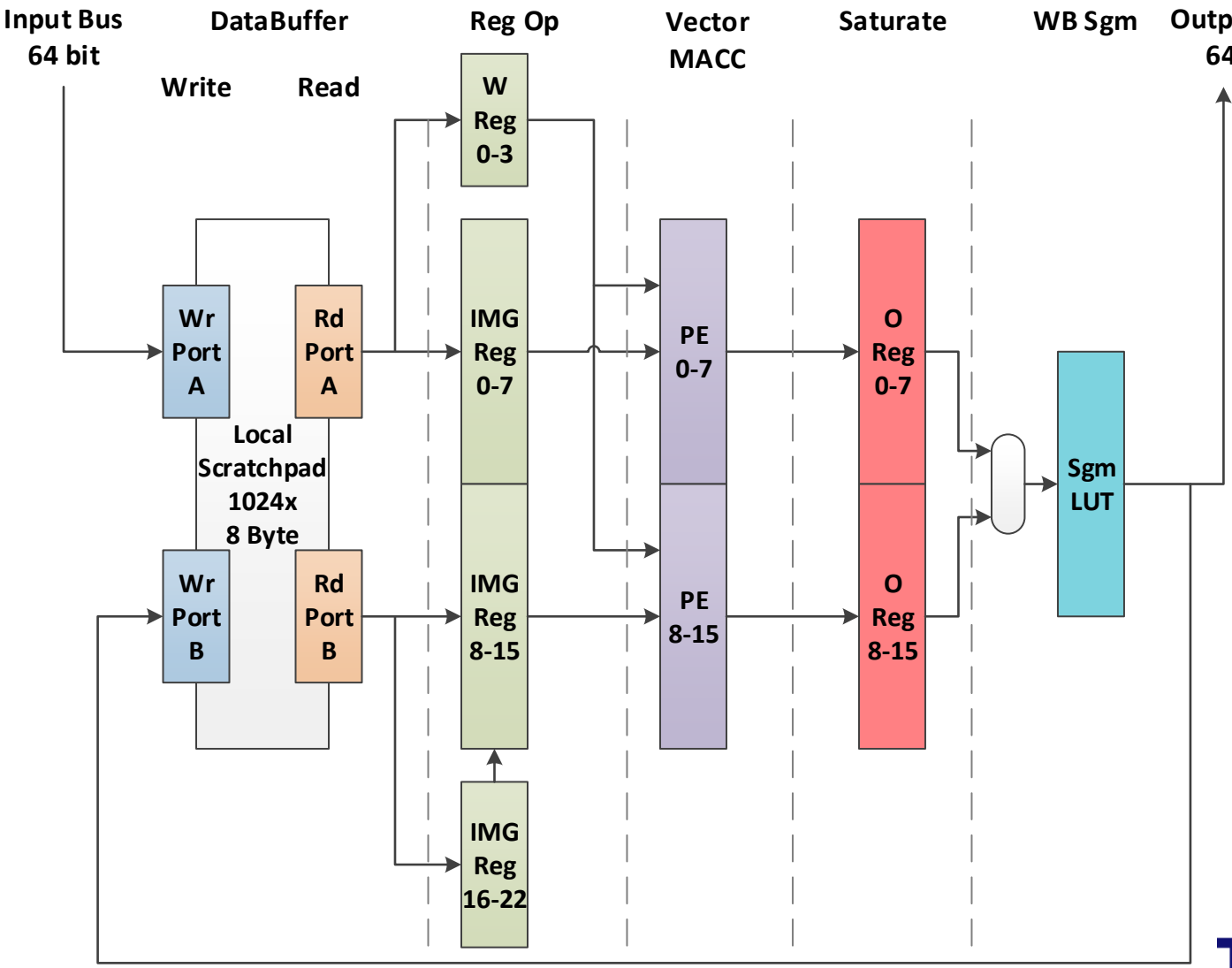
SIMD operations

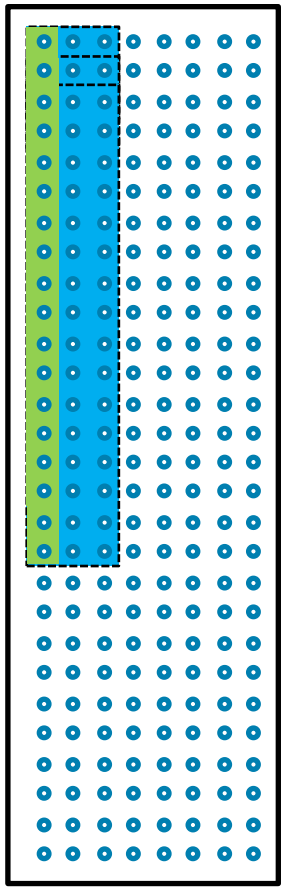
Deep pipelining

ILP

Locality oriented

NVE Operation





VLIW Programming Model

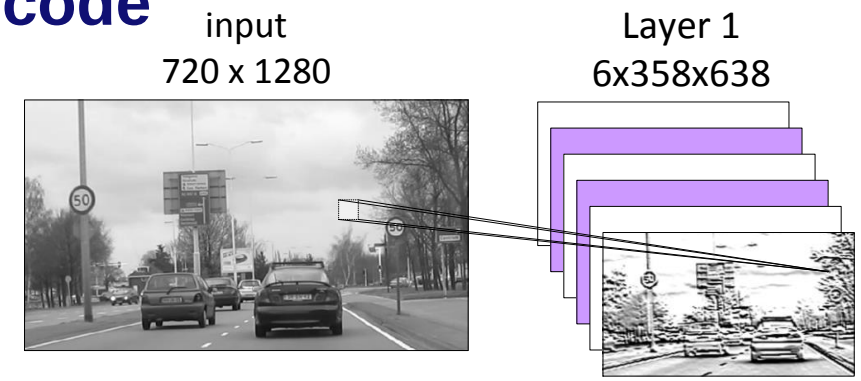
	Write Local
weights	Wr[b1 w0 w1 -], 0
	Wr[w2 w3 w4 -], 1
	Wr[w5 w6 w7 w8], 2
prolog img	Wr[c0,i0-i7], 4
	Wr[c0,i8-i15], 5
	Wr[c0,i16-i17 - - - -], 6
	Wr[c1,i0-i7], 7
	Wr[c1,i8-i15], 8
	Wr[c1,i16-i17 - - - -], 9
	Wr[c2,i0-i7], 10
	Wr[c2,i8-i15], 11
	Wr[c2,i16-i17 - - - -], 12

- 3x3 Convolution filter
- Software Pipelining
- Steady state 10 cycles
  - 16 neighboring 3x3 convolutions
  - 144 Multiply Accumulate ops
- Code reuse with instruction buffer

Write Local	Read Local	Reg Operation	EX VMAC	WB Sgm
10 Cycles	Ld a 0, [b0 w0-1]	Set i3, Shift i0 i1, Shift W	MAC,w6, c2 i0	
	Ld ab 4 5, [c0,i0-15]	Shift i0 i1, Shift W	MAC,w7, c2 i1	
Wr[c3,i0-i7], 4	Ld b 6, [c0,i16-17]	Set W0	MAC,w8, c2 i2	
	Ld a 1, [w2-4]	Set i0 i1, Shift W	Set, b0	
	Ld ab 7 8, [c1,i0-15]	Set i3, Shift i0 i1, Shift W	MAC,w0, c0 i0	
Wr[c3,i8-i15], 5	Ld b 9, [c1,i16-17]	Shift i0 i1, Set W1	MAC,w1, c0 i1	
	Ld a 2, [w5-8]	Set i0 i1, Shift W	MAC,w2, c0 i2	Sigm0, Wrback
	Ld ab 10 11, [c2,i0-15]	Set i3, Shift i0 i1, Shift W	MAC,w3, c1 i0	Sigm1, Wrback
Wr[c3,i16-i19 - - - -], 6	Ld b 12, [c2,i16-17]	Shift i0 i1, Set W2	MAC,w4, c1 i1	
	NOP	Set i0 i1, Shift W	MAC,w5, c1 i2	

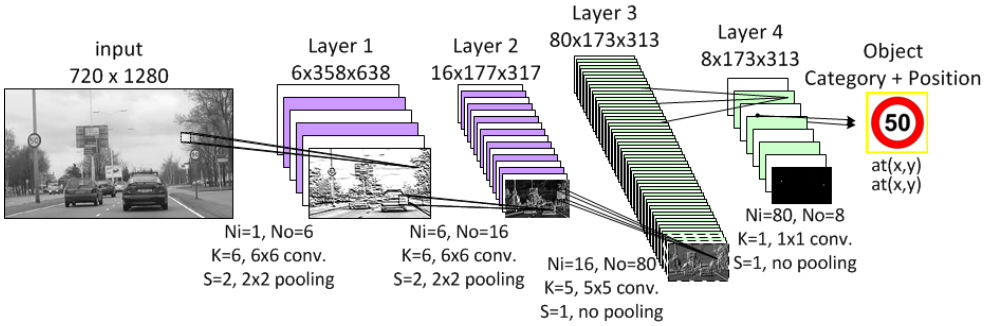
Are You With Me?

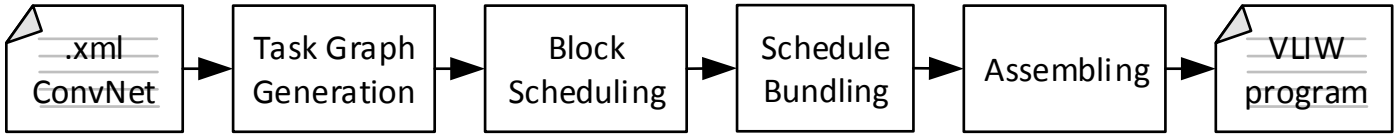
- 3x3 Convolution 20 lines of code
- Neural layer ~400 lines
- Expert programmer
  - 5 hours of coding per layer
  - Impossible for other users



Custom Compiler

- Abstract from the hardware





Write Local	Read Local	Reg Operation	EX VMAC	WB Sgm
10 Cycles	Ld a 0, [b0 w0-1]	Set i3, Shift i0 i1, Shift W	MAC,w6, c2 i0	
	Ld ab 4 5, [c0,i0-15]	Shift i0 i1, Shift W	MAC,w7, c2 i1	
Wr[c3,i0-i7], 4	Ld b 6, [c0,i16-17]	Set W0	MAC,w8, c2 i2	
	Ld a 1, [w2-4]	Set i0 i1, Shift W	Set, b0	
	Ld ab 7 8, [c1,i0-15]	Set i3, Shift i0 i1, Shift W	MAC,w0, c0 i0	
Wr[c3,i8-i15], 5	Ld b 9, [c1,i16-17]	Shift i0 i1, Set W1	MAC,w1, c0 i1	
	Ld a 2, [w5-8]	Set i0 i1, Shift W	MAC,w2, c0 i2	Sigm0, Wrback
	Ld ab 10 11, [c2,i0-15]	Set i3, Shift i0 i1, Shift W	MAC,w3, c1 i0	Sigm1, Wrback
Wr[c3,i16-i19 - - - -], 6	Ld b 12, [c2,i16-17]	Shift i0 i1, Set W2	MAC,w4, c1 i1	
	NOP	Set i0 i1, Shift W	MAC,w5, c1 i2	

What do we gain?

- ~ 20x speedup vs ARM A9
- ~ 1.2x speedup vs embedded GPU
- Ultra-low power 100mW

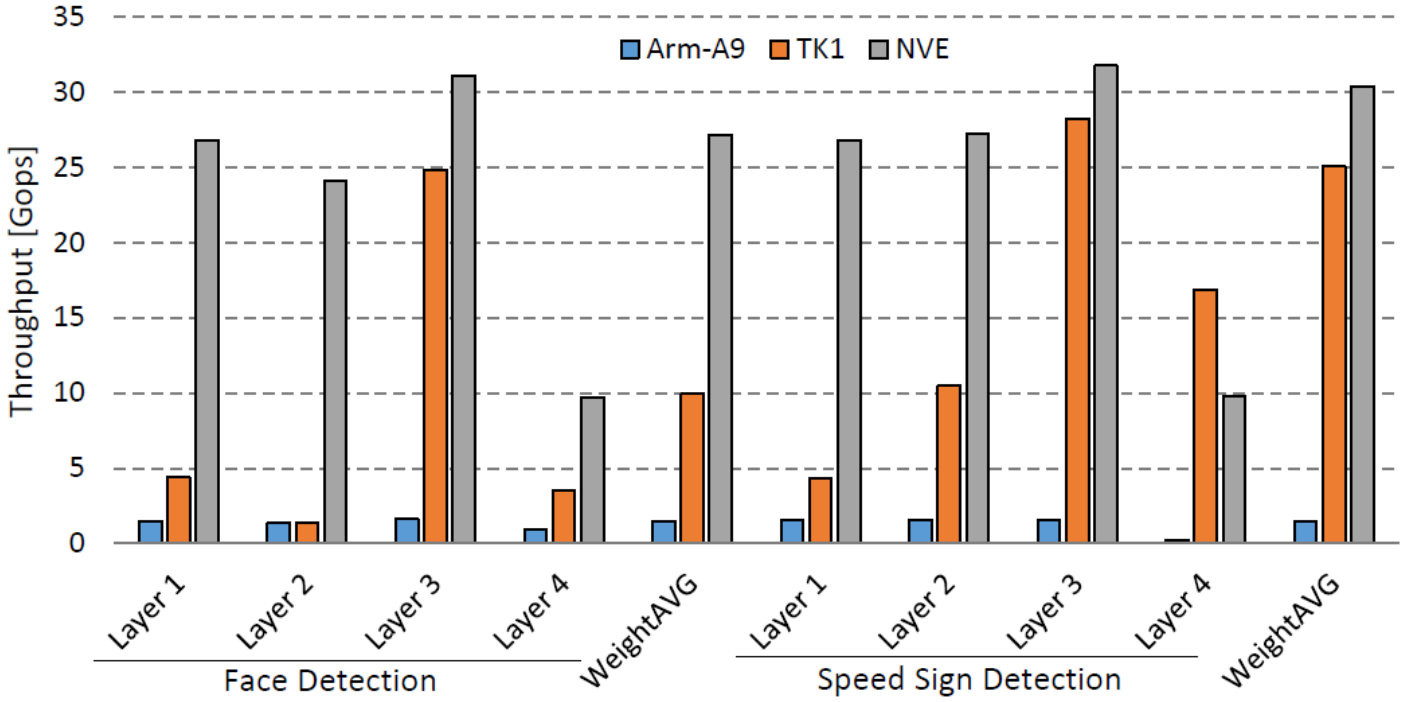
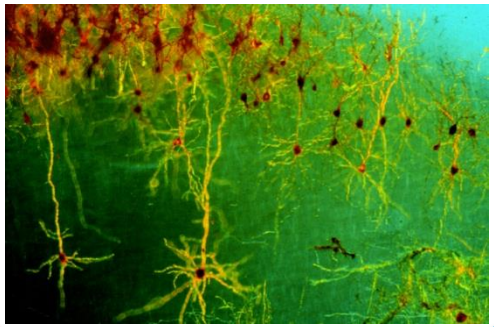


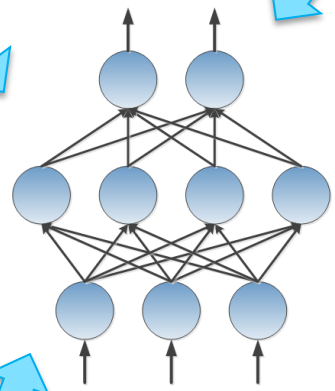
Fig. 8. Throughput comparison Arm-A9, NVidia Jetson TK1, and NVE

Convergence of different domains

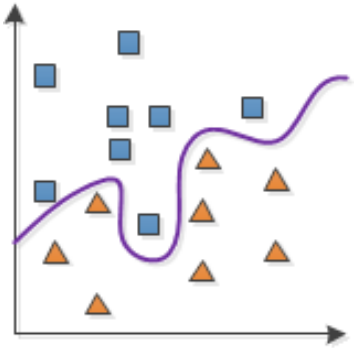
Neurobiology

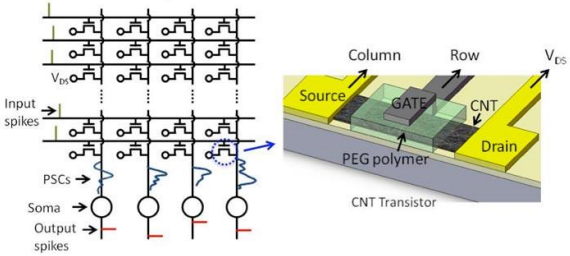


Applications

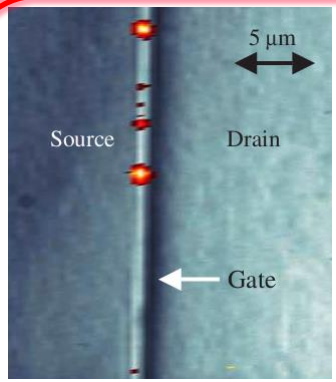


Machine Learning



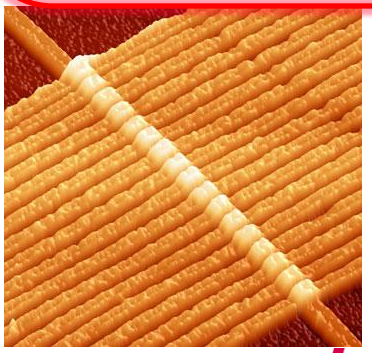


Neuromorphic



Technology

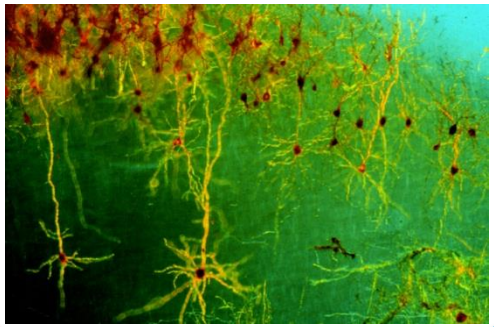
Constraints



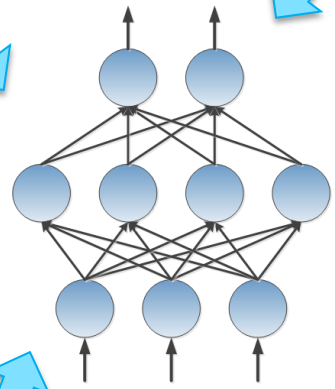
Innovations

Convergence of different domains

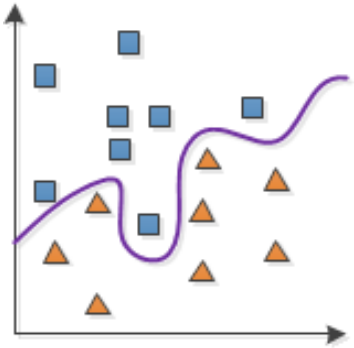
Neurobiology

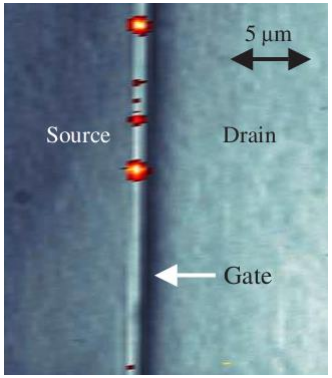


Applications



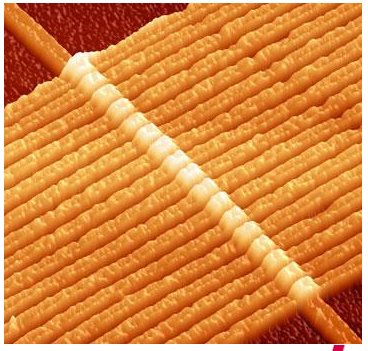
Machine Learning



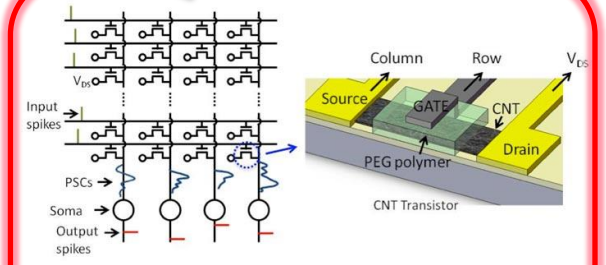


Constraints

Technology



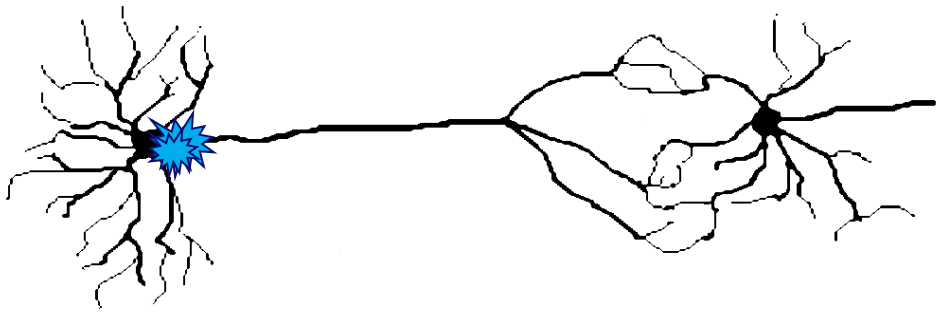
Innovations



Neuromorphic

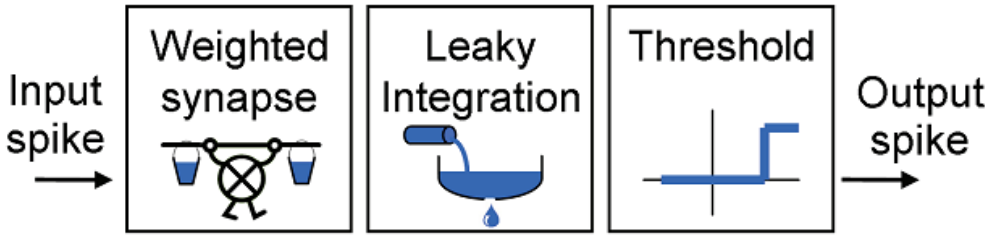
Size

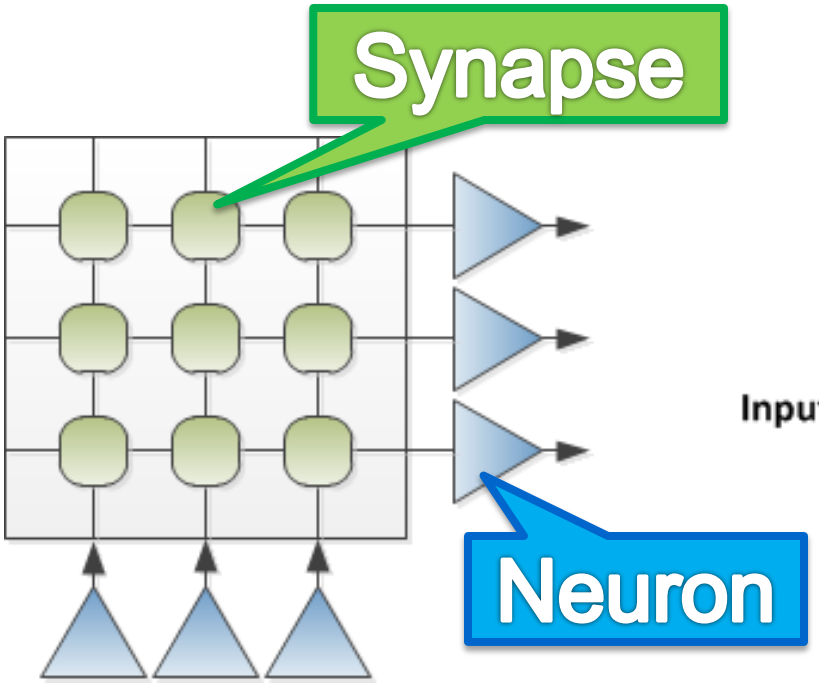
- Digital CMOS
  - Technology available
  - Implementation of useful accelerators
  - Not dense enough for largest bio-inspired networks
- Analog
  - Much more dense implementation
- Recall Biological Neuron



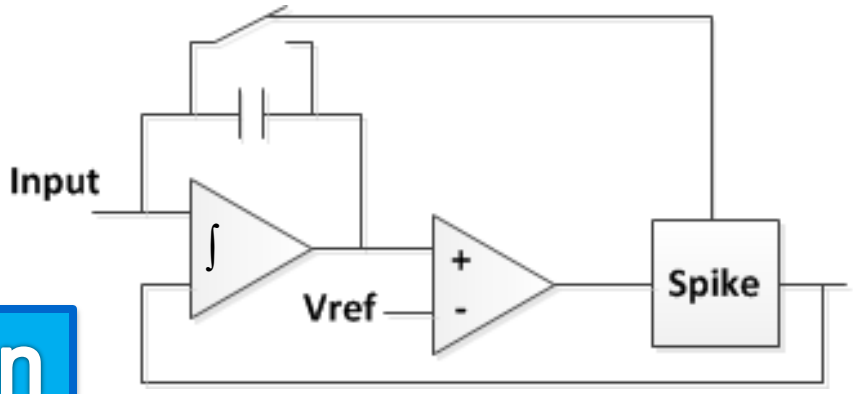
Analog Spiking Neurons

- **Kirchhoff's law**
- **Capacitive integration**
- **Leakage**



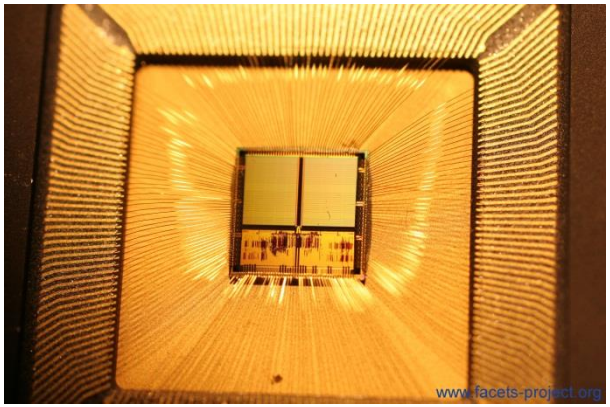


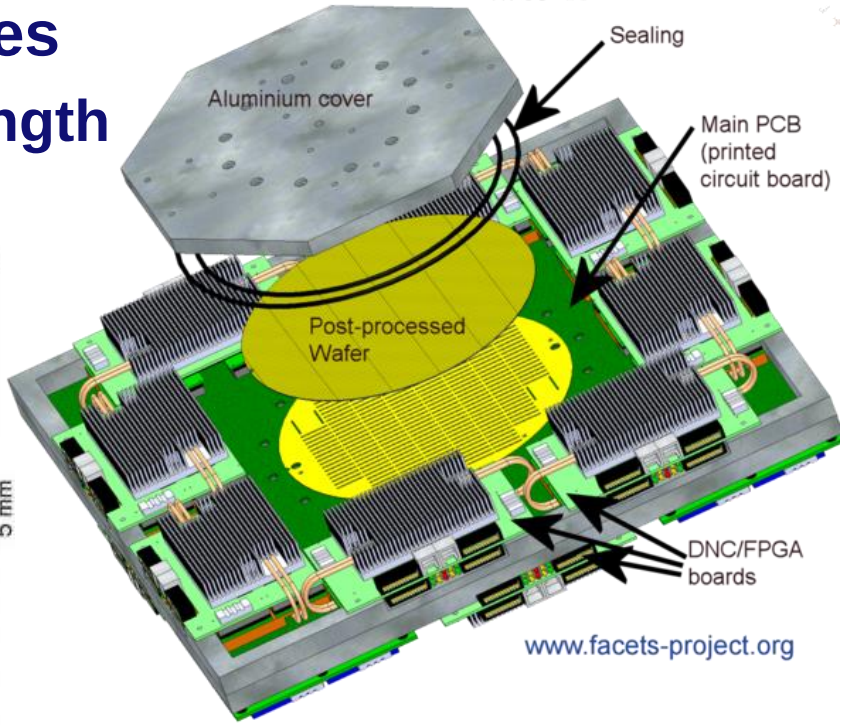
~14 transistors

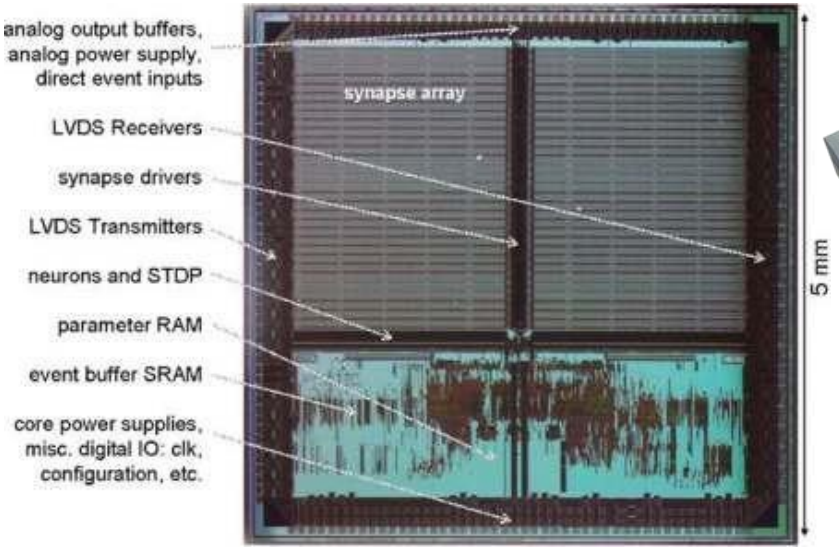


Architecture Facets Project

- Facets
  - Integrate & Fire
  - 250000 neurons wafer
  - 60 million synapses
- Most area used for synapses
  - Storage of connection strength
  - Interconnect 2-D

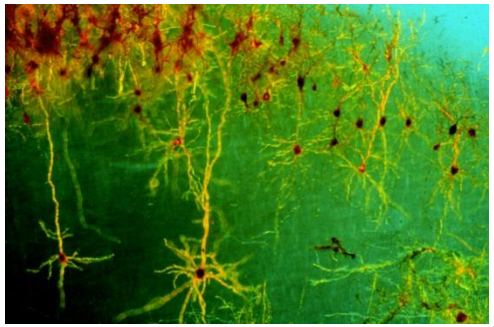




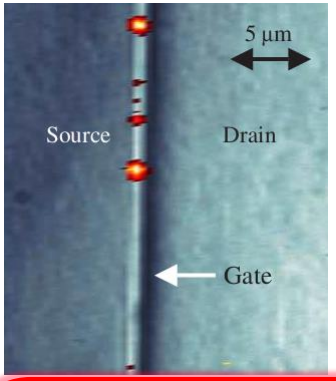


Convergence of different domains

Neurobiology

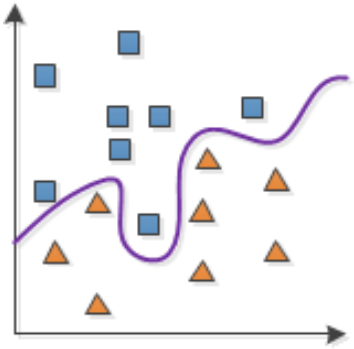


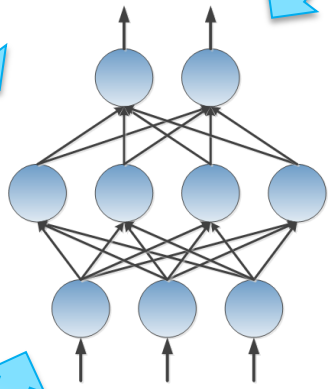
Applications



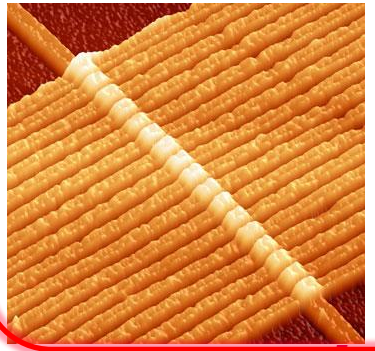
Constraints

Machine Learning

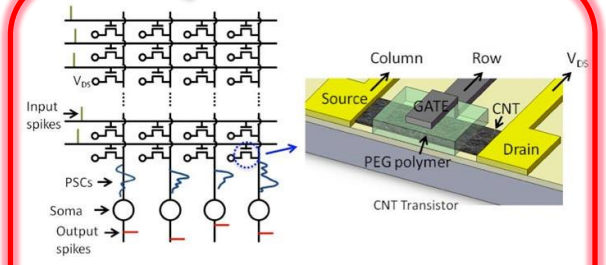




Technology



Innovations



Neuromorphic

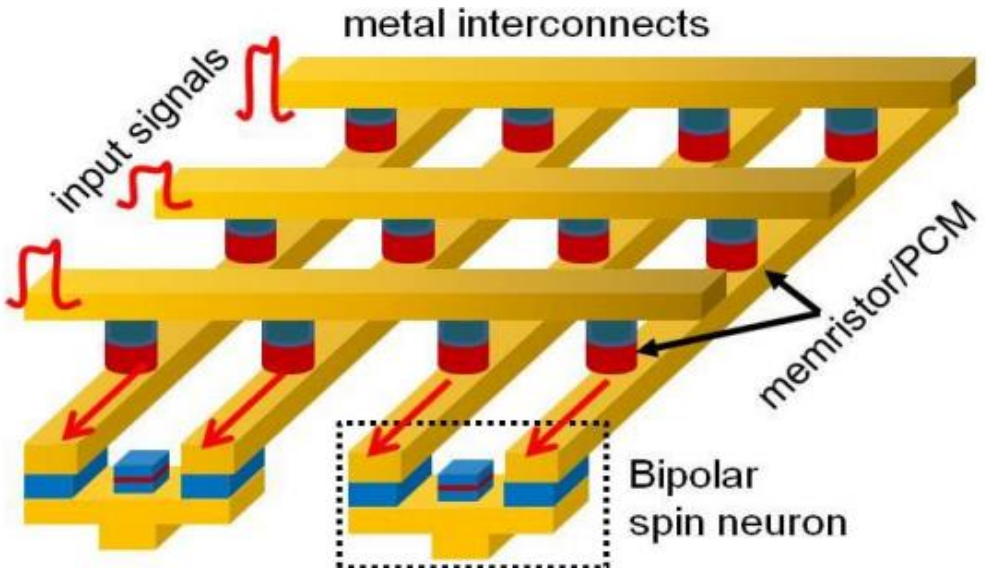
Synapses as Memristors from Intel

Proposal For Neuromorphic Hardware Using Spin Devices

[1]Mrigank Sharad, [2]Charles Augustine, [1]Georgios Panagopoulos, [1]Kaushik Roy
[1]Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
[2]Circuit Research Lab, Intel labs, Intel Corporation, Hillsboro, OR, US
msharad@.purdue.edu

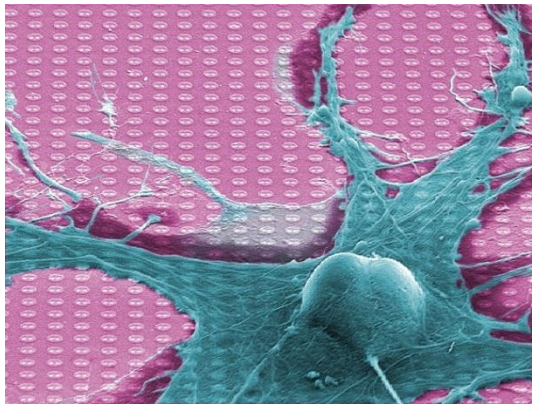
Abstract: We present a design-scheme for ultra-low power Rest of the paper is organized as follows. Section 2 introduces

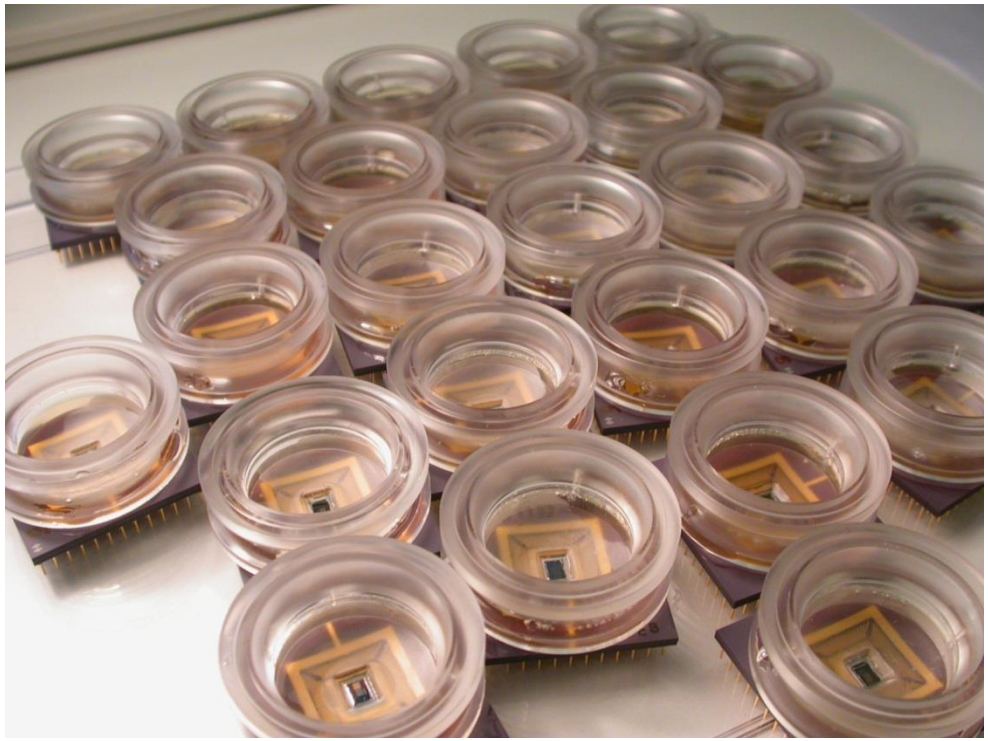
- **Memristor can be used as switch**
- **Also analog storage of memristance**

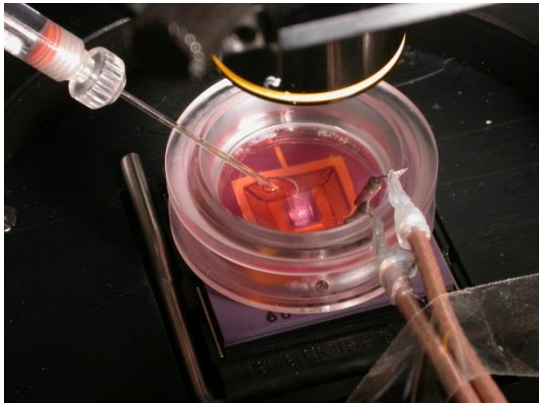


Beyond Silicon

- Infineon NeuroChip
- Directly uses biological networks
- Difficult to connect to other devices

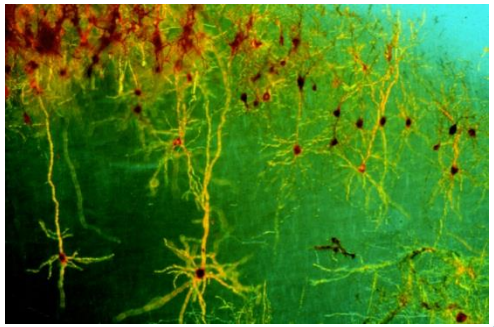




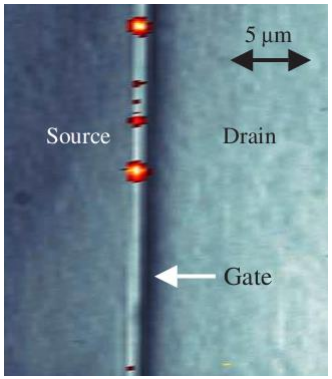


Convergence of different domains

Neurobiology

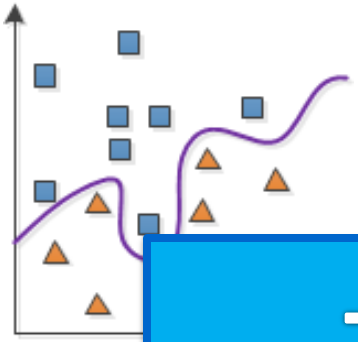


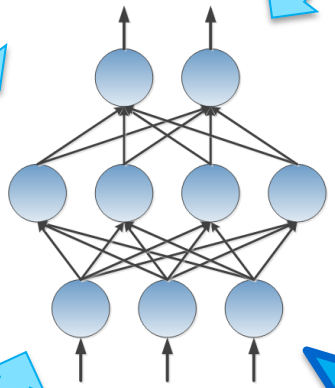
Applications



Constraints

Machine Learning





Technology

ovations

Thank you for your attention

Neuromorphic