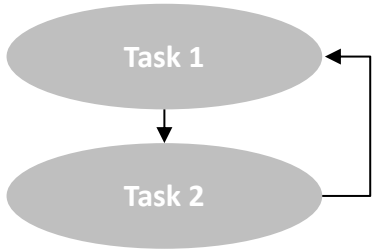
ECE 598NS: Machine Learning in Silicon
Fall 2020

Course Summary & Future Directions

Naresh Shanbhag
Professor
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

DL in Hardware

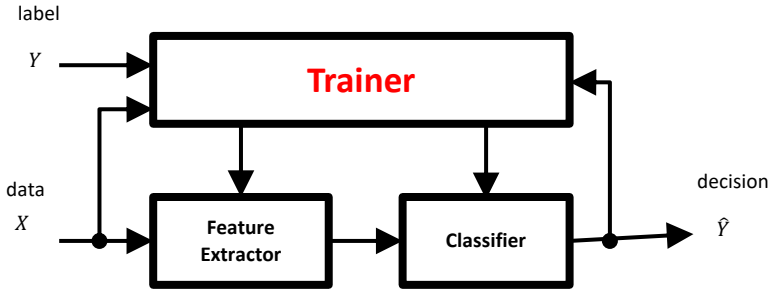
Applications



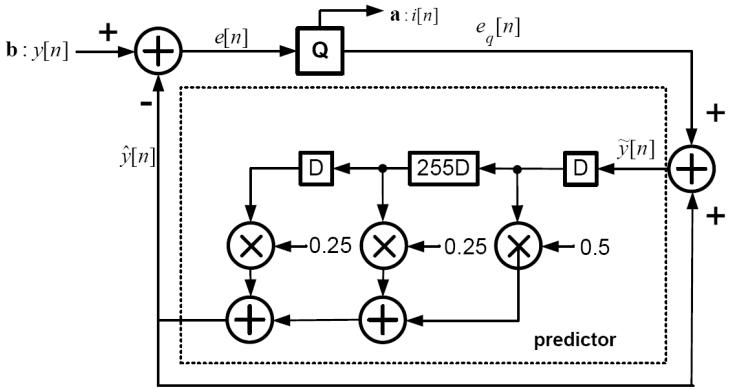
natural vision, EEG analysis, navigation, surveillance

Inference Tasks

Learning Models

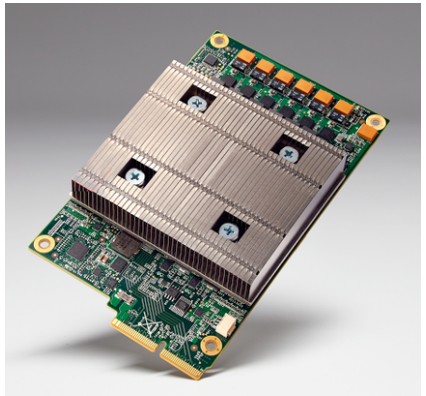


Finite-precision Architecture

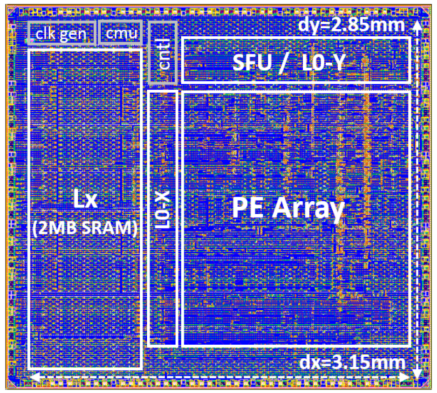


Hardware

module



integrated circuit



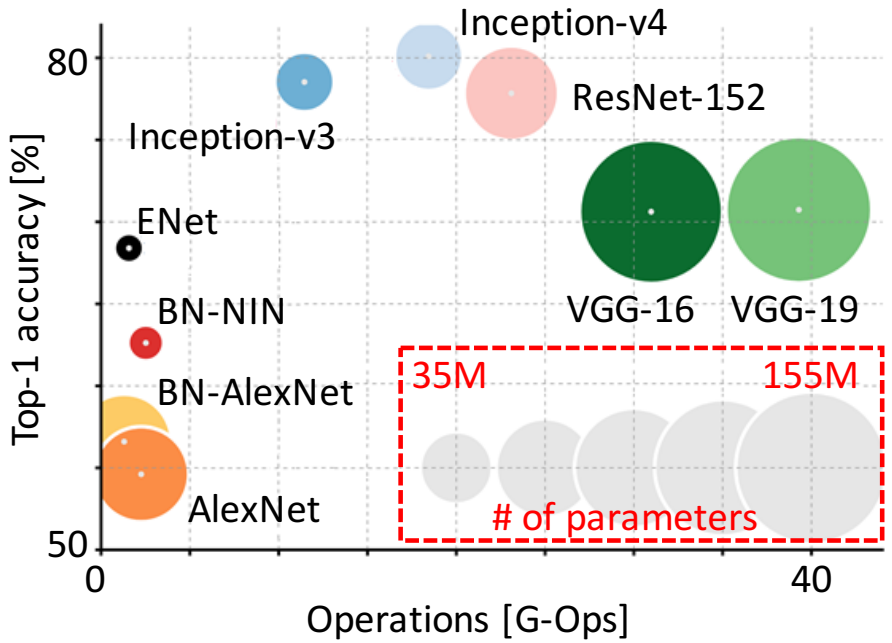
- deep learning basics – LMS, dot-product, fixed-point
- inference in fixed-point
- low-complexity networks – MobileNet, SqueezeNet,..
- DNN training
- model compression via training, binarization, ternarization

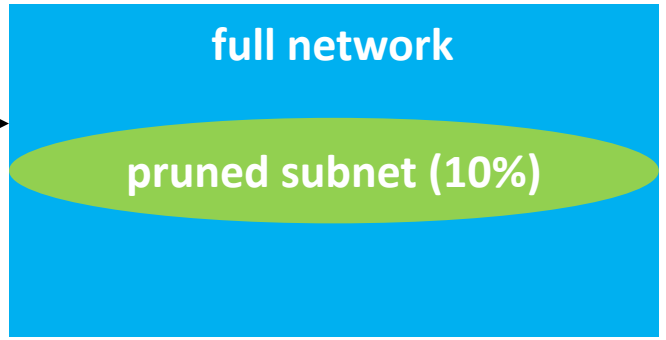
- DNN architectures – roofline & floorline plots
- algorithm-to-architecture mapping, algorithm transforms
- energy-delay trade-offs
- Shannon-inspired statistical error compensation

- in-memory computing
- the deep in-memory architecture & case studies
- near and in-memory architectures via case studies
- Future trends

- taking machine learning to the next level
- emerging architectures and models of computation

Accuracy vs. Complexity Trade-off

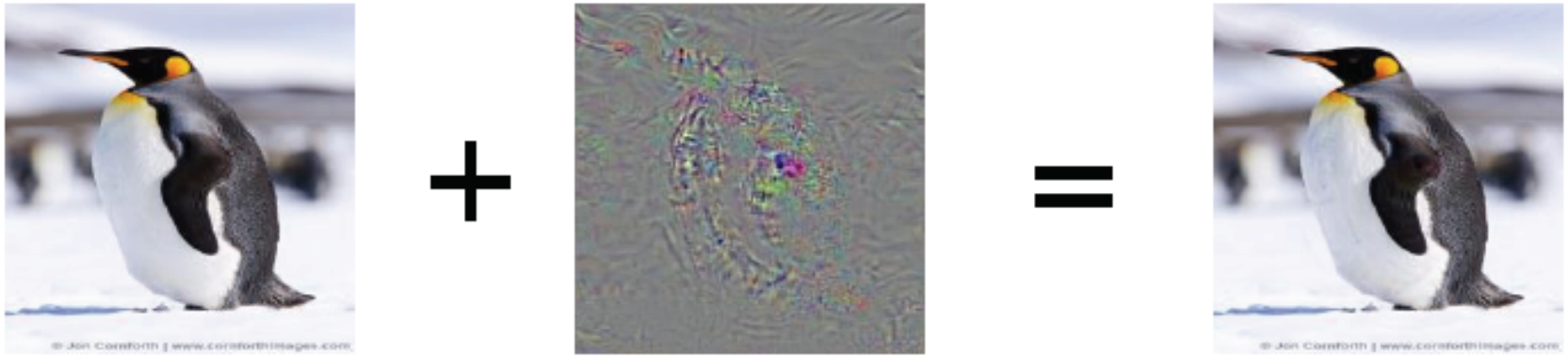




- accuracy generally improves with size of trained model … but
- post-training model compression can reduce model size (> 90%)
- [Frankle, ICLR'19]: the pruned subnet can be trained from scratch to reach full network accuracy with proper initialization
- so what exactly is the accuracy vs. complexity trade-off?

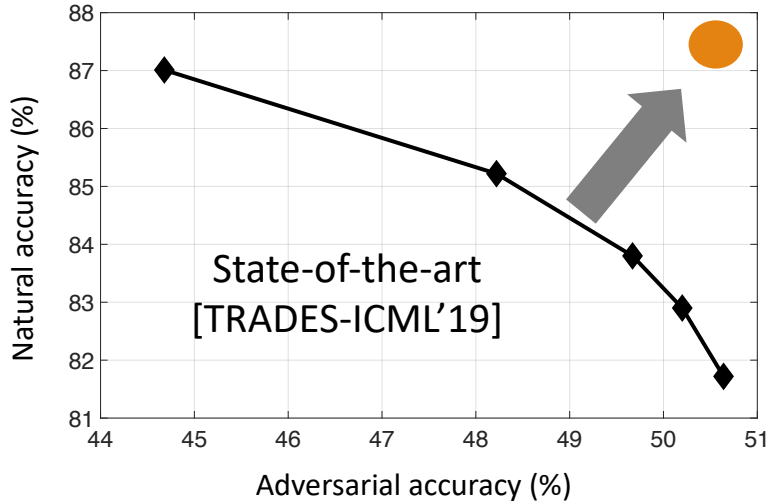
Accuracy vs. Robustness Trade-off

Cause is a mystery & cost of robustness is unclear



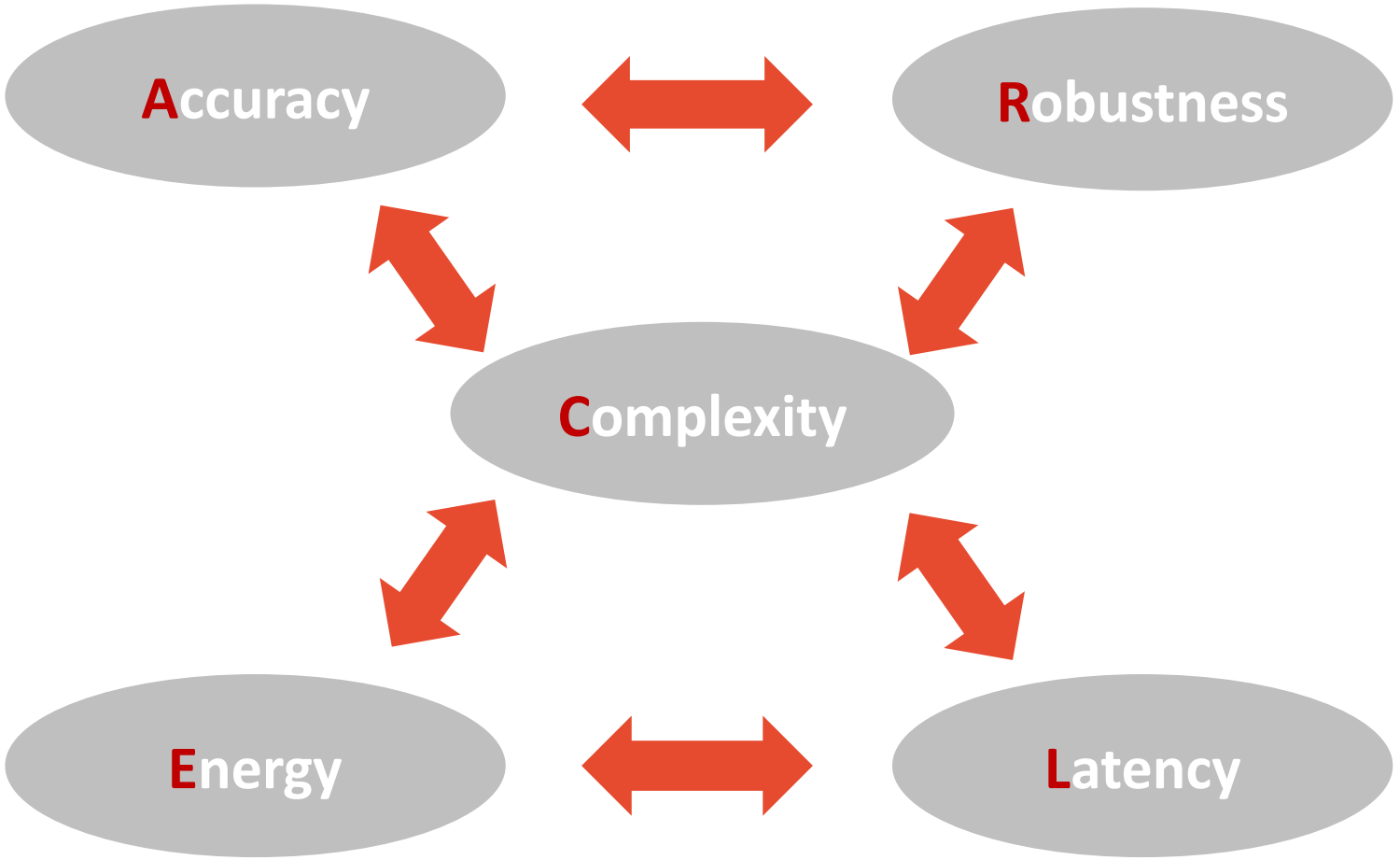
[Xie, ICLR'18]

ResNet-18 on CIFAR-10



- trade-off accuracy vs. robustness
- adversarial training → 10X more expensive than vanilla training
- robustness improves with model capacity

AI Systems Trade-off Space



Hard Challenge 1 – Machine Learning with Guarantees

- robustness, accuracy, complexity, fairness, security, privacy
- need to constrain machines capabilities?
- is there a Shannon-like theory for ML?

Hard Challenge 2 – AI for Social Good

[NeurIPS'19]

Establishing an Evaluation Metric to Quantify Climate Change Image Realism

Sharon Zhou*
Stanford University
sharonz@cs.stanford.edu

Alexandra Luccioni*
Mila, Université de Montréal
luccionis@mila.quebec

Gautier Cosne
Mila, Université de Montréal
cosnegau@mila.quebec

Michael S. Bernstein
Stanford University
msb@cs.stanford.edu

Yoshua Bengio
Mila, Université de Montréal
yoshua.bengio@mila.quebec

- humanistic AI, AI for climate change & income inequality, flow-inducing AI,…..
- engineering needs to engage arts & humanities
- universities need to lead here

- much interest in hardware-efficient supervised machine learning kernels - classifiers/regressors, e.g. DNN

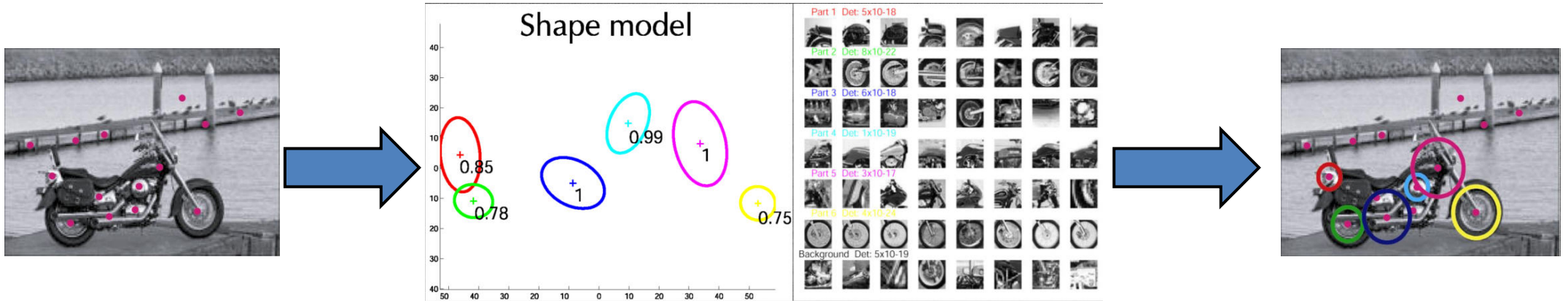
  but ..... these are low-level machine learning tasks

  The next wave of machines need to

- learn unsupervised and persistently
- develop knowledge of its environment with limited data
- learn collaboratively with other machines and humans
- interact with humans empathetically
- be robust, trustworthy and secure

One-Shot Learning

- Learning object categories from just a few examples – Bayesian "one-shot" learning
- Categories represented by probabilistic models
- Incorporate prior knowledge about objects



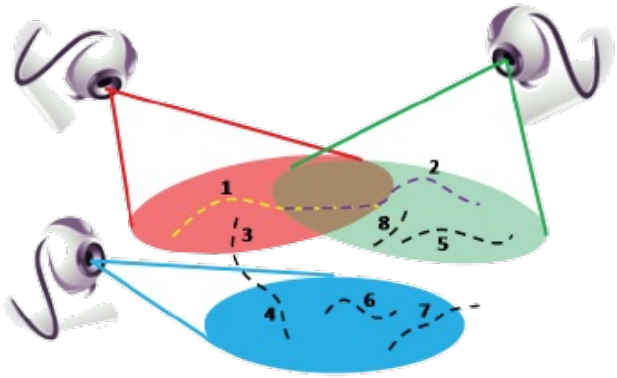
Feature detection *Recognition*

Communication-theoretic in its flavor, but models are learnt from data
(Open issue: how best to blend physical models with data-driven models?)
Bayesian models and factor graphs layered on top of feature-extracting front ends

Li Fei-Fei et al, *One shot learning of object categories,* IEEE PAMI 2006.
Wong and Yuille, *One shot learning via compositions of meaningful patches,* ICCV 2015.

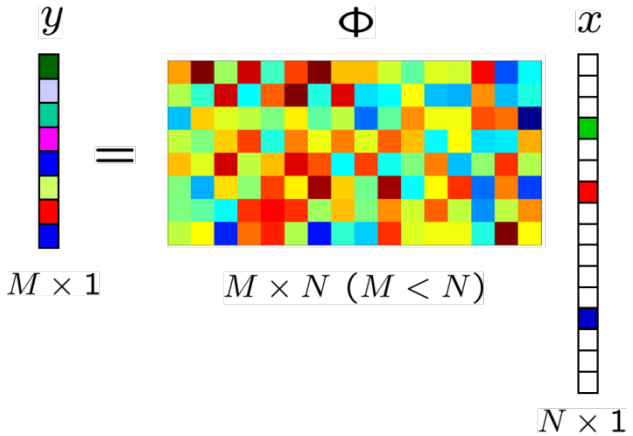
Unsupervised Learning

[Madhow (UCSB)]

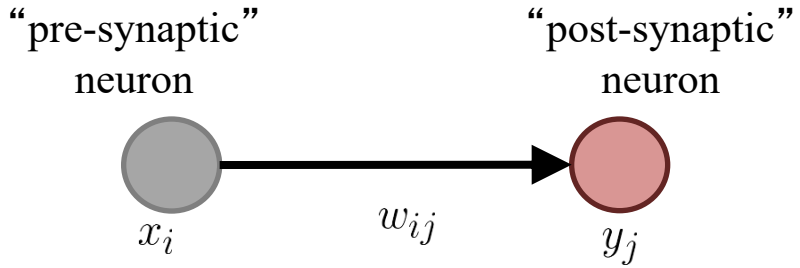


- Unsupervised learning at each sensor ➔ sparse distributed codes
- Bake in hardware constraints
- Various paradigms for sensor fusion: supervised, unsupervised, reinforcement

Two example approaches: Compressive and Hebbian



- Dimension reduction via random projection
- Robust to hardware nonlinearities

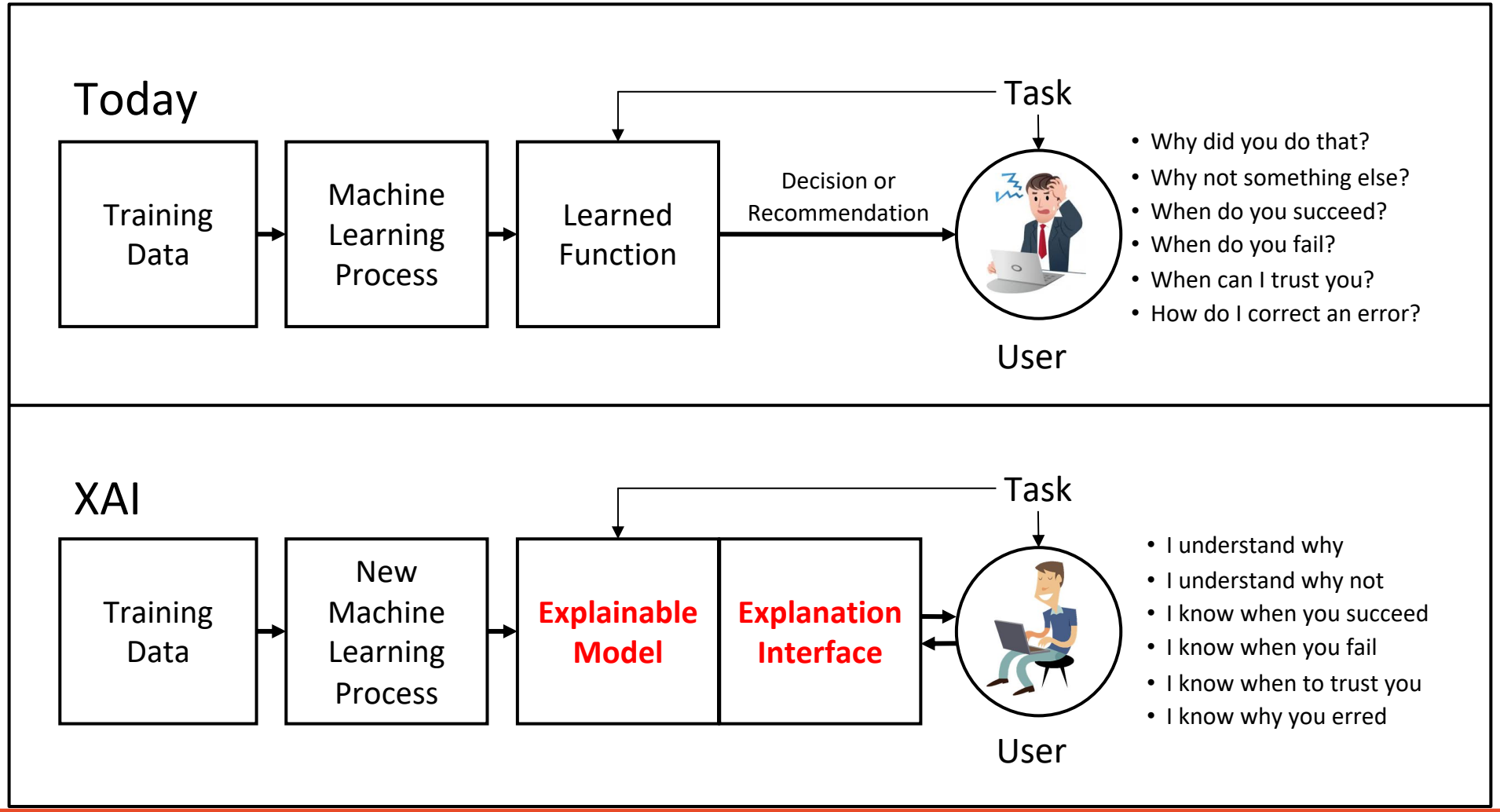


$$\Delta w_{ij} \propto x_i \cdot h(y_j) \qquad y_j = f(\mathbf{x}^T \mathbf{w}_j)$$

- Hebb principle: "fire together, wire together"
- Bottom-up learning paradigm

Interpretable Learning

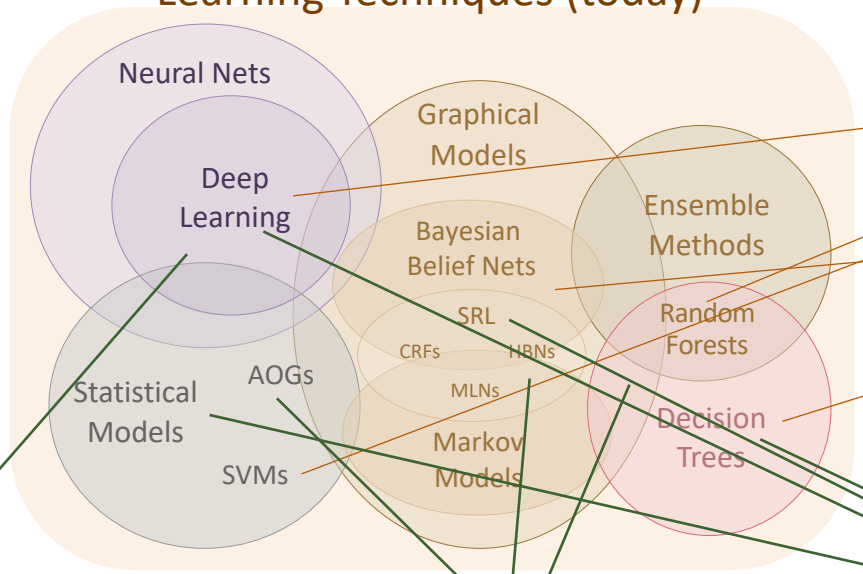
[Gunning, Explainable Artificial Intelligence (XAI)]



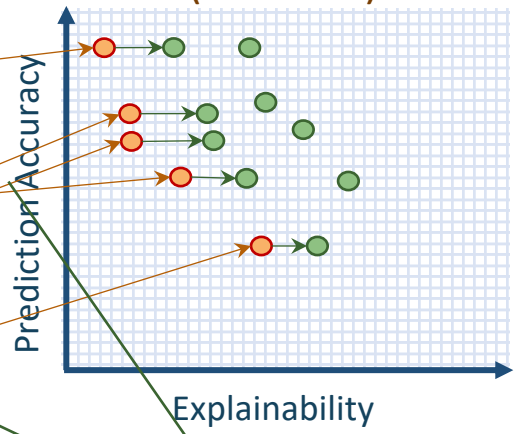
New Approach

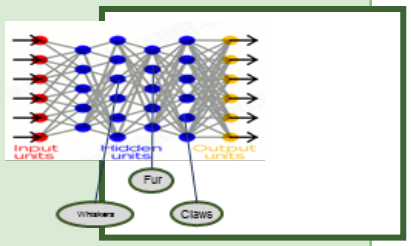
Create a suite of machine learning techniques that produce more explainable models, while maintaining a high level of learning performance

Learning Techniques (today)



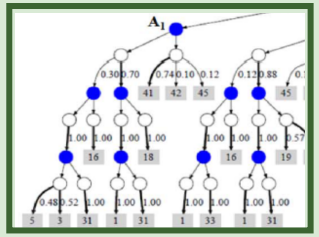
Explainability (notional)





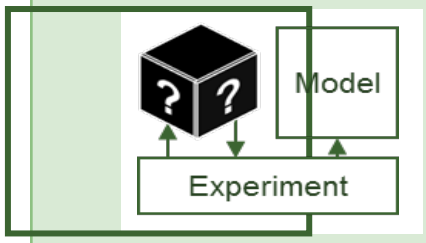
Deep Explanation

Modified deep learning techniques to learn explainable features



Interpretable Models

Techniques to learn more structured, interpretable, causal models

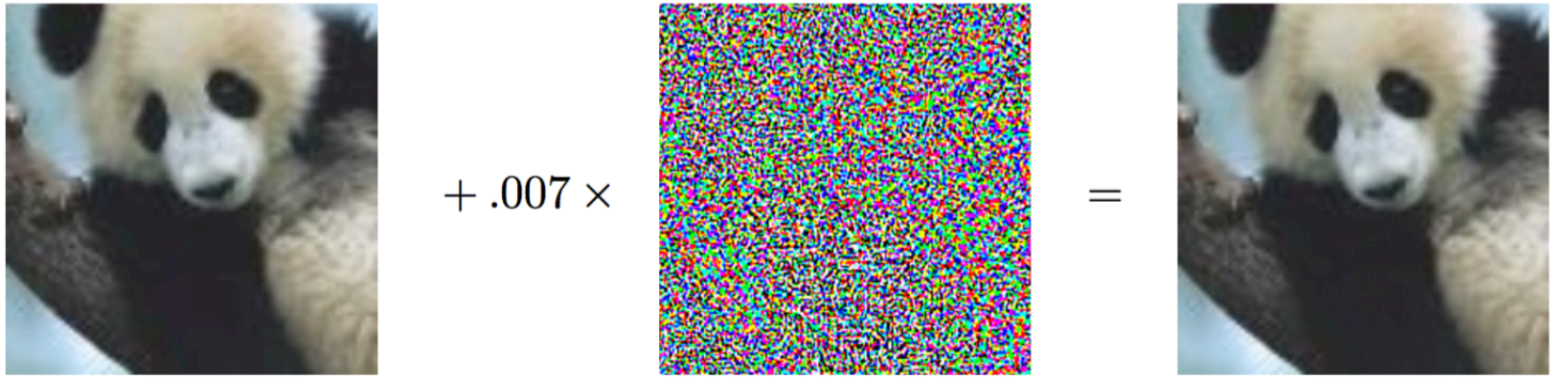


Model Induction

Techniques to infer an explainable model from any model as a black box

[Source: Gunning, Explainable Artificial Intelligence (XAI)]

Adversarially Robust Deep Learning



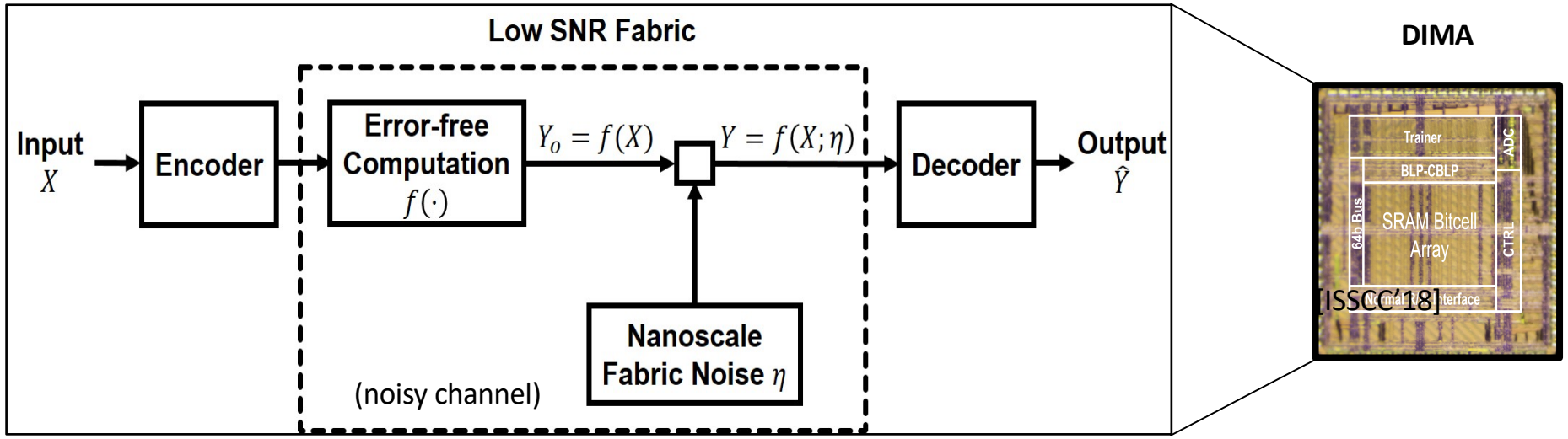
Fundamental problem with deep nets trained using gradient descent
Can fool the net using small perturbations chosen in adversarial directions

Goodfellow, Shlens, Szegedy, *Explaining and harnessing adversarial examples,* ICLR 2015.

- taking machine learning to the next level
- emerging architectures and models of computation

- Shannon-inspired statistical computing
- hyperdimensional computing
- stochastic (bit-stream) computing
- approximate computing
- neuromorphic computing

Shannon-inspired Model of Computing



1. use **information-based metrics** e.g., mutual information $I(Y_o; \hat{Y})$
2. design **low SNR fabrics**, e.g., deep in-memory architecture (DIMA)
3. develop **statistical error-compensation (SEC)** techniques

- Shannon-inspired statistical computing
- hyperdimensional computing
- stochastic (bit-stream) computing
- approximate computing
- neuromorphic computing

Hyperdimensional (HD) Computing

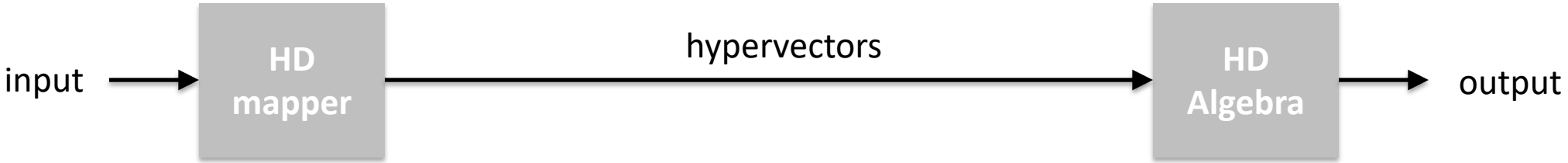
[Kanerva, Hyperdimensional Computing, 2009]

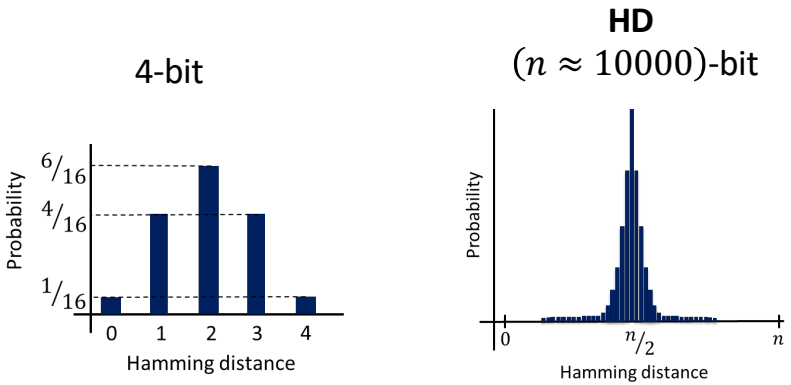
- neuro-inspired signal representation for cognitive applications
- heavily redundant and very high dimensional
- randomized mappings/sampling
- compute with large random patterns
  - manipulate *similarity* of patterns
- intrinsically robust (energy-efficient?)

- related to Shannon's random codes in his 1948 paper

HD Principle

[Rabaey, Olshausen, Mitra, Wong]





- almost all vectors are nearly orthogonal
- requires random mapping
- related to Shannon's random codes

$$\vec{S} = \vec{A} + \vec{B} + \vec{C}$$

($\vec{S}$ is *similar* to $\vec{A}, \vec{B}$ and $\vec{C}$)

$$\vec{S} = \vec{A} * \vec{B} * \vec{C}$$

($\vec{S}$ is *dissimilar* to all $\vec{A}, \vec{B}$ and $\vec{C}$)

- orthogonalization via local **M**ultiply (XOR)
- summarization via local **A**dd (OR)
- ordered summarization via **P**ermutation

Example

- Map 3-b vector $\boldsymbol{b}$ to 10^4-b vector $\boldsymbol{c}$

$$\overset{\textbf{3b}}{[010]} \rightarrow \overset{\textbf{10,000b}}{[100101 \ldots\ldots 11101]}$$

- Hyperdimensional space $\boldsymbol{C}$ has size

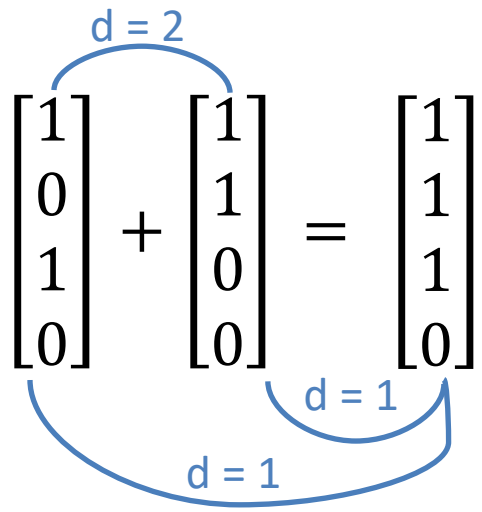
$|\boldsymbol{C}| = 2^{10,000} \approx 10^{3000}$ (huge!)

- Small fraction of vectors in $\boldsymbol{C}$ is needed

$$\vec{S} = \vec{A} + \vec{B} + \vec{C}$$

($\vec{S}$ is *similar* to $\vec{A}, \vec{B}$ and $\vec{C}$)

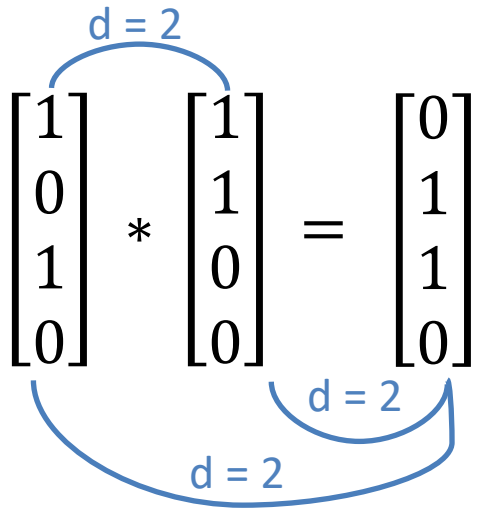
$+ \rightarrow \boldsymbol{OR}$



initial vectors are distant from each other but sum is close to both

$$\vec{S} = \vec{A} * \vec{B} * \vec{C}$$

($\vec{S}$ is *dissimilar* to all $\vec{A}, \vec{B}$ and $\vec{C}$)



$* \rightarrow \boldsymbol{XOR}$

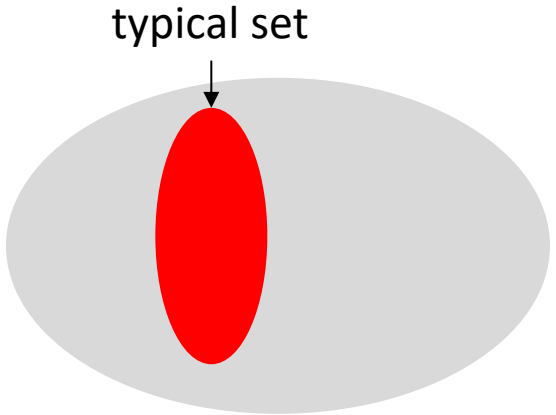
initial vectors are distant from each other and product is distant from both

Asymptotic Equipartition Property (AEP)

probability that a randomly selected Bernoulli sequence of parameter p and of length n has k of n bits as 1

$$P(w(C) = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

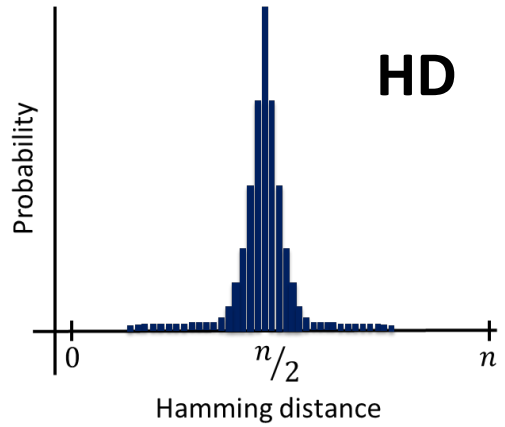
Binomial distribution



Cardinality of the typical set = 2^{nH}

- Asymptotic Equipartition Property (AEP) → a very long sequence of i.i.d RVs can be partitioned into a typical set and an atypical set
- Example of a typical set: a Bernouilli sequence of dimension n will (with high probability) have $\frac{n}{2}$ 1's and $\frac{n}{2}$ 0's as $n \to \infty$

HD and AEP



AEP

$$P(w(C) = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

- HD encoding can be viewed as directly leveraging AEP in hardware
- $d(A, B) = w(C) = \frac{n}{2}$ with very high probability

Hamming distance between two random hypervectors (HD)

Hamming weight of a randomly selected HD vector (AEP)

Implications

- A *randomly sampled* $\boldsymbol{c}$ will result in a vector with a normalized weight of 0.5 with probability approaching 1 as $N \to \infty$ **(hyperdimensional)**

- With a very high probability the vector $\boldsymbol{c}$ will have 5000 '1's if $N = 10000$.
- In this sense vectors $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are 'similar'
- It is also true that vectors $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are very 'dissimilar' (unrelated) because $D(\boldsymbol{c}_1, \boldsymbol{c}_2) = 5000!$

A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing

Abbas Rahimi
EECS Department
University of California
Berkeley
abbas@eecs.berkeley.edu

Pentti Kanerva
Redwood Center for
Theoretical Neuroscience
University of California
Berkeley
pkanerva@berkeley.edu

Jan M. Rabaey
EECS Department
University of California
Berkeley
jan@eecs.berkeley.edu

[ISLPED'16]

- language detection using a HD classifier
- energy efficiency due to memory centricity, extremely low precision (binary), mostly logic operations

HD and Shannon

- Shannon [1948] used random codes to prove capacity theorem → not meant to be used in practice due to limited channel bandwidth → led to search for structured codes with efficient decoders, e.g., algebraic, convolutional, concatenated codes, LDPC, turbo….

- In contrast, HD uses random codes for but for computation.

- **Q:** Is there an equivalent structured version of HD that will be computationally efficient?

- Shannon-inspired statistical computing
- hyperdimensional computing
- stochastic (bit-stream) computing
- approximate computing
- neuromorphic computing

Stochastic (bit stream) Computing

- [Gaines, 1967; Hayes, Reidel]
- employs a bit stream of length N with ones density p, where p represents an analog data value

- simplifies arithmetic units: multiplier → AND gate
- requires long sequences → large values of N to achieve the same $SQNR$

- bit stream can be generated via an oversampling (Sigma-Delta) converter output
- input may be originally in digital word format

Minimum N for Desired SQNR

- If 2's complement needs B bits to achieve a desired $SQNR$ then stochastic computing needs sequence length N:

$$SQNR = 7.76 - PAR + 10\log_{10}(N) = 6B + 1.76 - PAR$$

$$\downarrow$$

$$N = 10^{6(B-1)/10}$$

4b 2's complement data has same SQNR as 63b stochastic data

B	N
1	1
2	3.98
3	15.8
4	63
5	251.2

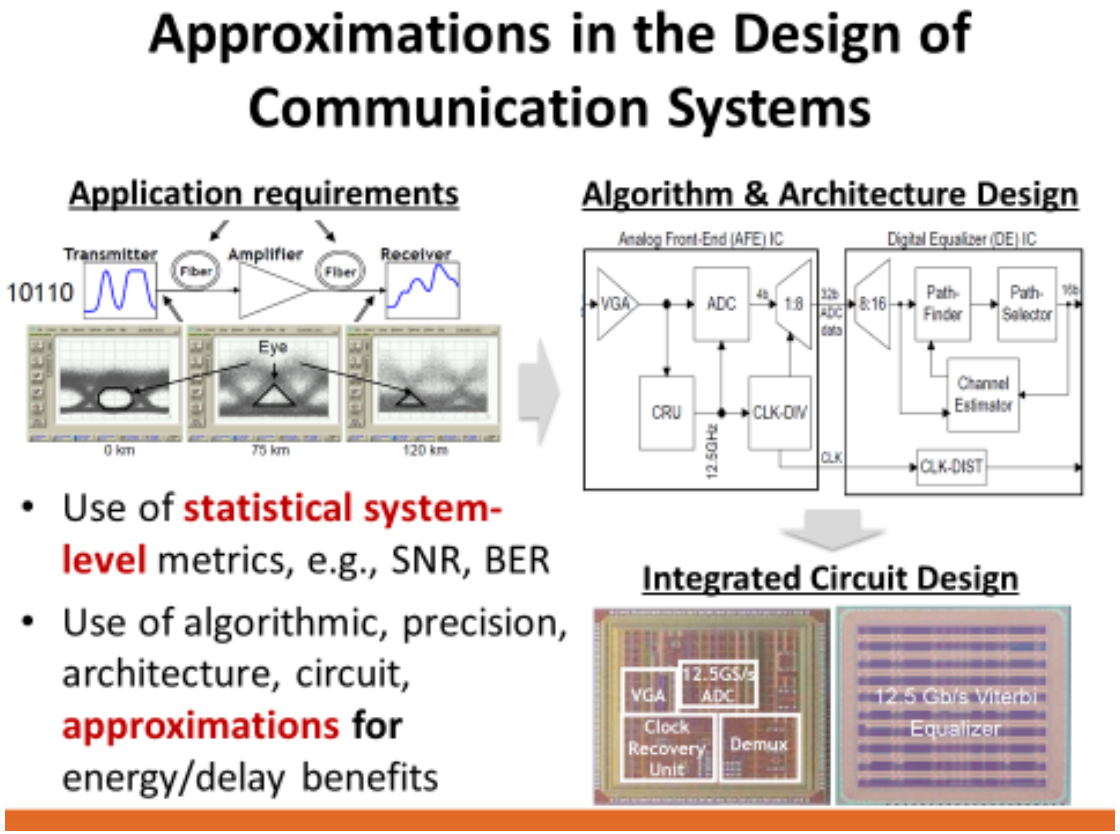
- Required sequence length grows ***exponentially*** with required precision

Comments

- Gaines-style stochastic computing is a number representation and its associated arithmetic unit → this is not a new model of computation
- each number representation, e.g., 2's complement, has its own unique arithmetic unit architecture
- stochastic computing's arithmetic units are simple (Boolean gates)
- but the complexity of the number representation is exponential in the required bit accuracy → questionable energy benefits
- highly redundant number representation → robust to bit errors

- Shannon-inspired statistical computing
- hyperdimensional computing
- stochastic (bit-stream) computing
- approximate computing
- neuromorphic computing

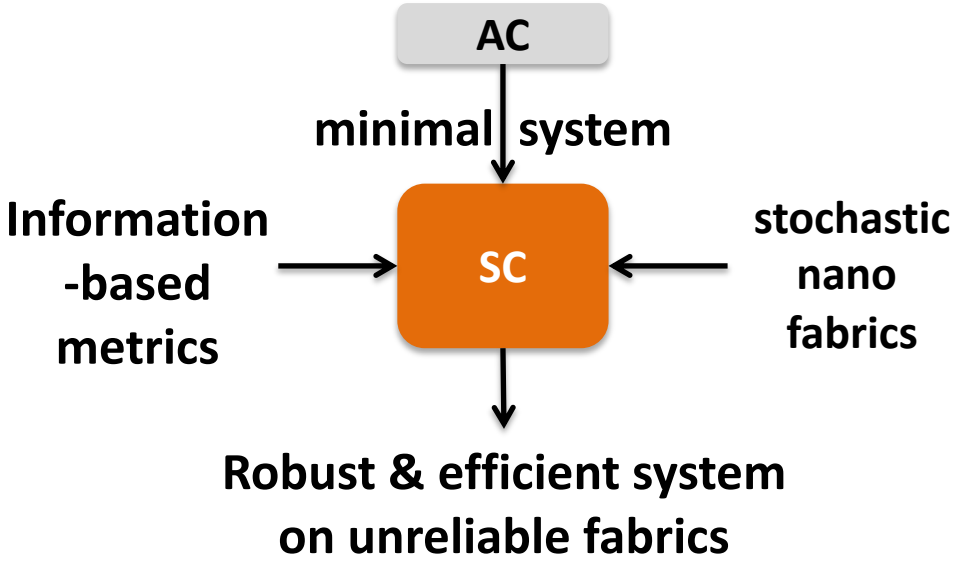
Approximate Computing



- approximate computing (AC) **systematizes** the process of applying approximations
- AC **removes redundancy** to enhance efficiency (energy/throughput)
- AC **operating at its limits** is fragile to errors in the nanoscale fabric (**stochastic fabrics**)

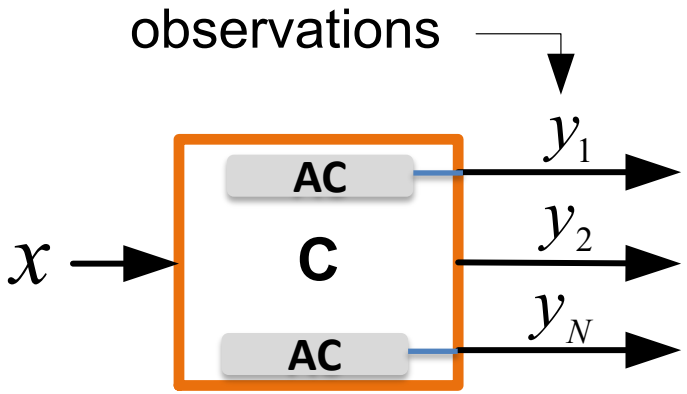
Relating AC to Shannon-inspired Statistical Computing (SC)

use AC to initialize SC



AC → source coder
SC → channel coder

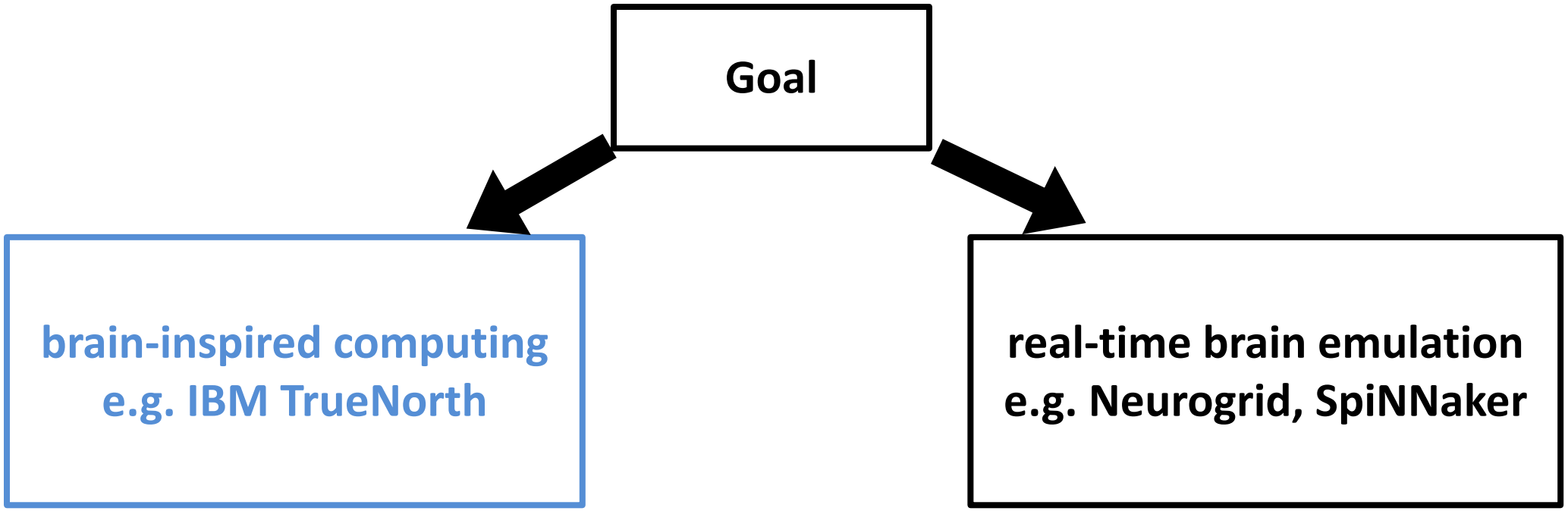
use AC to enhance diversity gain in SC



- Shannon-inspired statistical computing
- hyperdimensional computing
- stochastic (bit-stream) computing
- approximate computing
- neuromorphic computing

Neuromorphic Computing

- mimicking brain's biological neural structures using VLSI components
- first introduced by C. Mead in 1989



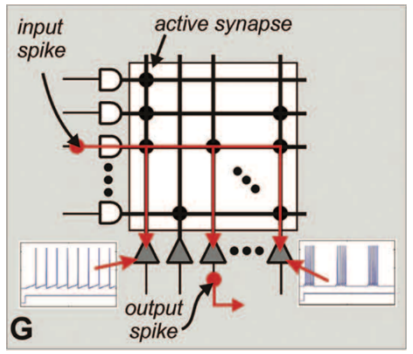
- Different approaches:
  - spiking neural network implementations
    - fully-digital implementations, e.g. IBM TrueNorth
    - analog or mixed-signal implementations
  - employing beyond-CMOS devices e.g. RRAM, spintronics

- ***Underlying question***: will VLSI implementations mimicking neuronal circuits lead to the energy-efficiency of the brain for inference tasks?
- **Key question:** Do we understand the principles of information processing in brain?

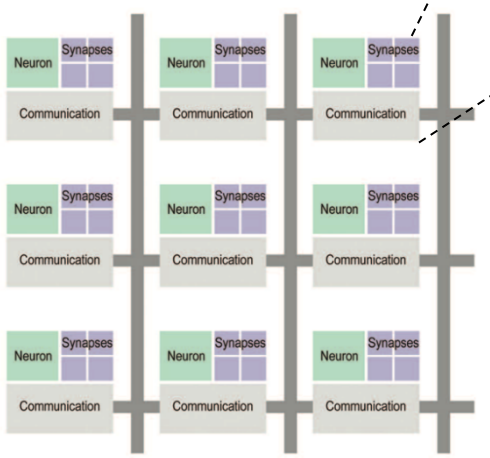
IBM True North

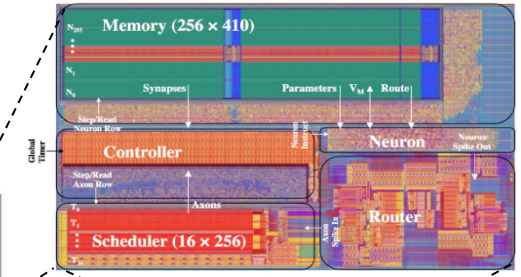
Gigantic (highest number of transistors), fully-digital, distributed off-chip learning

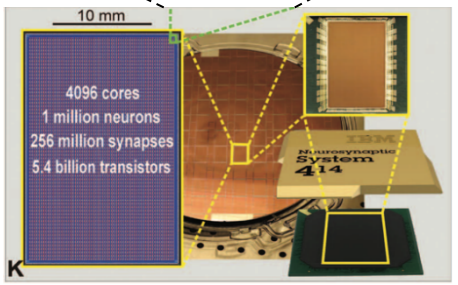
- Each core (tile) implements 256×256 neurons
- fully digital, clocked (1kHz) implementation of neuron dynamics
- Co-located memory and computation in each tile
- asynchronous intercore communication



Neurosynaptic core
(atomic processing tile)







[P. Merolla et al., Science, 2014]

Analog Realizations

SYNAPSE & LEAK INTEGRATION

$$V_j(t) = V_j(t-1) - \lambda_j + \sum_{i=0}^{255} A_i(t) \times w_{i,j} \times S_j^{G_i}$$

THRESHOLD, FIRE, RESET

if $V_j(t) \geq \alpha_j$

Spike

$V_j(t) = R_j$

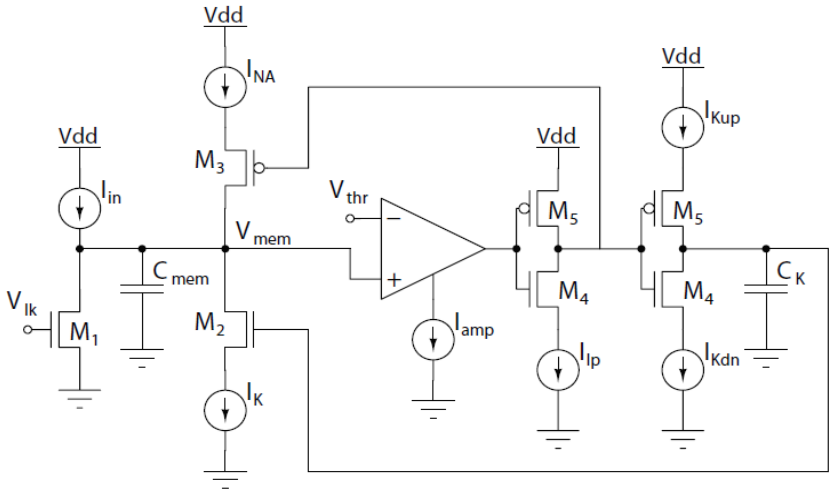
endif

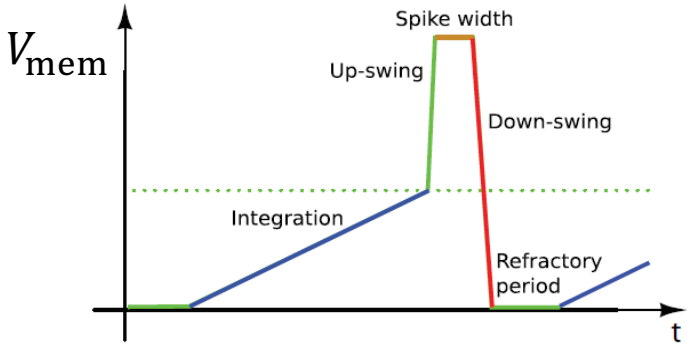
- exploit device properties to realize efficient synapses and neurons
- synapse → variable resistance/weighting operation
- neuron → a form of thresholding
- addition of weighted inputs via current or charge summing
- devices used – CMOS (traditional) and beyond CMOS (spin, RRAM/memristors)

CMOS based Neurons

Simple integrate and fire neuron using CMOS transistors

- V_{mem} denotes the neuron potential
- If $V_{\text{mem}} > V_{\text{thr}}$, spike generation
  - M_3 switches ON, spike begins
  - V_{mem} is increases rapidly due to charging on C_{mem} via I_{NA}
  - C_K is also simultaneously charged via I_{kup}
  - Once voltage across C_K becomes high enough, M_2 switches ON
  - C_{mem} discharges via I_K, V_{mem} reduces & spike ends

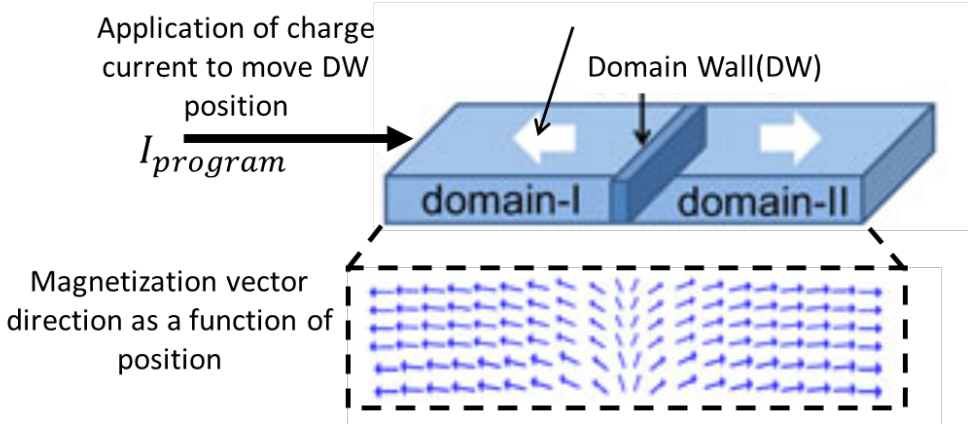




[G. Indiveri et al., Front. in NeuroSci, 2011]

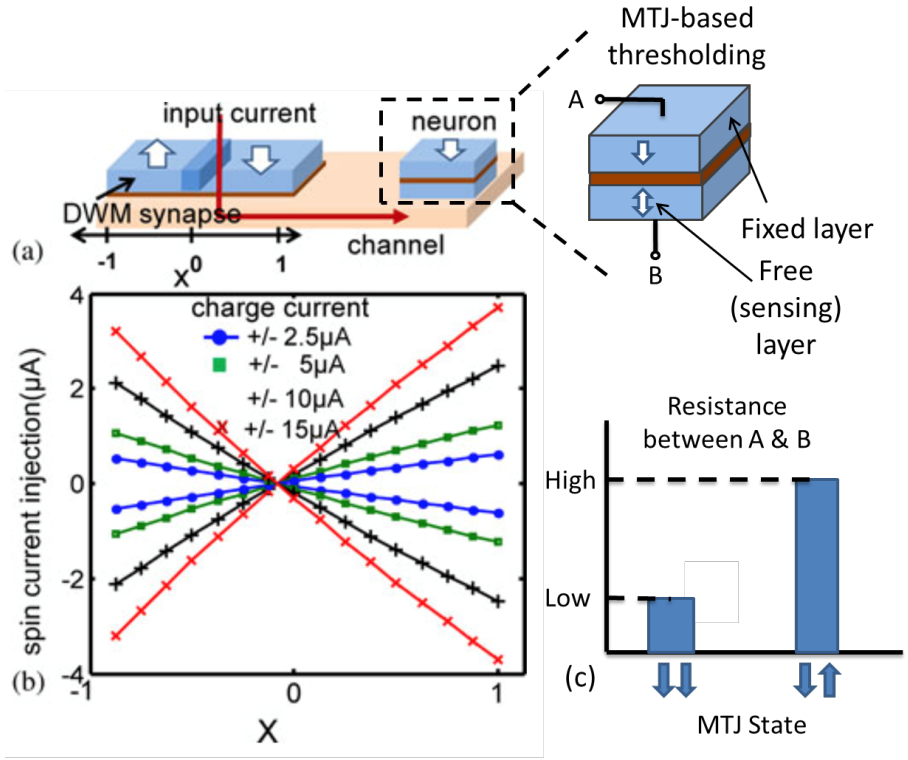
Spin-based Neurmorphic Computing

Spin-based synapse:



- employing *domain wall (DW)* magnets as *variable resistors* to weigh the spin current
- location of (DW) decides the magnitude of spin current injected
- DW position can be changed by $I_{program}$

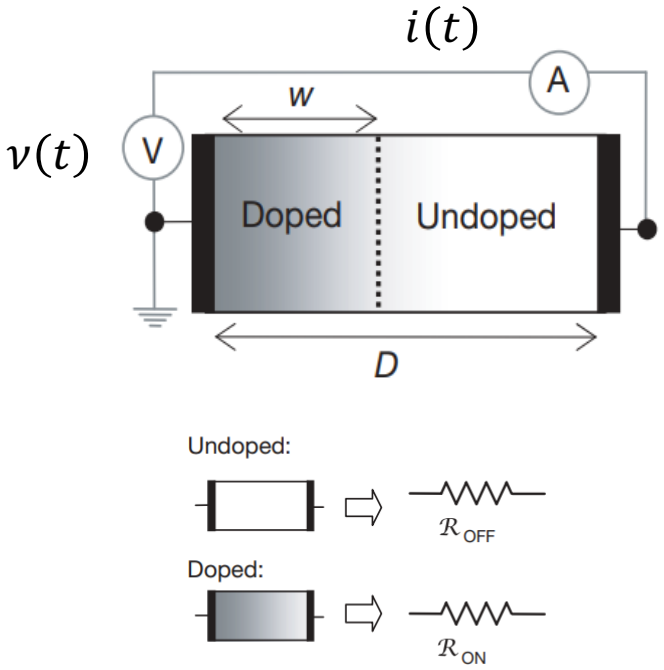
Magnetic tunnel junction (MTJ)-based thresholding neuron:



[M. Sharad, IEEE Trans. Nano., 2012]

Memristor-based Neurmorphic Computing

Memristor as a variable resistor

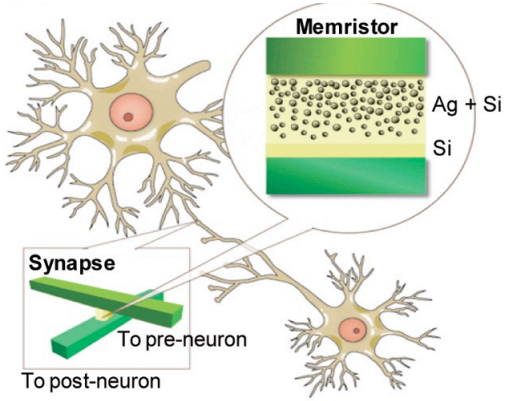


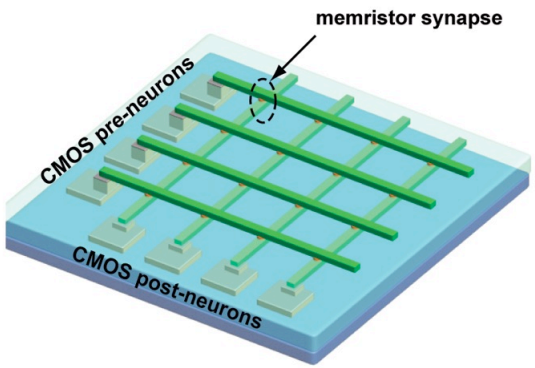
Crossbar of two-terminal memristors

- Memristor resistance depends upon the location of *conduction front*

$$v(t) = \left(\mathcal{R}_{ON}\frac{w(t)}{D} + \mathcal{R}_{OFF}\left(1 - \frac{w(t)}{D}\right)\right) i(t)$$

- Location of conduction front can be altered by passing sufficiently large current



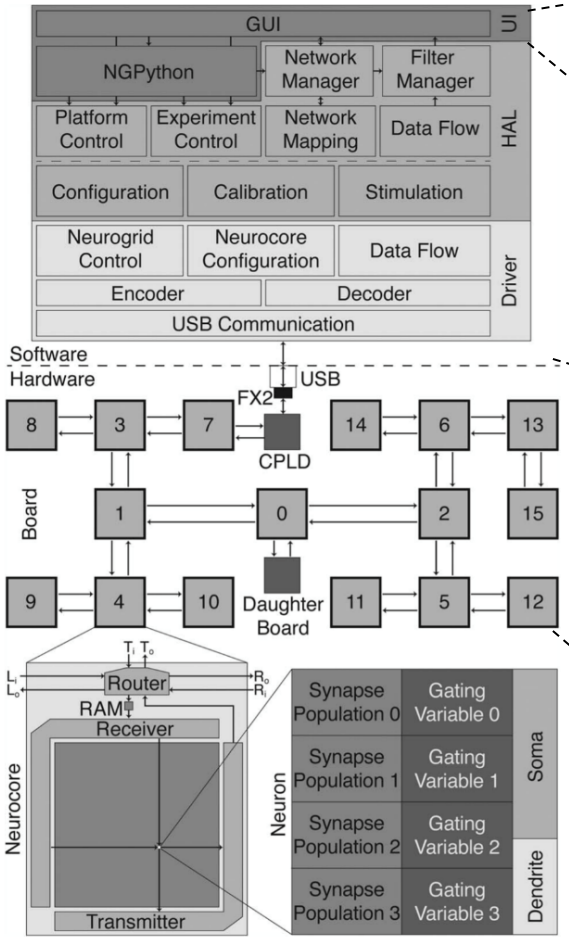


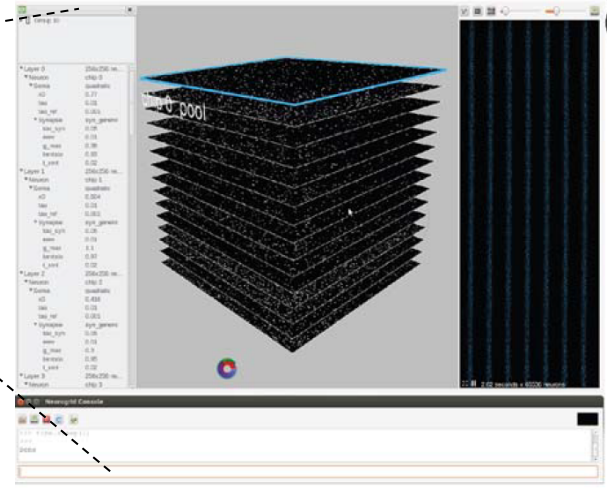
[D. Strukov et al, Nature, 2008] [S. H. Jo et al, Nano Letters, 2010]

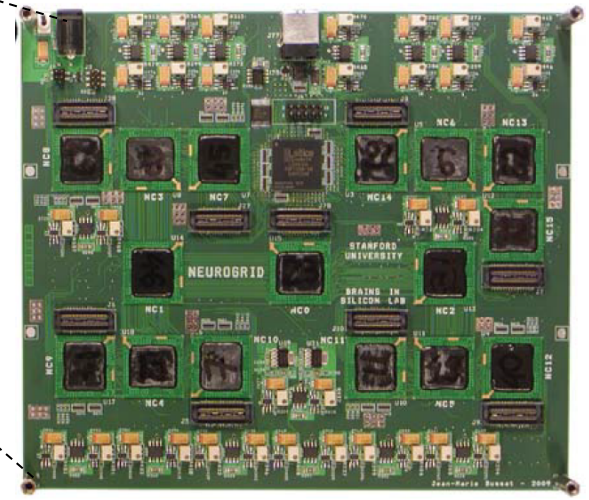
Neurogrid

large scale, multi-chip mixed-signal neural activity simulator

silicon-to-software stack:







GUI:
Allows one to change neuron model parameters, view and plot neural activity

Board:
Consists of 16 chips ("neurocores")
Each neurocore implements 256×256 neurons

[B. Benjamin, IEEE Proc.- 2014]

Summary

- AI is here to stay – our society is being transformed like never before
- Machine learning (ML) – a element of AI systems
- Deep learning (DL) is an ML method showing record performance on many inference tasks
- But DL is very expensive in energy and latency which is limiting its deployment
- ECE 498NSU/598NSG – we studied how to DL systems work and how to minimize their energy-latency product
- Future – more power algorithms will demand more aggressive methods for reducing decision EDP without compromising accuracy, robustness, interpretability, privacy and other metrics

Course Web Page

https://courses.grainger.illinois.edu/ece598nsg/fa2020/
https://courses.grainger.illinois.edu/ece498nsu/fa2020/

http://shanbhag.ece.uiuc.edu