Lecture

1

ECE216B

Introduction to AI/ML Hardware

Prof. Dejan Marković

ee216b@gmail.com

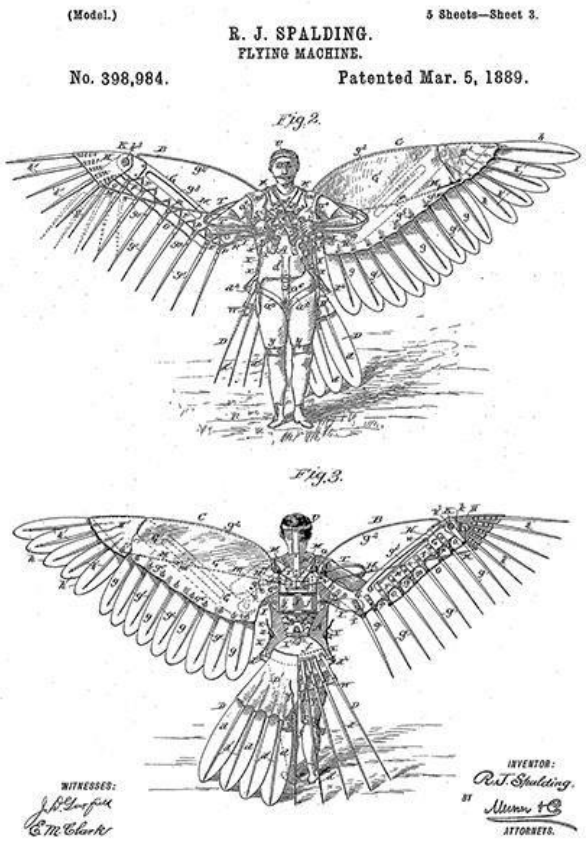
Class Presentations next Tue (5/21/24)

Modules / Week 6		Campus students (up to 5 students / paper)			
Paper	**Name**	**Name**	**Name**	**Name**	**Name**
24S-R1	Umair Siddique				
24S-R2	Dennis Chiu	Shengyi Wei	Yanan Li	Yang-Ho Wu	Wesley Weng
24S-R3	Keith Chen	Cennet Tugce Turan			
24S-R4	Egor Glukhov	Qinghua Gu	Jiyu Zhou	Jinyuan Piao	Hugh Lin
24S-R5a+R5b	Selasi Etchey	Rafael Guerrafuentes	Revati Kulkarni	Frank Sheng	

Aim for about 5 minutes per person

The Quest for Artificial Intelligence

Biologically Inspired Flying Machines?


(Model.) 5 Sheets—Sheet 3.

R. J. SPALDING.

FLYING MACHINE.

No. 398,984. Patented Mar. 5, 1889.

Fig. 2.

Fig. 3.

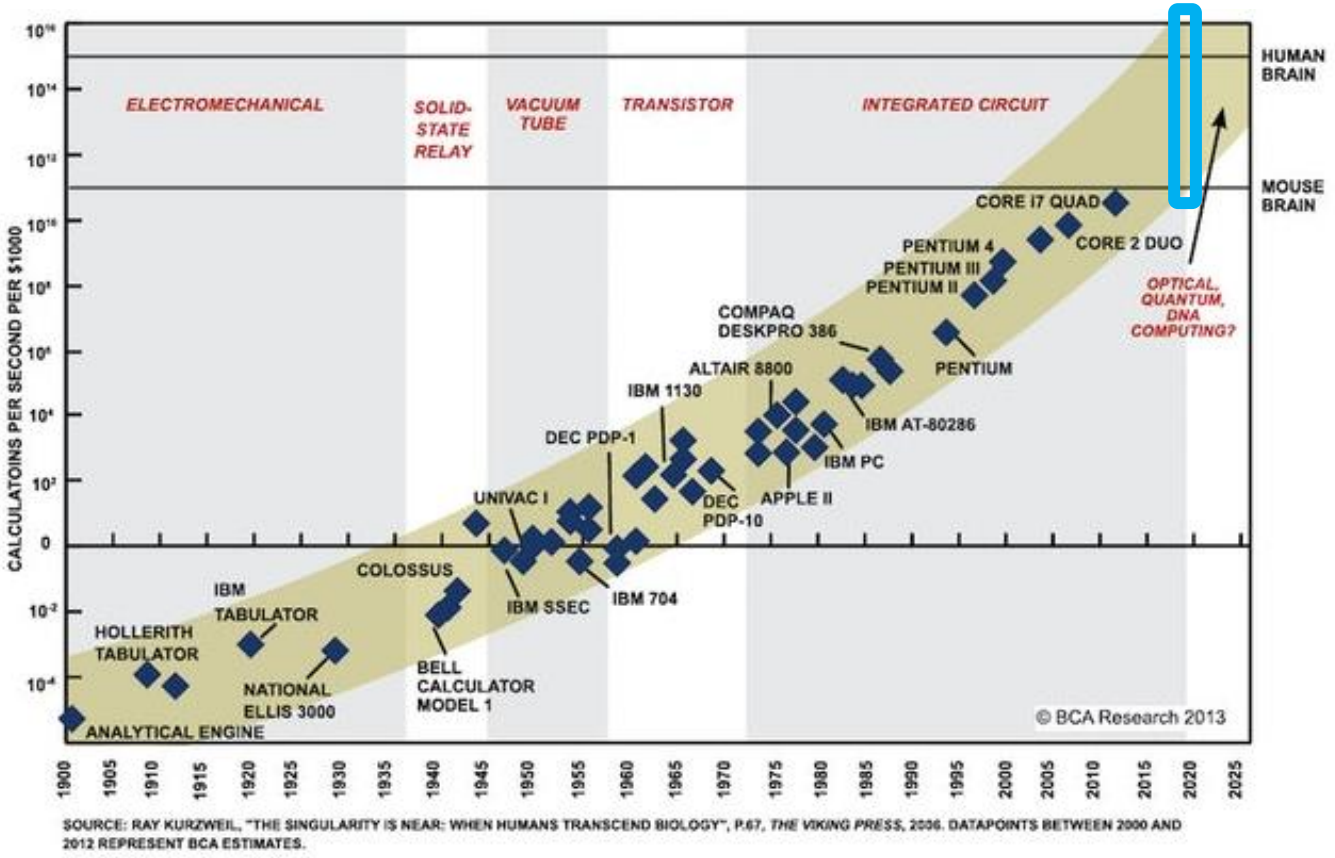
WITNESSES:

INVENTOR:

ATTORNEYS.

- **Biology and technology operate differently**
- **Airplanes are tech-like, not bird-like**

Mixing Biology and Technology (with AI Spin)

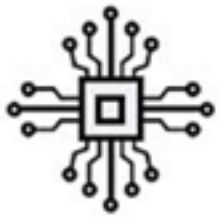


SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

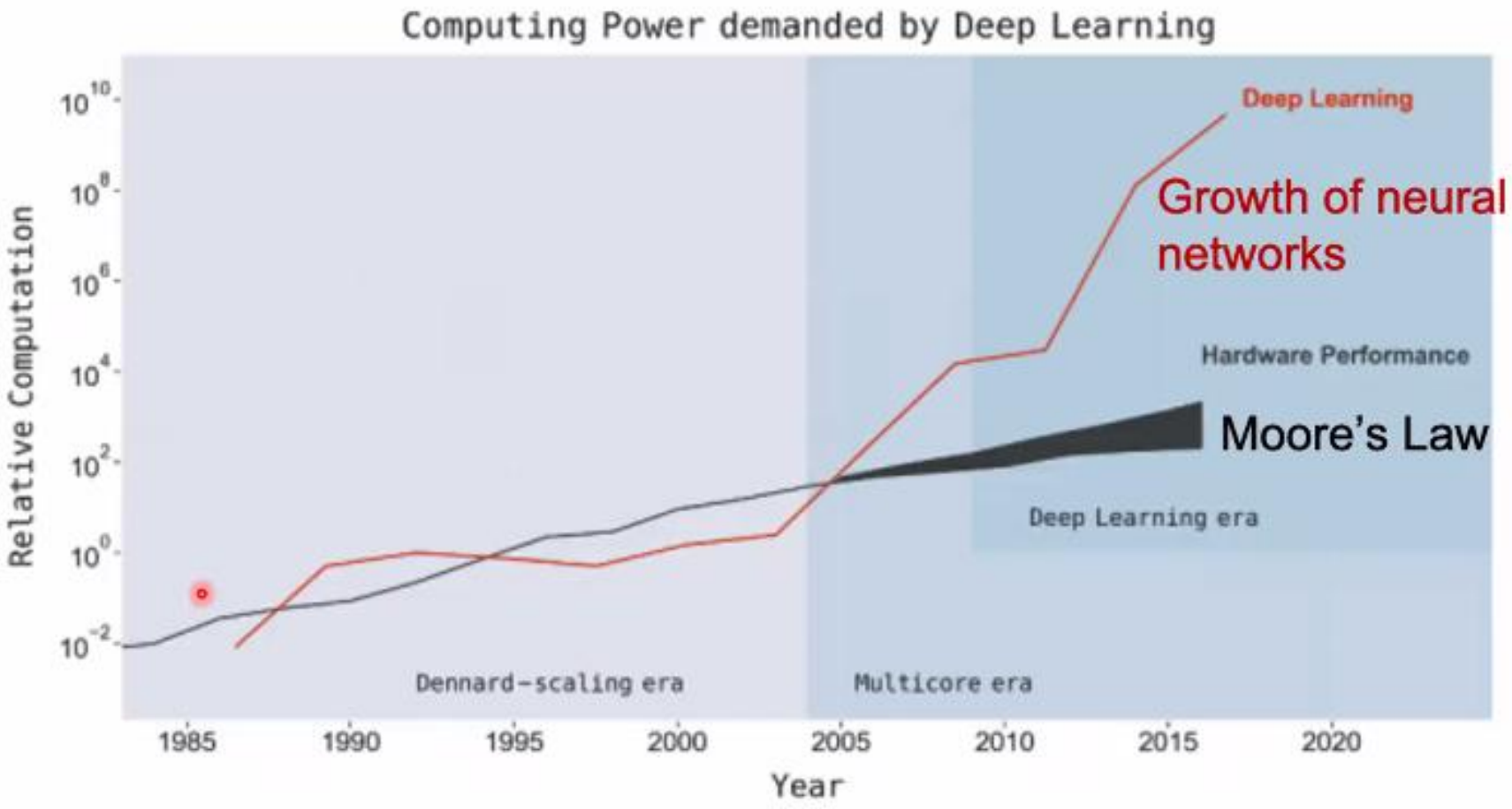
- There yet?
- **Singularity point gets delayed**
  - In 1998: 2020
  - In 2016: 2045
- **The meaning of AI is also redefined**
  - 1997: chess / IBM
  - Wide range of disembodied & embodied stuff

Brain vs Computer?

- **These "comparisons" aren't very useful…**

	Weight	Space	Processor Speed	Energy Efficiency
	3 pounds (1.4 kg)	1/6 basketball (80 cubic inches or 1,300 cm^3)	Up to 1,000,000 trillion operations per second	20 watts
	150 tons	Basketball court (cabinets over 4,350 square feet, or 400 m^2)	93,000 trillion operations per second	10 million watts

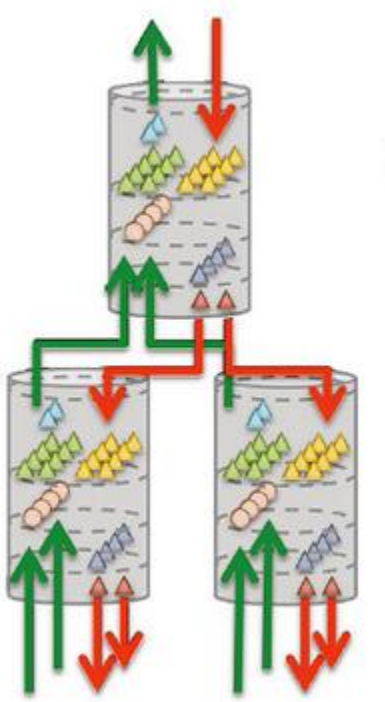
Neural Networks Outpace Moore's Law



Neil C. Thompson et al. (MIT) 2020 https://arxiv.org/pdf/2007.05558.pdf

Insights from Biology

Human Vision Model



- Hierarchical Temporal Memory (HTM)
- Bayesian Theory
- Particle Filtering
- Visual Cortex

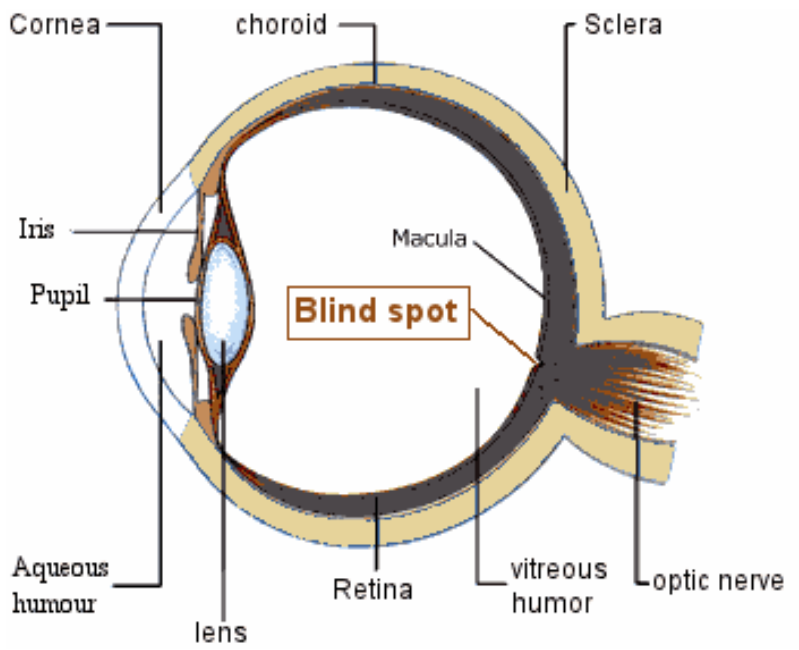
What is HTM?

- **Theory championed by Jeff Hawkins (creator of the Palm Pilot)**

  He noticed that in brain anatomy:

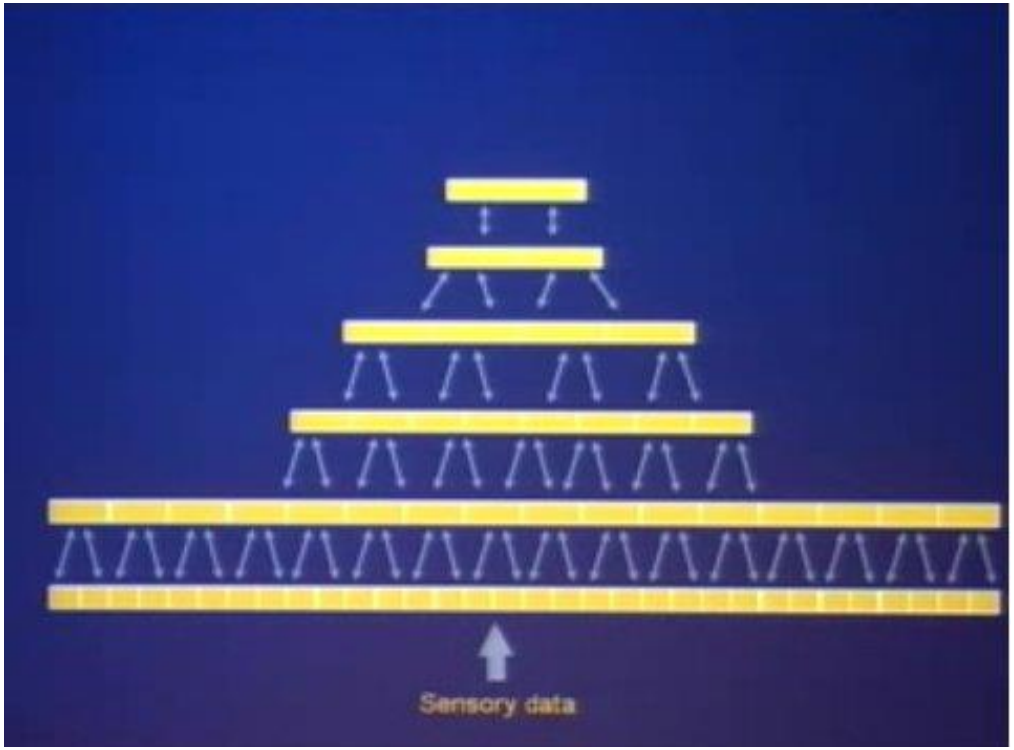
  - All layers look similar
  - Must have common underlying algorithm
- **Brain processes patterns**
- **Builds model of the world based on patterns**
- **Make predictions based on models**
- **Prediction is the foundation of intelligence**

Example in Vision



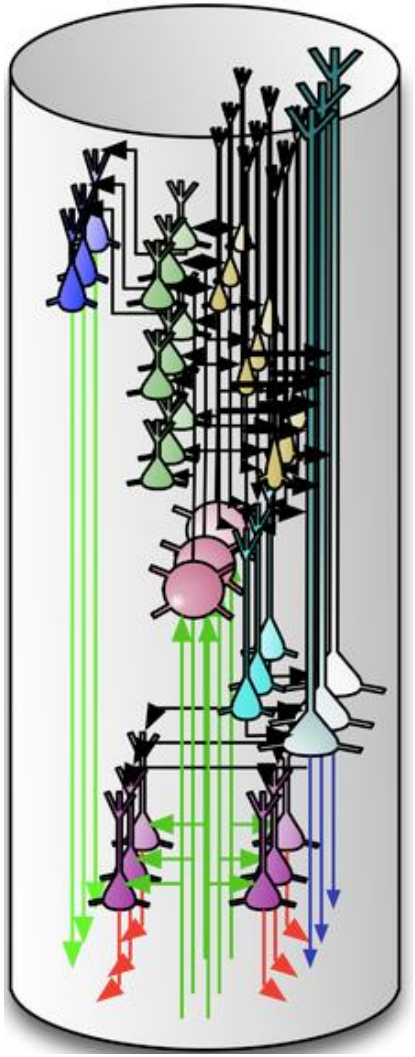
- **Everyone has 2 Blind Spots (one in each eye)**
- **Even when one eye is closed, you will not notice your blind spot**
- **The brain fills in the blind spot by making predictions about what the image in the blind spot should look like based on prior knowledge**

The Hierarchy



- **At each level or node**
  - Learns common spatial and temporal patterns
  - Learns common sequences
  - Forms representations
- **Each node sends representation up to the next node**
- **Higher nodes are more stable**

HTM



- Hierarchy is important because it allows the reuse of components
- Makes learning and storing information efficient
- Not a one-way feed-forward system
- Lower nodes have a lot of noise and ambiguity
- Stable representation is picked using Bayesian Belief Propagation

Bayesian Theory

- Use knowledge of prior events to predict future events
- Find Probability of A, given B

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- Probability of events are updated as more detail is given

Uses

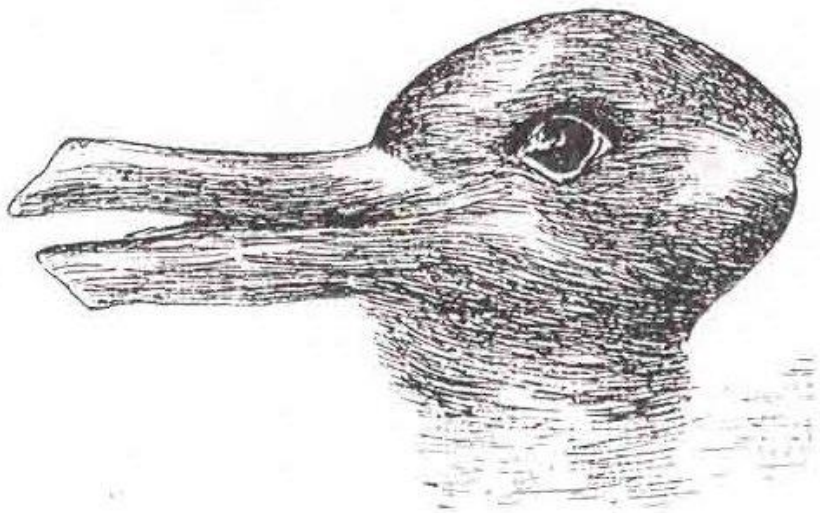
- Google Search
- Google Gmail and Priority inbox
- Microsoft Notifications on Windows Phone
- Voice Recognition Technology
- Used in medicine to correlate symptoms with diseases

Particle Filtering

- In HTM, each node sends representation of highest probability up to the next
- What happens when two competing interpretations both have high probabilities (the input is ambiguous)?
- Particle Filtering approach is to allow time for longer feedback loops to have an influence

Examples of Particle Filtering

- **What do you see in these images?**



Organization of the Retina

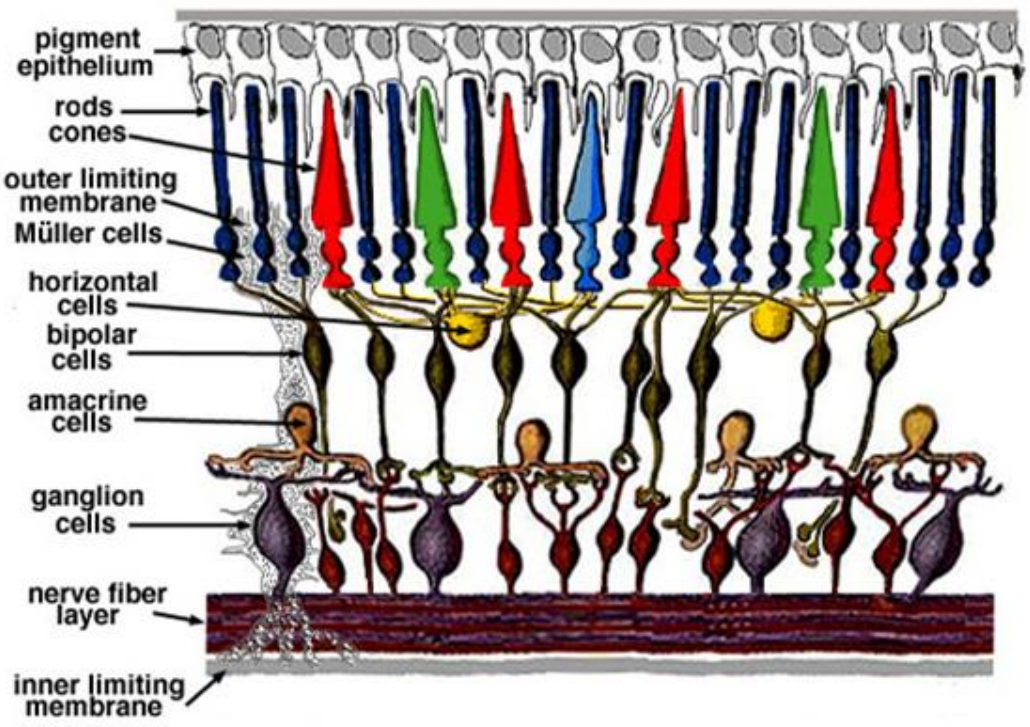
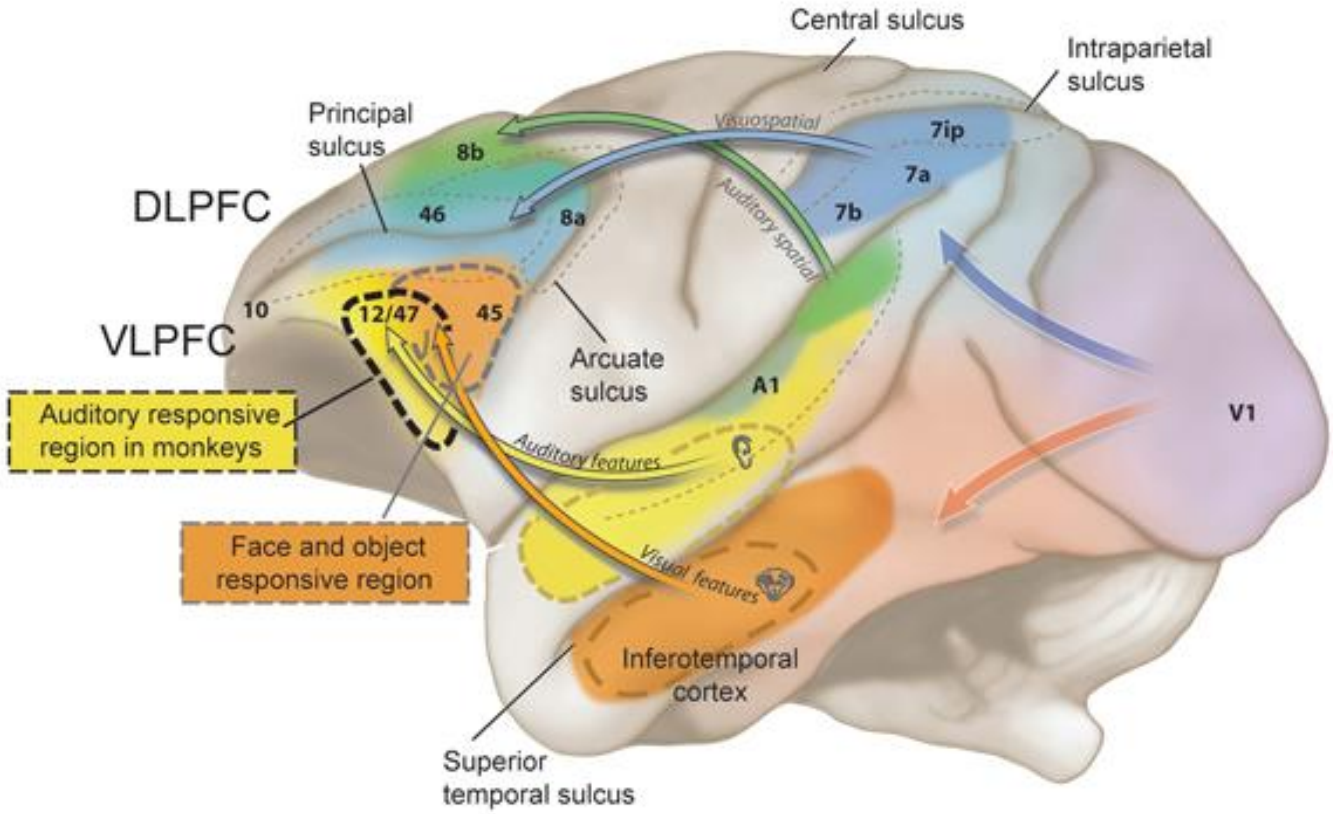


Fig. 2. Simple diagram of the organization of the retina.

- **Retina converts light into neural signals and sends these neural signals to the brain for visual recognition**
- **Rods enable vision in poor light, cones enable color**
- **Horizontal cells regulate signals from rods and cones**
- **Bipolar cells Tx signals from photoreceptors to ganglion**
- **Pigment: a protective layer**

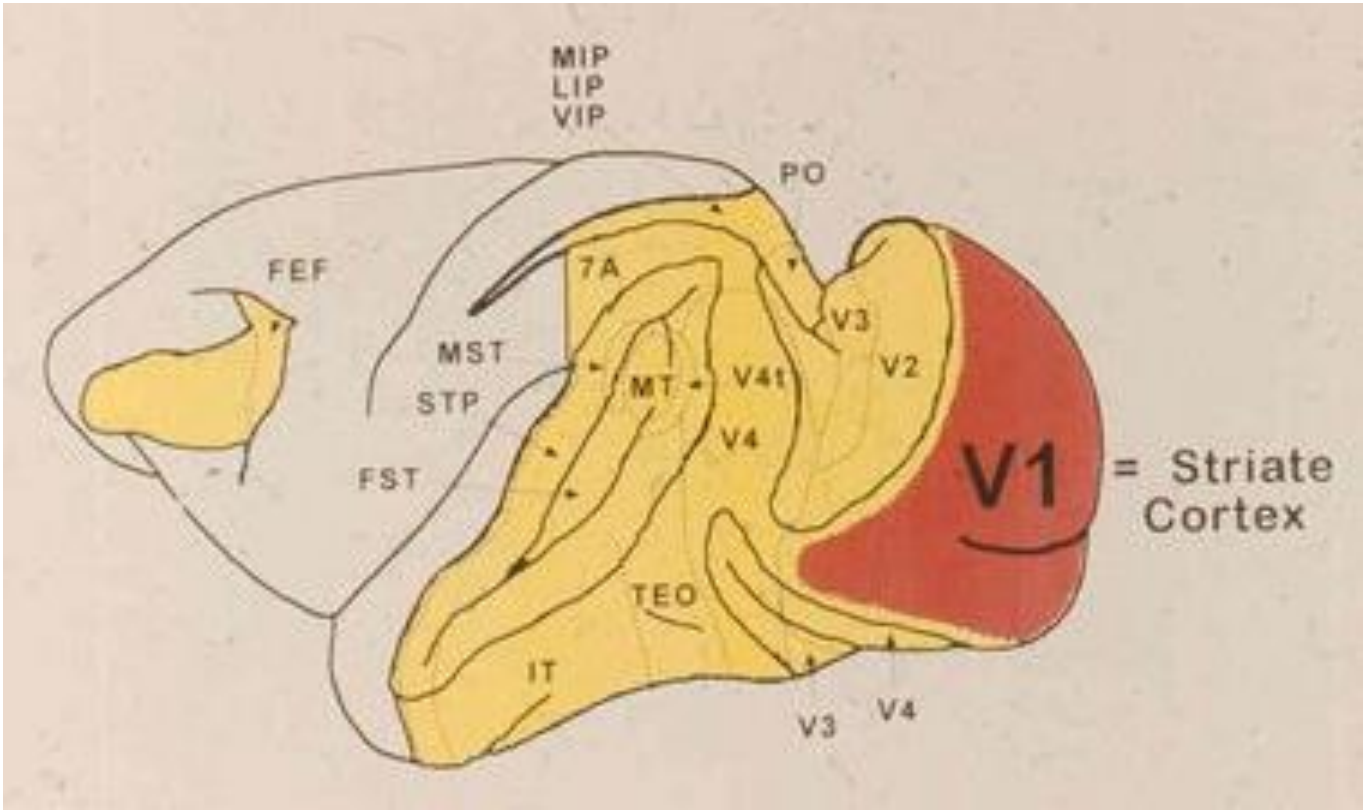
Adapted from: webvision.med.utah.edu

Visual Cortex



- **Retina**
- **LGN (Lateral Geniculate Nucleus)**
- **V1 , V2, V3, V4, V5/MT, IT**

Visual Cortex



- Retina
- LGN (Lateral Geniculate Nucleus)
- V1 , V2, V3, V4, V5/MT, IT

Visual Cortex (1/2)

- Lower levels have higher spatial and temporal resolution
- V1 neurons respond to precise small areas from the retina
- IT neurons respond to larger areas, loses resolution
- Low levels (V1) generally process from a fine to a coarse manner
- Higher levels (IT) does the opposite: coarse to fine (and there is a dense feedback network to lower levels)

Visual Cortex (2/2)

- Visual system has generally been thought of as a feed-forward system
- Lower levels send information up and the information converges to form a representation of something
- In order for HTM to work, there must also be a feedback loop
- Higher levels must send information down that influences the activity of the lower levels

Topographical representations of mental images in primary visual cortex

Stephen M. Kosslyn*†, William L. Thompson*, Irene J. Kim* & Nathaniel M. Alpert‡

*Department of Psychology, Harvard University, Cambridge, Massachusetts 02138, USA
†Department of Neurology and ‡Department of Radiology, Massachusetts General Hospital, Boston, Massachusetts 02114, USA

WE report here the use of positron emission tomography (PET) to reveal that the primary visual cortex is activated when subjects close their eyes and visualize objects. The size of the image is systematically related to the location of maximal activity, which is as expected because the earliest visual areas are spatially organized[1–5]. These results were only evident, however, when imag-

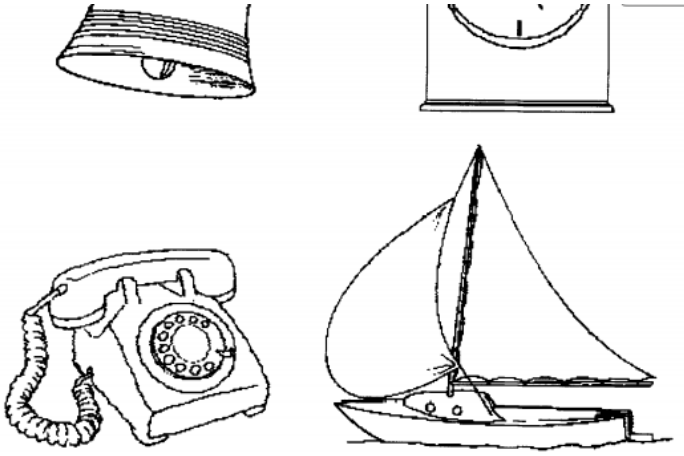
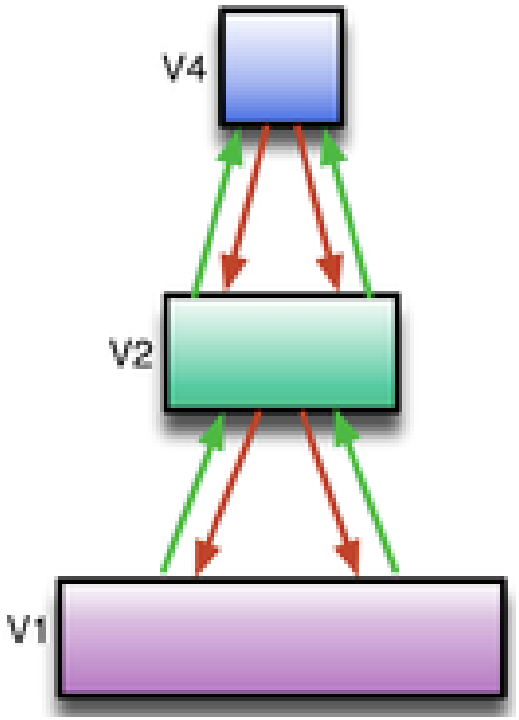


FIG. 1 Stimuli used in four of the five conditions. In the listening baseline condition, subjects received trials of the following sort. First they heard the name of a common object (such as 'anchor'), and 4 s later heard a spatial comparison term (such as 'right higher'), and then responded. One second later, another trial was presented. Subjects were told to close their eyes and respond as quickly as possible on

- Kosslyn et al. showed with fMRI studies that V1 region responds differently when patients were asked to close their eyes and imagine different objects
- Objects with more fine details activated the V1 region more
- Shows that even when process is begun on a higher level, the lower level V1 will still be activated
- V1 will only be activated if scene is ambiguous without high resolution information

Interaction Between Levels



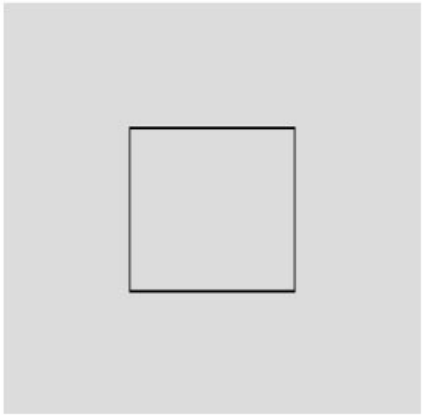
- When input is received, higher levels are sensitive to global context
- Lower levels process on a local scale

- As the levels interact, the lower levels become sensitive to global context while the higher level become sensitive to more precise detailed information

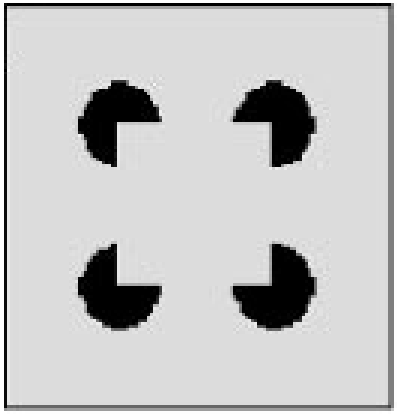
Edge Detection

- **Studies show that V1 neurons are involved in edge (contour) detection**
- **V2 neurons are involved in illusory contour**

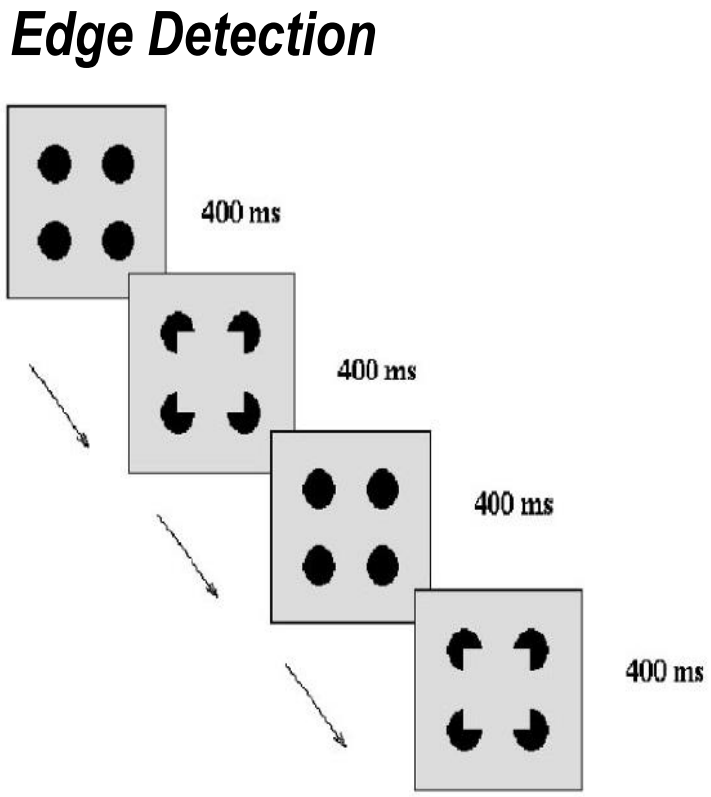
V1 =



V2 =

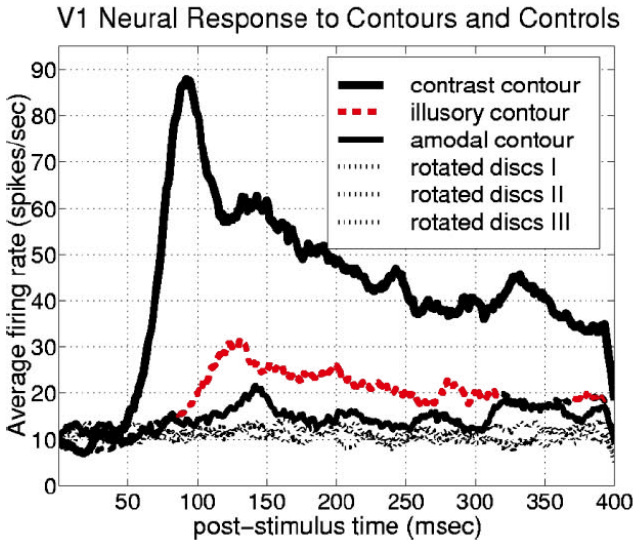


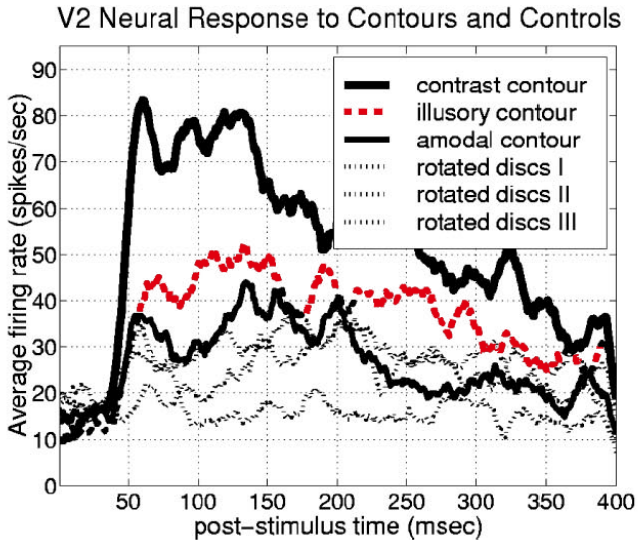
Contour Test

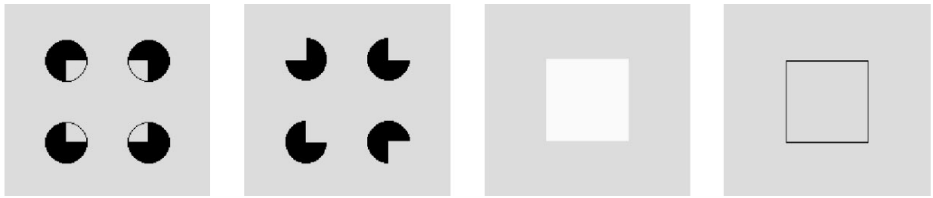


- Monkeys are shown series of 4 dots
- "Pac-man" dots are arranged so that there seems to be an illusionary square in the middle
- Electrodes measure response activity of the monkey's V1 and V2 regions

Results


V1 Neural Response to Contours and Controls
contrast contour
illusory contour
amodal contour
rotated discs I
rotated discs II
rotated discs III
Average firing rate (spikes/sec)
post–stimulus time (msec)


V2 Neural Response to Contours and Controls
contrast contour
illusory contour
amodal contour
rotated discs I
rotated discs II
rotated discs III
Average firing rate (spikes/sec)
post–stimulus time (msec)


Amodal
Rotated
Contrast
Normal

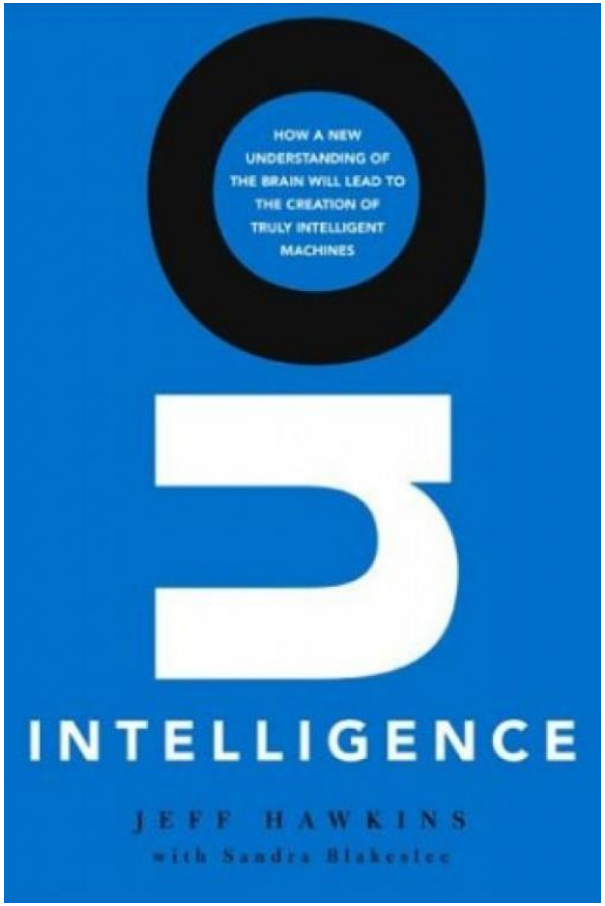
What Does This Mean?

- V1 was able to respond to illusory contours but at latency of ~35ms after V2
- V2 detects existence of illusory contour by integrating information from spatially more global context
- V2 then feedback to V1 and modulates V1 to become sensitive to illusory contours
- This is an example of particle filtering: V1 has to wait for feedback from V2

Conclusions from the Study

- Thus it is possible that visual cortex fits in a HTM model and is governed by Bayesian principles and particle filtering
- Low levels form “representations” or hypotheses about input
- Higher levels modulates the probability distribution of competing hypotheses using prior knowledge
- There are both feedforward and feedback signals relaying between higher and lower levels

Further Reading



- Dileep George & Jeff Hawkins. *Towards a Mathematical Theory of Cortical Micro-circuits.* PLoS Computational Biology, October 2009, Vol. 5, Issue 10.
- Lee TS, Mumford D. *Hierarchical Bayesian inference in the visual cortex.* Journal of the Optical Society of America. 2003;2:1434–1448.
- S. Kosslyn, W. L. Thompson, I. J. Kim, and N. M. Alpert. *Topographical representations of mental images in primary visual cortex.* Nature 378, 496–498 (1995)
- D. George, J. Hawkins. *A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex.* IEEE Int. Joint Conf. on Neural Networks, 2005.

⇦ ***Great book*** **| download** PDF

- Hawkins J, Blakeslee S. On Intelligence. New York: Henry Holt and Company; 2004.

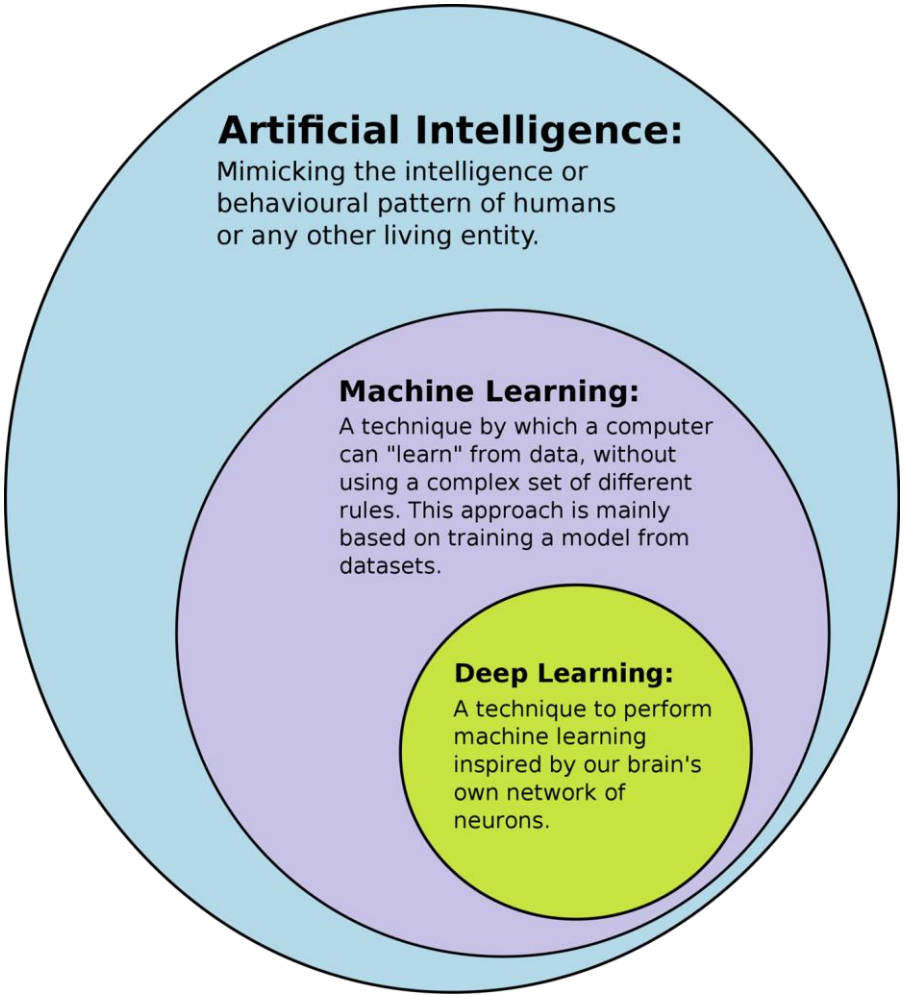
Towards AI/ML Hardware

Deep Learning Revolution

- **Introduction**
- Deep Learning Hardware
- Current Research
- Future: Self-Supervised Learning
- Neuromorphic Computing and Chips

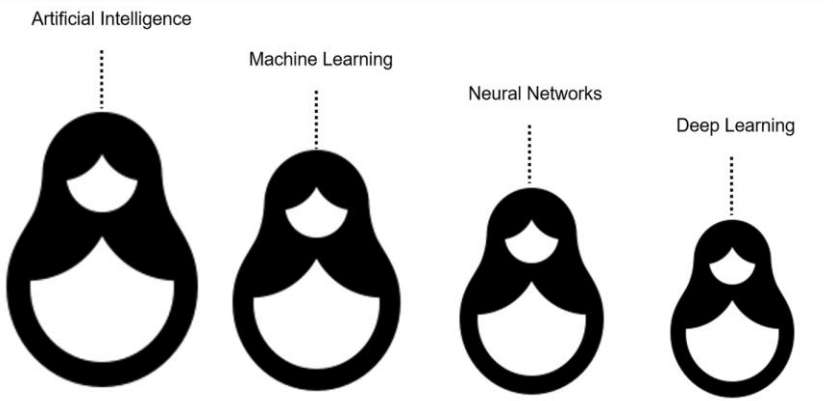
What is Deep Learning?

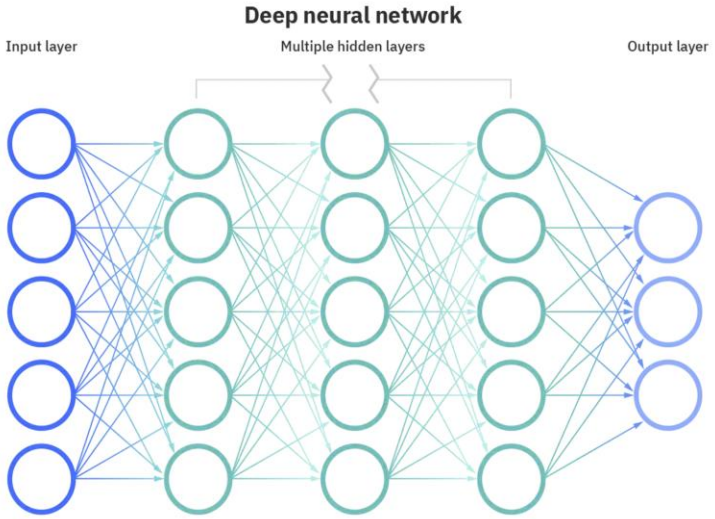
- Deep Learning is a neural network with multiple layers and tries to mimic how the human brain processes information and learns.
- Deep Learning is a machine learning algorithm that uses multiple layers to extract higher-level features from the raw input.



Machine Learning and Deep Learning

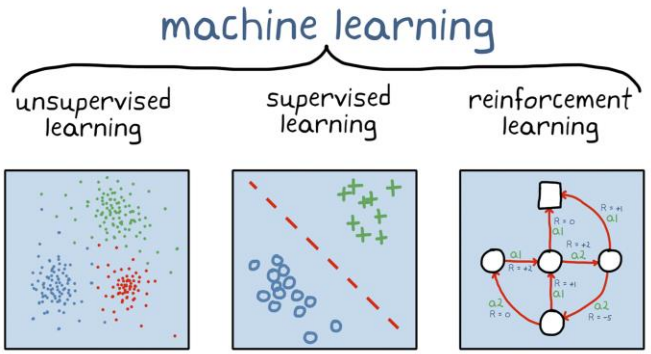
- Machine Learning requires more preprocessing to allow the algorithms to work
- Deep Learning can use unstructured data for its algorithms and uses more layers typically





Supervised, Unsupervised and Reinforcement Learning

- Supervised Learning
  - The most common one
  - Trains and tests on a labelled datasets
- Unsupervised Learning
  - Uses unlabeled datasets
  - It “discovers” hidden features and patterns without human processing
- Reinforcement Learning
  - Focus on finding a balance between exploration and exploitation
  - Taking an action to get maximum reward

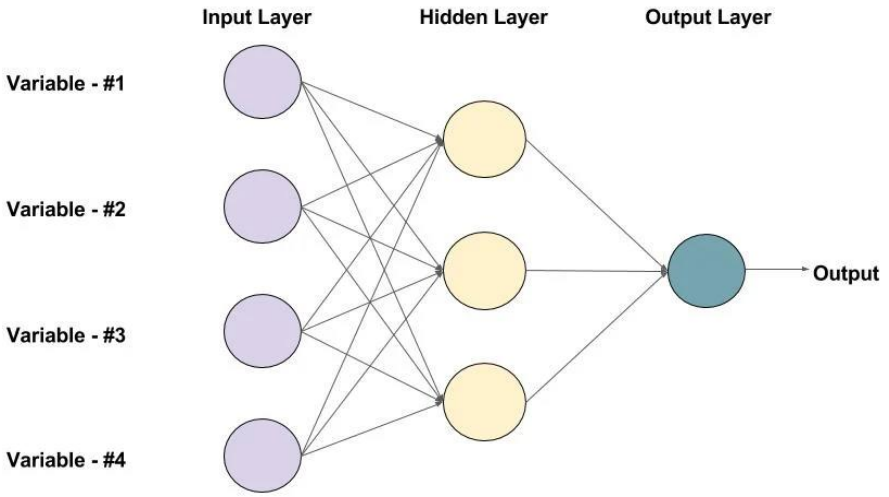


Types of Neural Networks

- Feedfoward Neural Network
- Recurrent Neural Network (RNN)
- Convolutional Neural Network (CNN)

Feedforward Neural Network

- Most basic neural network
- Information flows from one layer to the next
- Takes a long time to train for large datasets

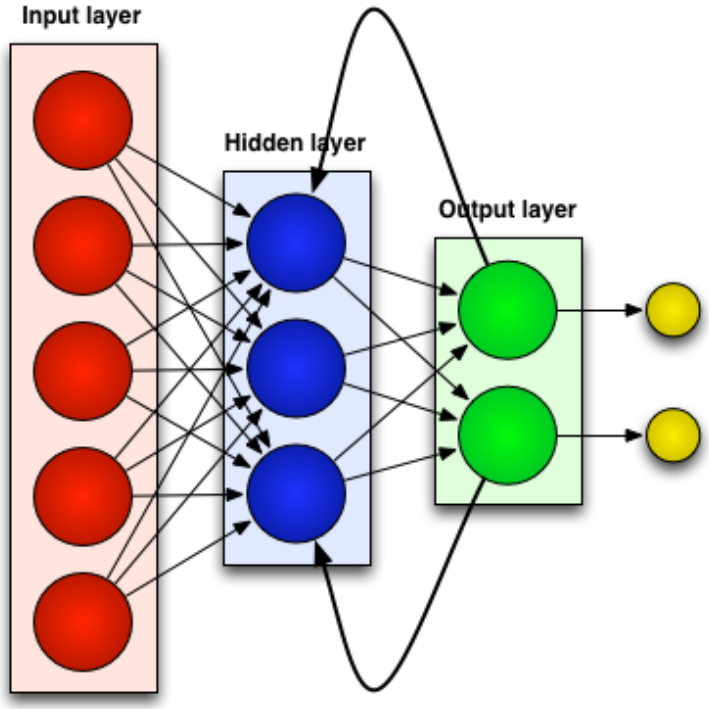


An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

Recurrent Neural Network

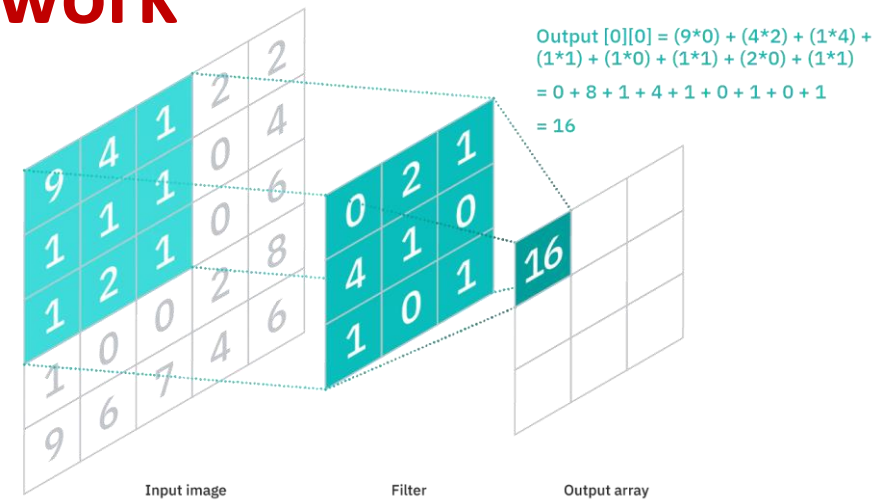
- Adds a feedback component
  - Gives the network a time dependence

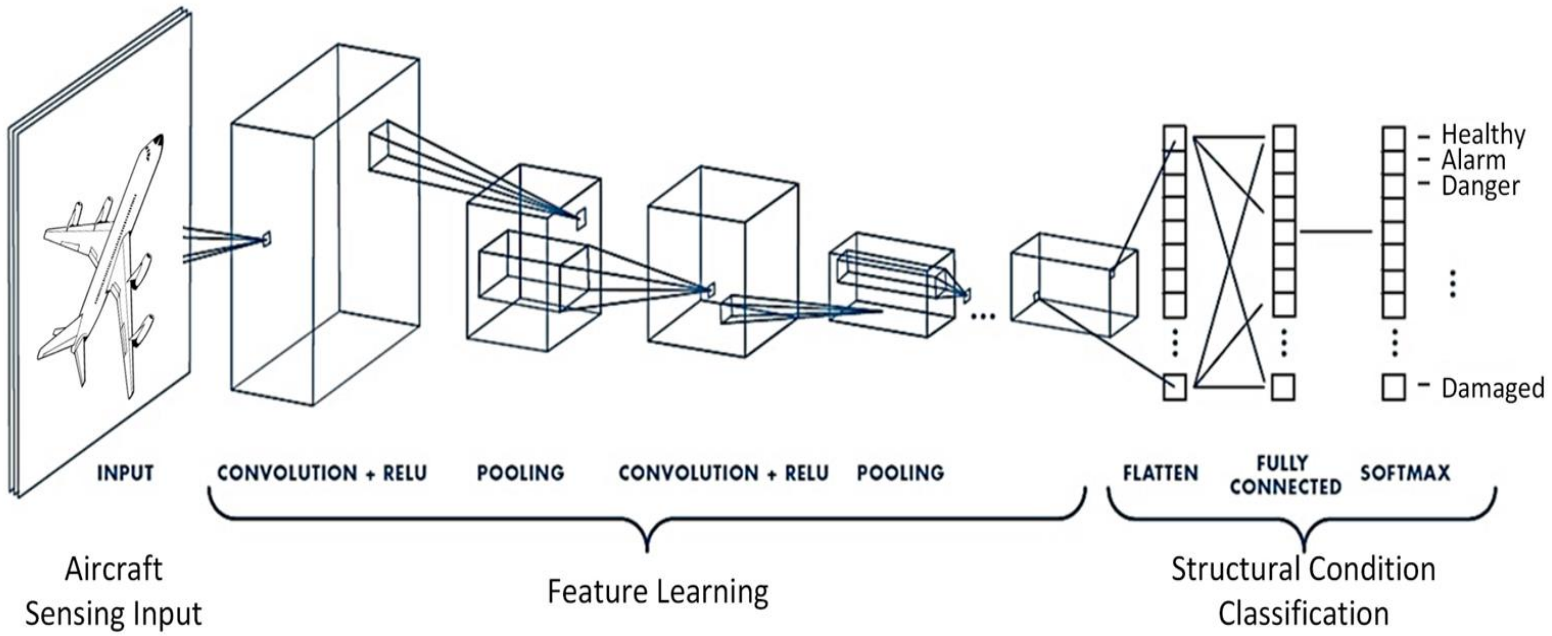
- Used in speech and handwriting recognition



Convolutional Neural Network

- Takes advantage of convolution to reduce computations
- Used in image recognition





Video tutorial: *https://youtu.be/pj9-rr1wDhM?si=W8nkK_yVLrOXgWzg*

History of Deep Learning and Hardware

Limitations

- hardware
- open source software
- datasets

Timeline

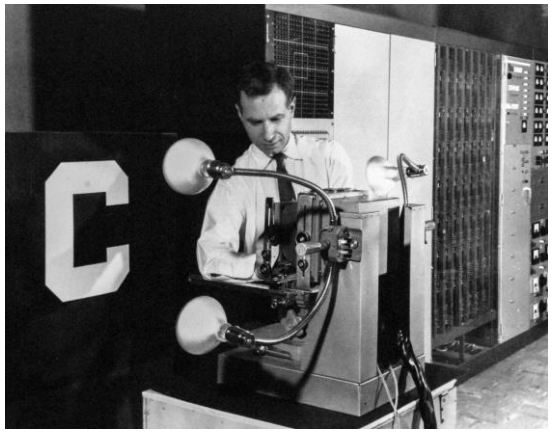
- 1957
  - Perceptron ~ motorized potentiometers
  - Adaline ~ electrochemical memistors
- 1980s
  - Back Propagation
  - Neural Network Chips
  - CNN development ~ shift registers
- 2000s
  - rebranded domain as Deep Learning
  - ReLU
  - applications in speech recognition
  - further use of CNN

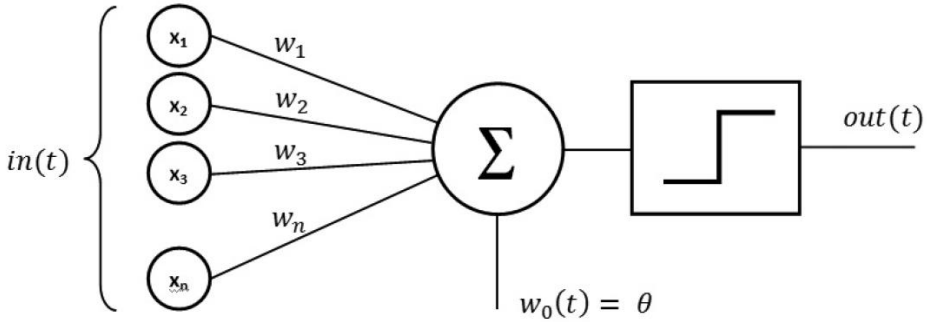
Deep Learning Revolution

- Introduction
- **Deep Learning Hardware**
- Current Research
- Future: Self-Supervised Learning
- Neuromorphic Computing and Chips

Perceptrons

- Invented by Frank Rosenblatt in 1957
- Analog computer with 400 photocells as input, with weights that were variable resistance potentiometers adjusted by motors
- Early precursor of deep learning networks, able to classify patterns/images
  - Single Layer Neural Network - classifies input into two possible categories
- Discovers a set of weights automatically through training examples
  - Makes a prediction, then tweaks itself to make a more informed prediction next time
- Limitation: Could only separate categories that are linearly separable





Source: The Deep Learning Revolution - Machine Intelligence Meets Human Intelligence

Hybrid Digital/Analog Chips

54 neuron mixed analog-digital chip (1987)

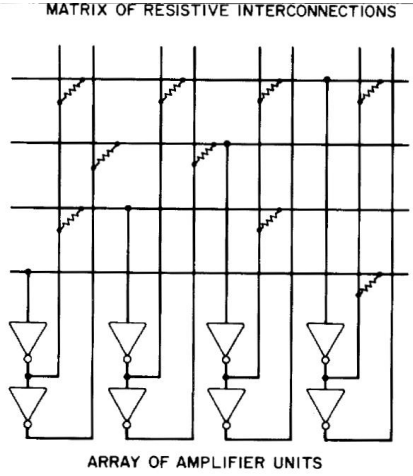


FIGURE 1—Circuit schematic. Connections between input and output lines, drawn as resistors, are provided by coupling elements shown in Figure 5.

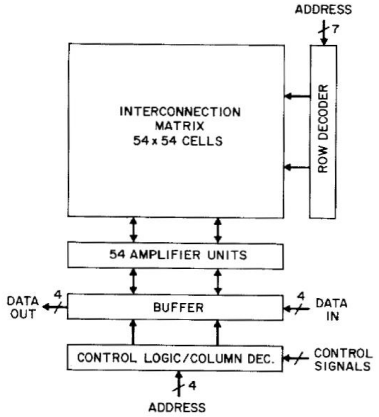


FIGURE 3—Block diagram of chip.

- Resistor array implements vector-matrix multiplication
- Issues with I/O bandwidth

Source: H. Graf, P. de Vegvar, “A CMOS Associative Memory Chip Based on Neural Networks”, ISSCC, pp. 304- 305, 1987.

Net32k, 256 neurons (1991)

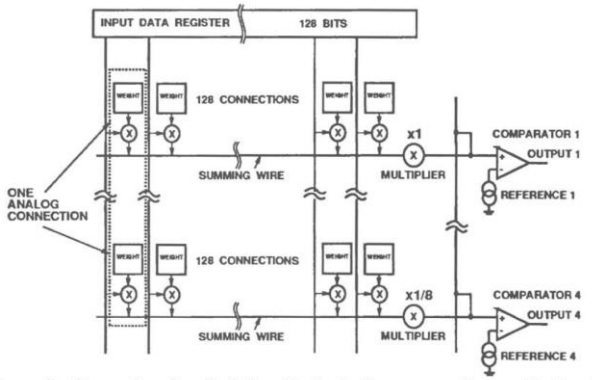


Figure 2: Connecting four building blocks to form connections with four bits of resolution

- Used shift registers to enable convolutional neural networks
- Used for image processing applications

Source: H.P. Graf, R. Janow, D. Henderson, R. Lee, “Reconfigurable Neural Net chip with 32K Connections”, Advances in Neural Information Processing Systems, pp. 1032-1038, 1991.

Computational Power

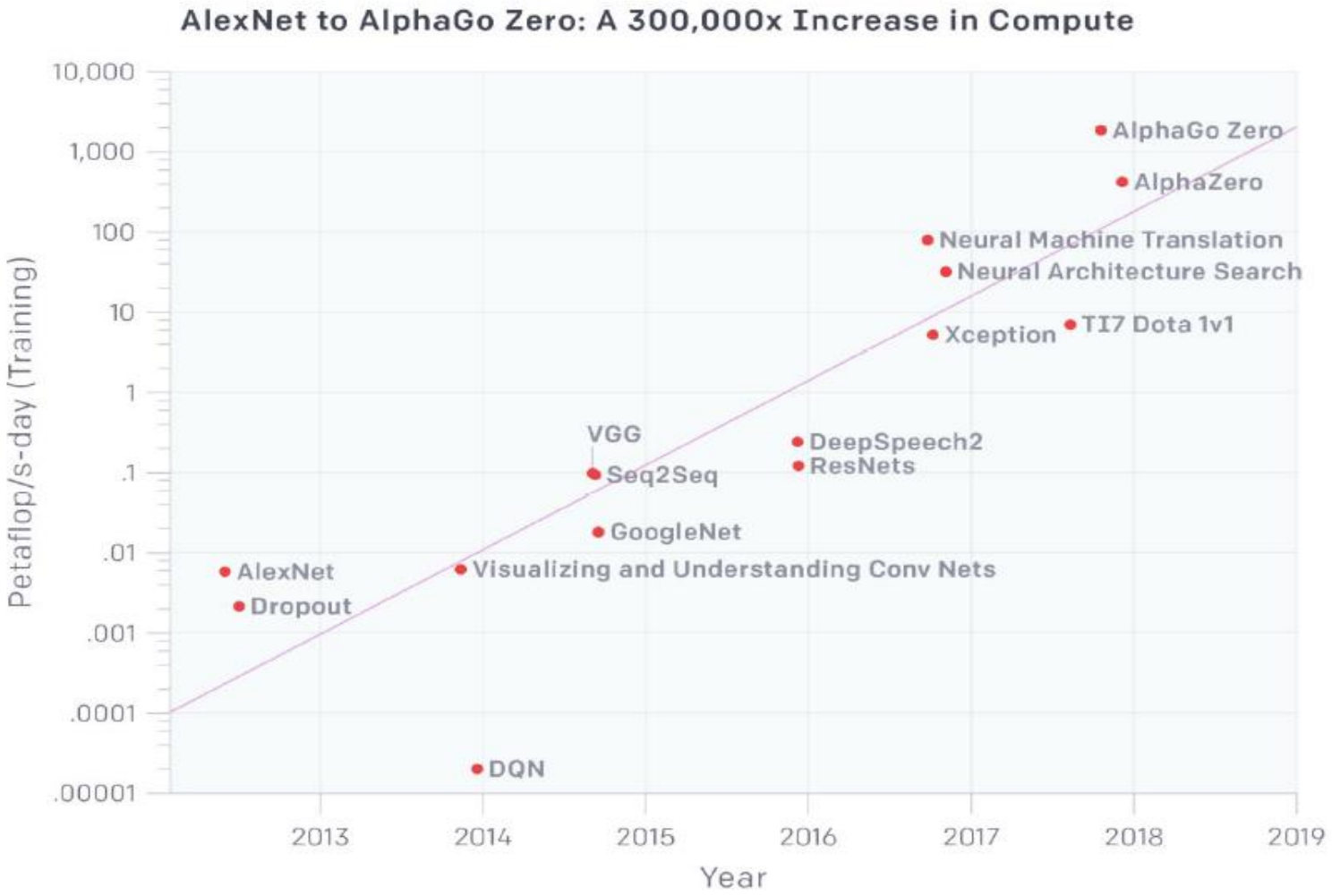
- Not until the 2000's that computers became powerful enough to train large neural networks on realistic, real-world problems
- Development of chips for deep learning:
  - Computations consist of dense linear algebra calculations
  - Highly parallelizable

Table 2: Comparing state-of-the-art AI chips to state-of-the-art CPUs

	Training		Inference		Generality[88]	Inference accuracy[89]
	Efficiency	Speed	Efficiency	Speed		
CPU	1x baseline				Very High	~98-99.7%
GPU	~10-100x	~10-1,000x	~1-10x	~1-100x	High	~98-99.7%
FPGA	-	-	~10-100x	~10-100x	Medium	~95-99%
ASIC	~100-1,000x	~10-1,000x	~100-1,000x	~10-1,000x	Low	~90-98%

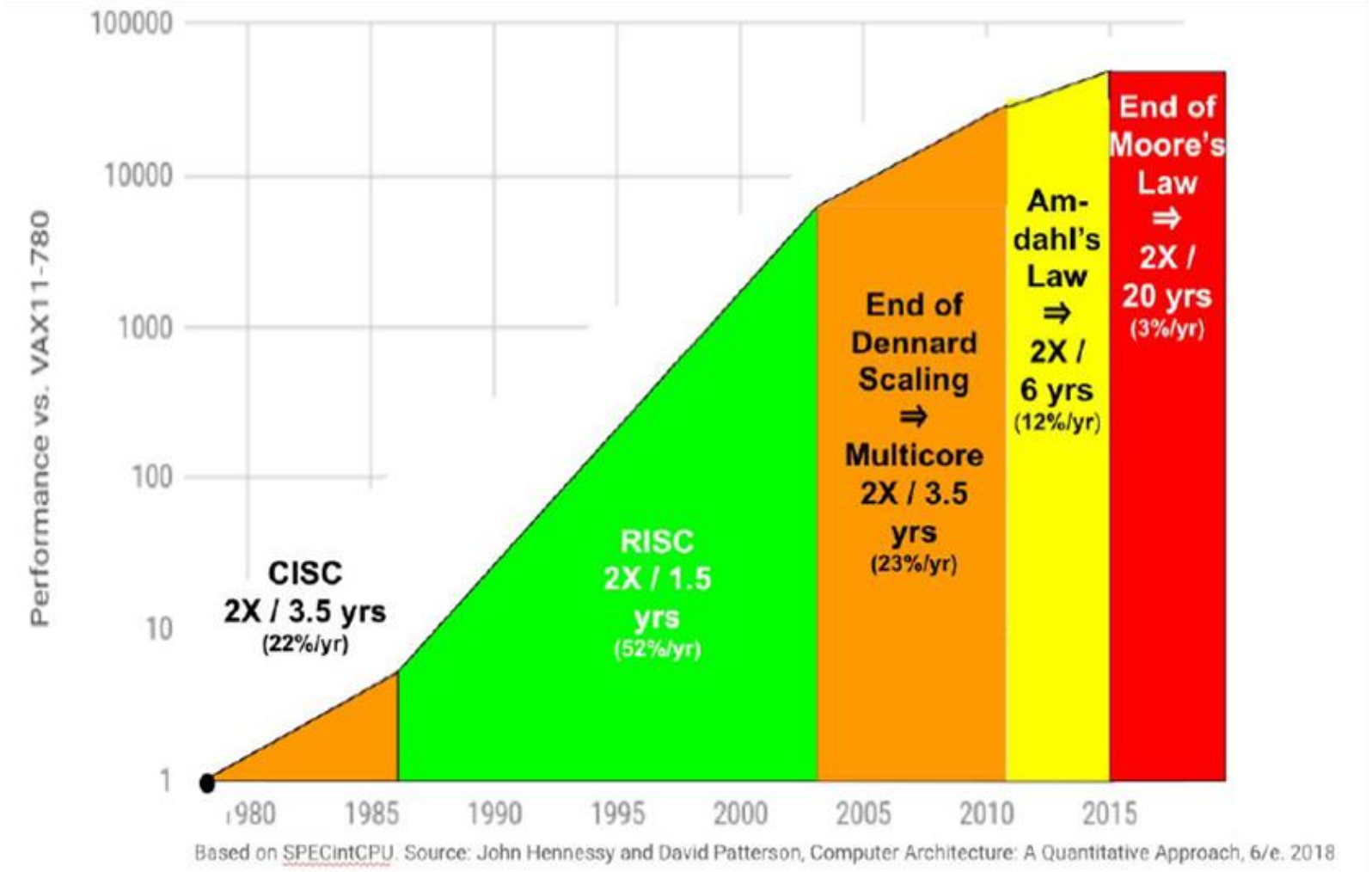
Saif M. Khan and Alexander Mann, "AI Chips: What They Are and Why They Matter" (Center for Security and Emerging Technology, April 2020), cset.georgetown.edu/research/ai-chips-what-they-are-and-why-they-matter/.

Computational Capacity for Learning



Dean, Jeffrey. "1.1 the deep learning revolution and its implications for computer architecture and chip design." *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020.

Computational Power



Deep Learning Hardware Industry

- Many startups that began in the past decade
- Focused on different areas such as low power mobile SoCs and data centers
- Other Technology
  - Silicon Photonics (Lightelligence)
  - Neuromorphic Hardware (Rain Neuromorphics)
  - Wafer Scale Integration (Cerebras)

IC Vendors	Intel, Qualcomm, Nvidia, Samsung, AMD, Xilinx, IBM, STMicroelectronics, NXP, Marvell, MediaTek, HiSilicon, Rockchip, Renesas Electronics, Ambarella, Sony	17
Tech Giants & HPC Vendors	Google, Amazon_AWS, Microsoft, Apple, Aliyun, Alibaba Group, Tencent Cloud, Baidu, Baidu Cloud, HUAWEI, Fujitsu, Nokia, Facebook, HPE, Tesla, LG, SK Telecom	16
IP Vendors	ARM, Synopsys, Imagination, CEVA, Cadence, VeriSilicon, Videantis	7
Startups in China	Cambricon, Horizon Robotics, Bitmain, Chipintelli, Thinkforce, Unisound, AISpeech, Rokid, NextVPU, Canaan, Enflame, Eesay Tech, WITINMEM, TSING MICRO, Black Sesame, Corerain	16
Startups Worldwide	Cerebras, Graphcore, PEZY, Tenstorrent, Blaize, Koniku, Adapteva, Knowm, Mythic, Kalray, BrainChip, AImotive, Leepmind, Krtkl, NovuMind, REM, TERADEEP, Deep Vision, Groq, Kneron, Esperanto Technologies, Gyrfalcon Technology, SambaNova Systems, GreenWaves Technology, Lightelligence, Lightmatter, ThinkSilicon, Innogrit, Kortiq, Hailo,Tachyum,AlphaICs,Syntiant, aiCTX, Flex Logix, Preferred Network, Cornami, Anaflash, Optaylsys, Eta Compute, Achronix, Areanna AI, Neuroblade, Luminous Computing, Efinix, AISTORM, SiMa.ai,Untether AI, GrAI Matter Lab, Rain Neuromorphics, Applied Brain Research, XMOS, DinoPlusAI, Furiosa AI, Perceive, SimpleMachines, Neureality, Analog Inference, Quadric, EdgeQ, Innatera Nanosystems, Ceremorphic	62

Source: https://basicmi.github.io/AI-Chip

Deep Learning Revolution

- Introduction
- Deep Learning Hardware
- **Current Research**
- Future: Self-Supervised Learning
- Neuromorphic Computing and Chips

Edge Computing and Deep Learning

- Why?
  - Lower Latency
  - Bandwidth and energy cost of communication to the cloud
  - Security and Privacy

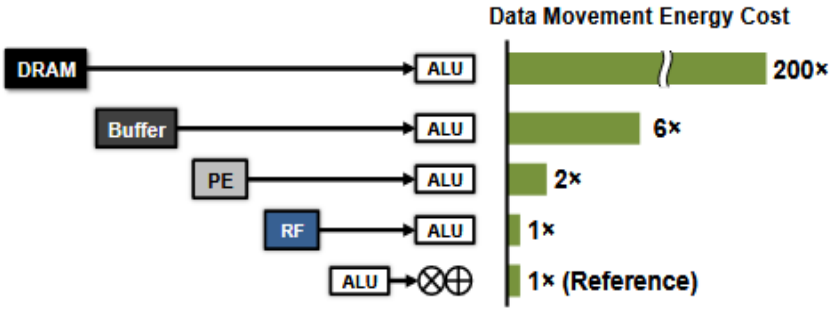
- Metrics to be Optimized
  - Latency
  - Energy consumption
  - Accuracy
  - Cost/Area

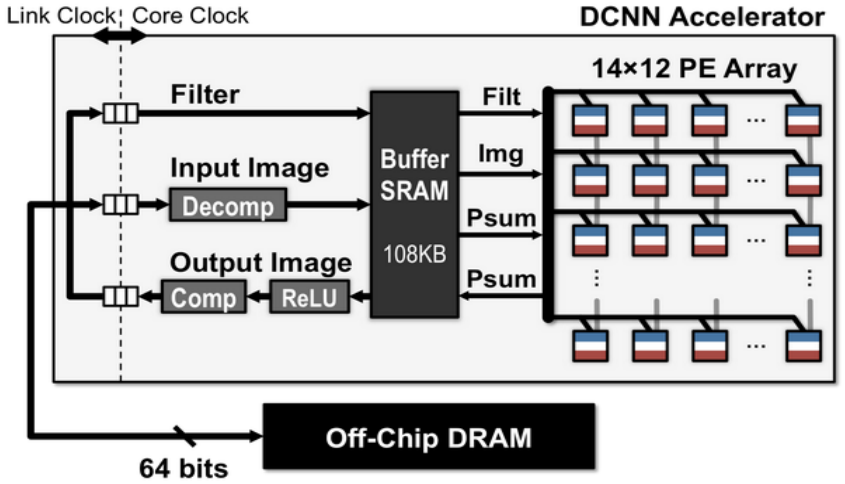
CPUs and GPUs

- Traditionally used for ML applications
- Perform multiply-accumulates (MACs) using highly parallelized SIMD architectures.
- Classification represented by Matrix multiplications.
- Have efficient memory caches to minimize access to RAM.
- Consume more energy than optimized hardware.

Accelerators - FPGAs and ASICs

- Low Power consumption
- Computational throughput comparable or higher than CPU/GPUs
- Must access external DRAM for data and weights - require data-reuse based design.
- FPGAs:
  - Highly reprogrammable
  - Low Memory - crucial for DNNs
  - Sacrifice Performance for Flexibility
- ASICs:
  - Highest performance
  - Low cost and energy
  - Require off-chip memory
  - Low generality





Eyeriss Architecture

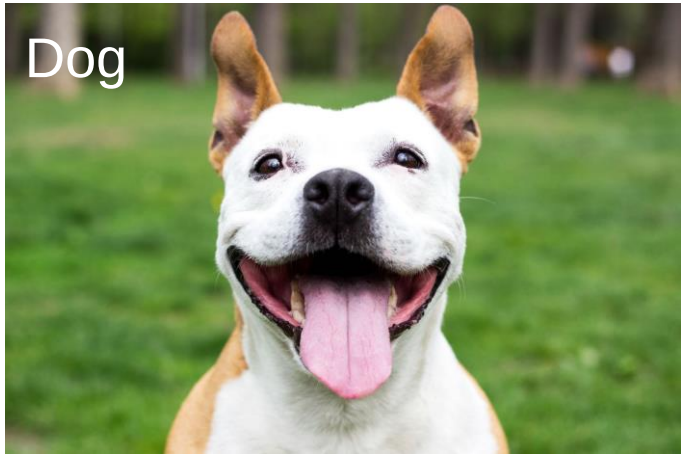
Algorithmic Improvements

- Quantization:
  - 8-bit and 16-bit fixed point operations are sufficient in many Deep Neural Networks and can lead to up to 2-3x improvement in energy usage or throughput.
  - Tolerable drops in accuracy.
  - Novel architectures use binary weights (+1/-1) for large energy and performance efficiency gains.
- Sparsity and Pruning:
  - Removing energy consuming weights with a relatively low impact on classification.
  - Transformations of weights to increase sparsity to reduce the number of MACs.

Deep Learning Revolution

- Introduction
- Deep Learning Hardware
- Current Research
- **Future: Self-Supervised Learning**
- Neuromorphic Computing and Chips

Self-Supervised Learning



Self-Supervised Learning

Self-Supervised Learning

(a) Input context

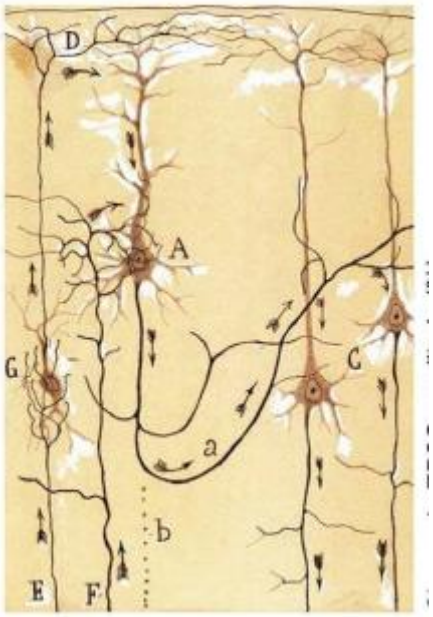
(b) Output

Source: Arxiv

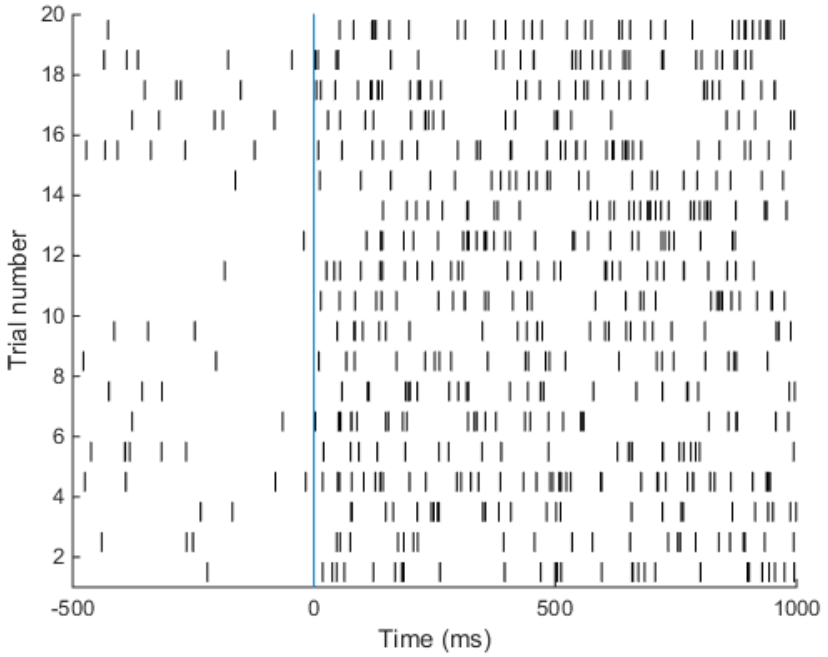
Deep Learning Revolution

- Introduction
- Deep Learning Hardware
- Current Research
- Future: Self-Supervised Learning
- **Neuromorphic Computing and Chips**

Neuromorphic Computing

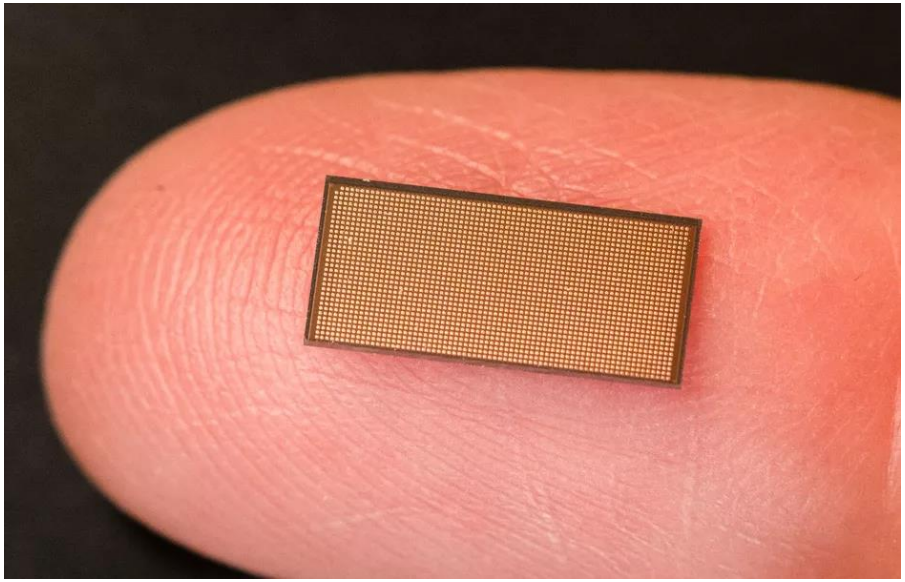


Drawing of neuron connections in brain



Spike train of a single neuron in the absence and presence of stimuli

Neuromorphic Chips



Intel’s Loihi 2

“We can expect AIs to have operating systems comparable to the one in our brain by 2050.” - J. Seijnowski, Terrence. *The Deep Learning Revolution*. MIT Press, 2018

Deep Learning / References

http://yann.lecun.com/exdb/publis/pdf/lecun-isscc-19.pdf

https://www.ibm.com/cloud/learn/deep-learning

https://eyeriss.mit.edu/

Talib, M.A., Majzoub, S., Nasir, Q. *et al.* A systematic literature review on hardware implementation of artificial intelligence algorithms. *J Supercomput* **77,** 1897–1938 (2021). https://doi.org/10.1007/s11227-020-03325-8

V. Sze, Y. -H. Chen, J. Emer, A. Suleiman and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 2017, pp. 1-8, doi: 10.1109/CICC.2017.7993626.

Graph Neural Networks (GNNs)

GNNs / Outline

1. Brief background on neural networks
2. Data as Graphs
3. Introduction and Motivation for GNNs
4. Architecture Overview
5. Applications

Neural Networks Review

- Neural network - artificial representation of brain neurons
  - Typically organized in layers with weighted edges (weights)
  - Foundational in modern machine/deep learning
- NNs contain trainable parameters, typically adjusted through backpropagation
  - Output layer used to complete either regression or classification task
  - “Learning” typically seeks to minimize loss/error function
- Multiple layers allow network to learn complex models for increasingly complex tasks

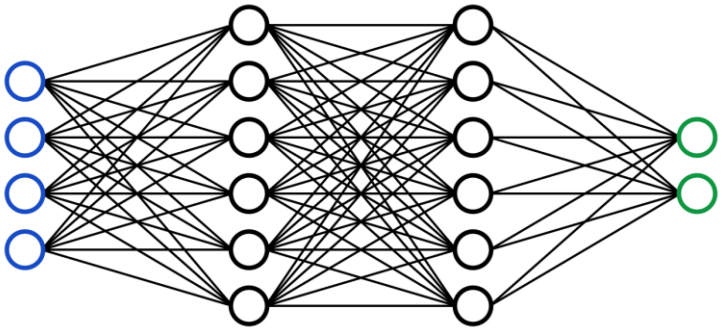


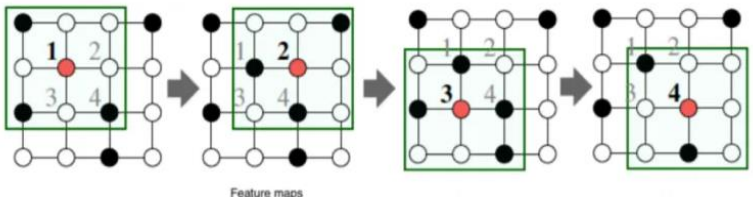
Fig. 1: Example fully-connected neural net

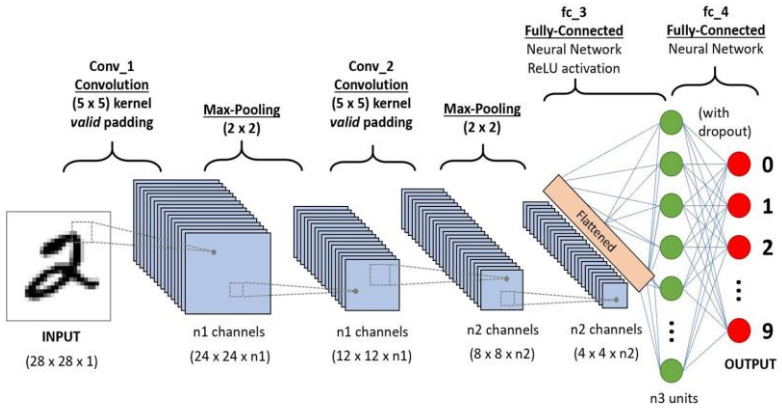
Supervised, Semi-Supervised, Unsupervised Learning

- Supervised learning - all training data is labeled
  - Classification
    - Image classification (CIFAR), character recognition (MNIST), any task where data is categorical. Output is a class
  - Regression
    - Predicting a cost given trends, predicting weather, tasks where data is continuous. Output is a single value
- Unsupervised learning - all training data is unlabeled
  - Clustering - can be used to identify patterns in data that is not overtly classified together
- Semi-supervised learning - mix of labeled and unlabeled data
  - Uses labeled examples to further correlate unlabeled data

Convolutional Neural Networks

- CNNs rose to prominence in 2012 (AlexNet)
- Particularly useful for data easily represented in 2D or 1D kernels
  - Images pixel by pixel, for example
- Convolutions identify the presence of local spatial features
  - Images of specific class often contain similar characteristics
- CNNs are superior at feature extraction with significantly fewer trainable parameters
  - Much more computationally efficient

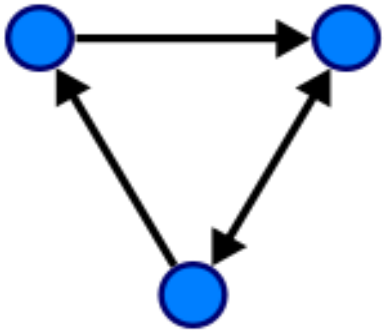


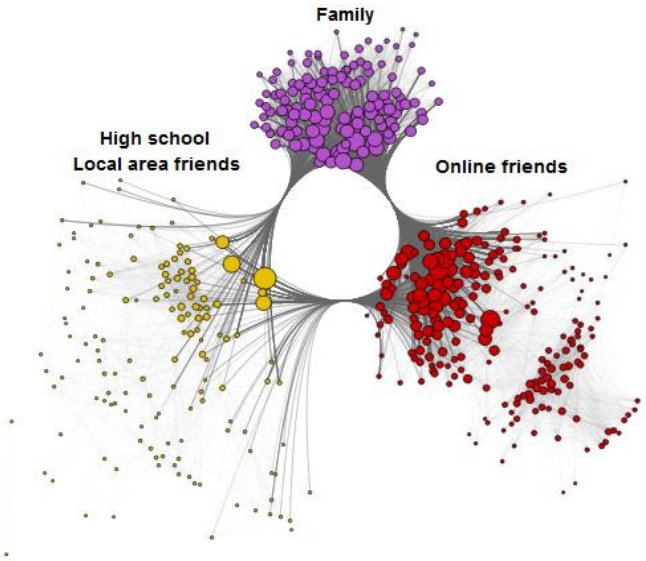


Source: [CNN introduction]

Data as Graphs

- Graphs, consisting of nodes and edges, represent data items and their relationships
- Ex: social network example
- Unlike images, graphs lack spacial localities
  - Two graphs representing identical information and relationships can be visualized and arranged in numerous ways
- Traditional CNNs thus struggle to generalize for inference tasks based on graph data

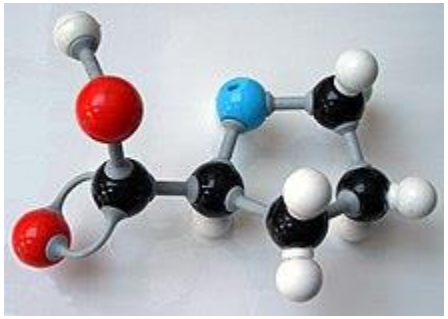


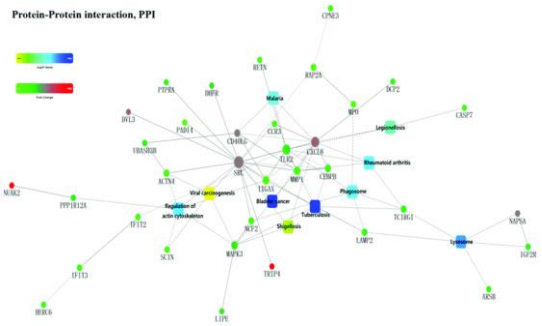


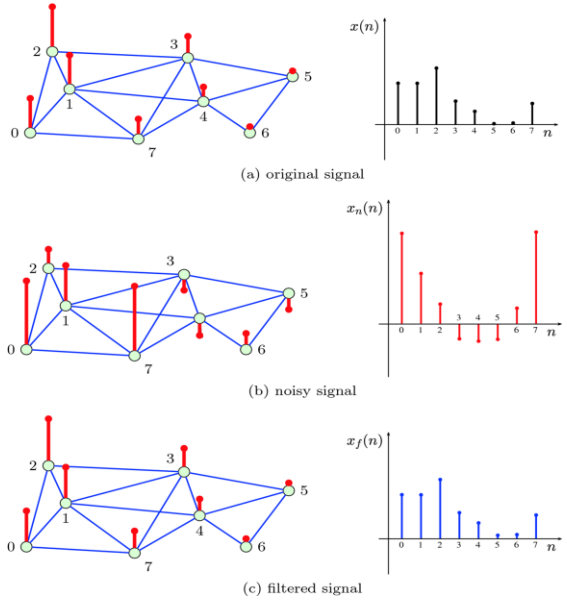
Source: link

Graph-based Learning Motivations

- Why might graph-based learning be desirable?
  - Inference tasks on large, complex sets of related data points. Example applications:
    - Molecular-level structures and interactions
    - Social networks
    - Knowledge graphs
    - Physical systems
    - Graph generation
    - Relationship extraction
    - Graph signal processin
    - …and more



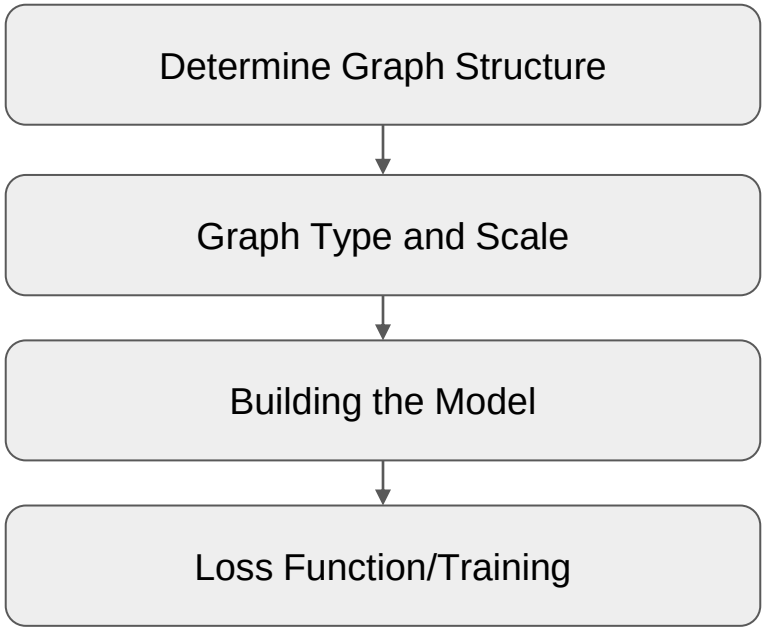




Graph Neural Networks

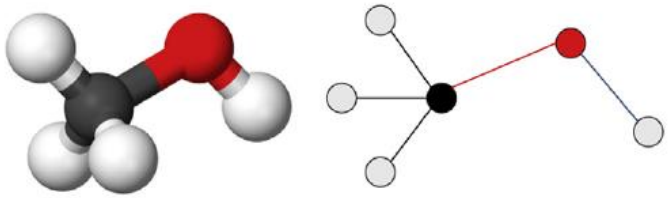
- GNNs represent a class of learning algorithms/architectures capable of performing supervised, semi-supervised, and unsupervised learning and inference on graph-represented data
- GNNs, as a generalization, extend CNN architectures - convolution, fully-connected, pooling layers
- Challenge: How do we structure GNNs to extract features, despite lack of spatial localities?

GNN Design Pipeline



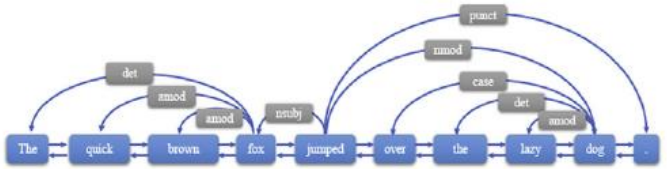
Graph Structure

- Structural
  - Data has inherent graph-like structure (e.g. molecules and social networks)



(b) Molecule

- Non-structural
  - Data does not have an explicitly graph-like structure and must be first translated into graphs before a GNN can be applied (e.g. text or images for image-recognition)

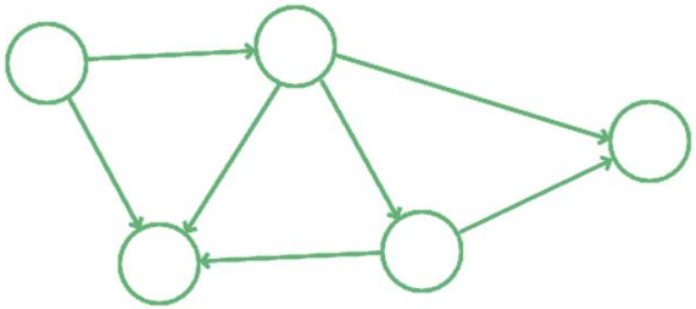


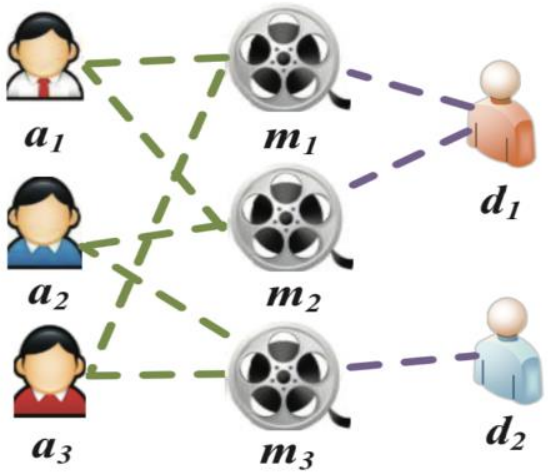
(d) Text

Graph Type and Scale

- Directed vs Undirected
  - Directionality of graph edges
  - Directed edges generally contain more information

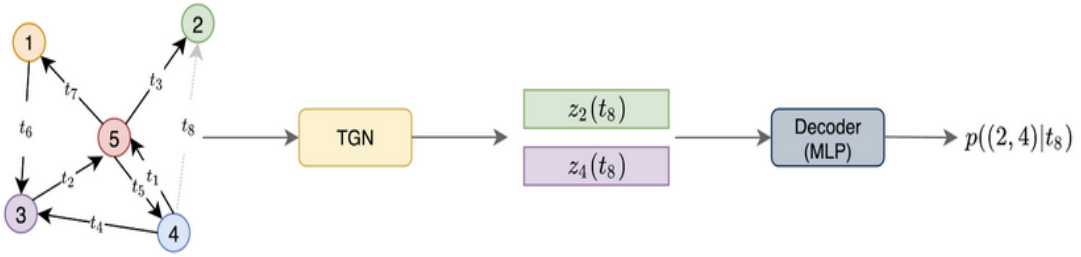
- Homogeneous/Heterogeneous
  - Homogenous
    - Edges and Nodes are of the same type
  - Heterogeneous
    - Edges and Nodes can be of varying types (example right)





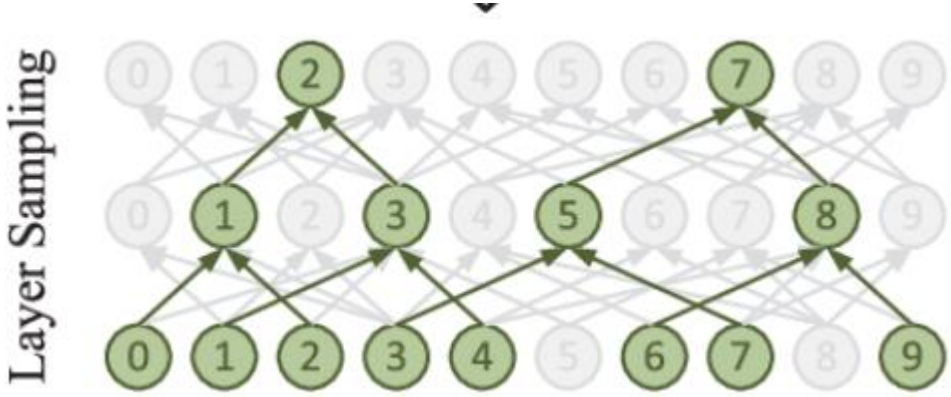
Graph Type and Scale

- Dynamic/Static
  - Topology/Input variance over time



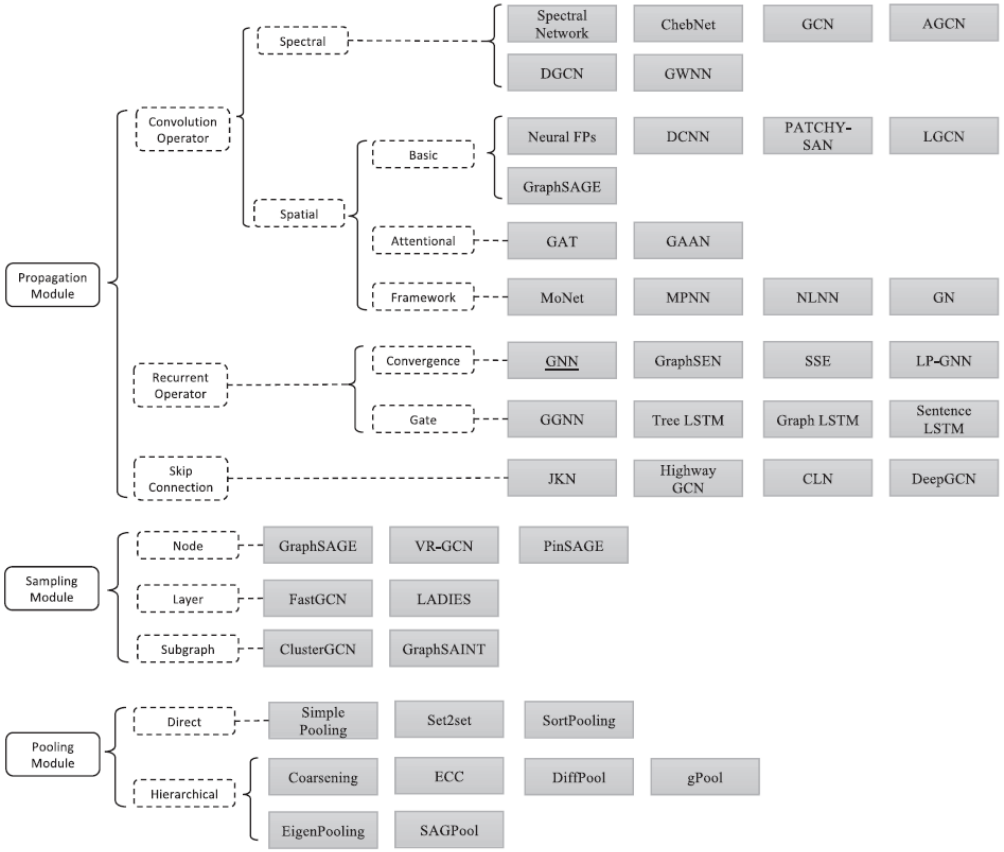
GNN predicting a new edge at a future timepoint

- Graph Scale
  - Graphs may be too large to compute node representations for every layer



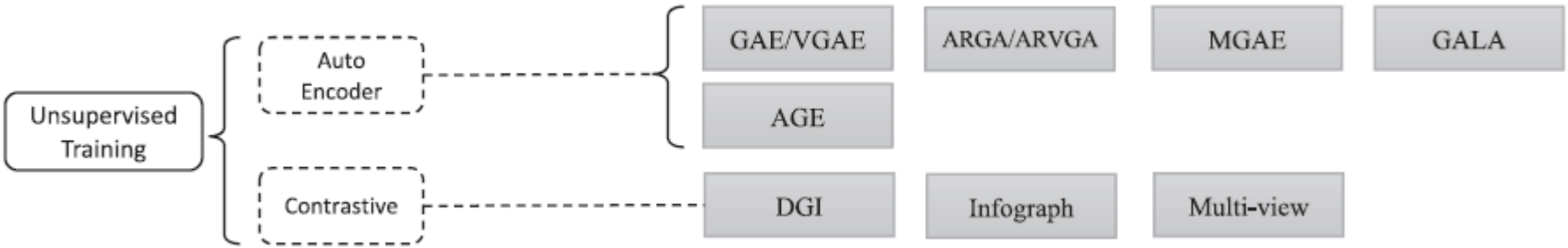
Building the Model

- Propagation
  - Passing messages between nodes, aggregating information from node neighborhood between levels of the GNN
- Sampling
  - Used for large graphs or deeper GNNs where “neighbor explosion” is an issue
- Pooling
  - Pulling information from nodes for higher-level representations

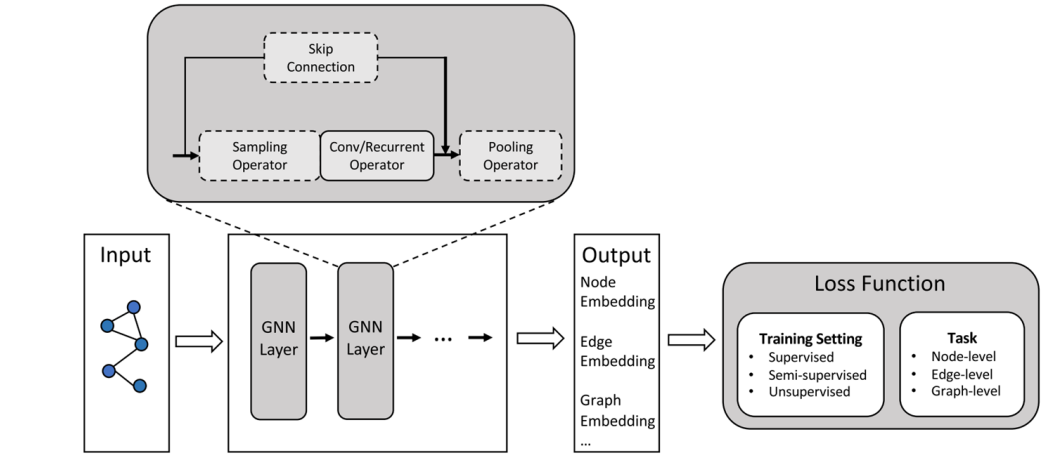


Loss Function/Training

- Learning Tasks
  - Graph level, edge level, node level
- Data Supervision
  - Supervised, Semi-supervised, and unsupervised



Architecture: Building Model By Computational Module



- Propagation Module
  - Propagate information between nodes so that aggregated information could capture feature and topological information
  - Neighbor: Use **Convolution Operator & Recurrent operator**
  - Historical Representation of nodes: **Skip Connection operation**
- Sampling Module
  - **Sampling Operator**
  - Conduct propagation on large graph
  - Combine with propagation module
- Pooling Module
  - **Pooling Operator**
  - Extract higher level information

Propagation Module

- Convolution operator (Mostly used)
  - Generalize convolution from other domain to this graph domain
  - Spectral approach: Methods based on graph signal processing
  - Spatial approach: define convolution directly on graph based on topology
- Recurrent Operator
  - Diff to Convolution: use same weights in different layers
  - Mostly for acyclic graph
- Skip Connection
  - Deeper model would result in no performance gain or worse
  - Noisy information would propagation, especially with exponentially grow of neighbor
  - Module added for GNN to go deeper

Sampling Module

- Mean:
  - GNN model needs information from each neighbor in previous layer → grow exponentially
  - Memory issue, computational complexity
- Node Sampling
  - Select subset from each node's neighbor
  - EX: GraphSAGE: sample a fixed small number of neighbor (2-50 neighborhood size)
  - EX: PinSage: importance based sampling method, random walks starting from target nodes, find highest normalized visit counts
- Layer Sampling
  - Matain small set of nodes from last layer
- Subgraph Sampling
  - Define multiple subgraphs, and restrain sampling within each subgraph
  - EX: ClusterGCN samples subgraph by graph clustering algorithm

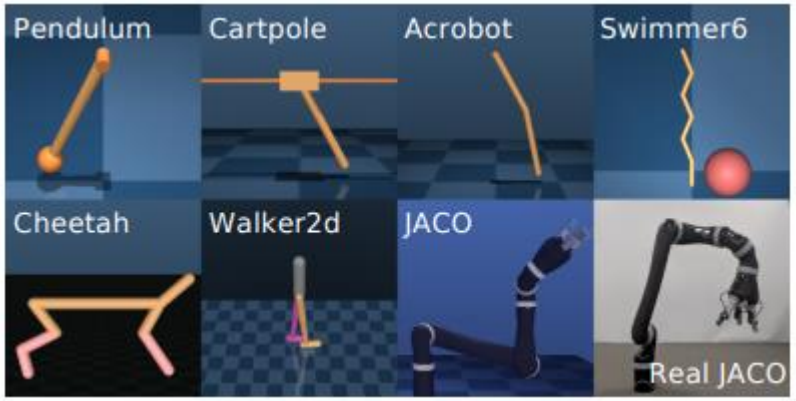
Pooling Module

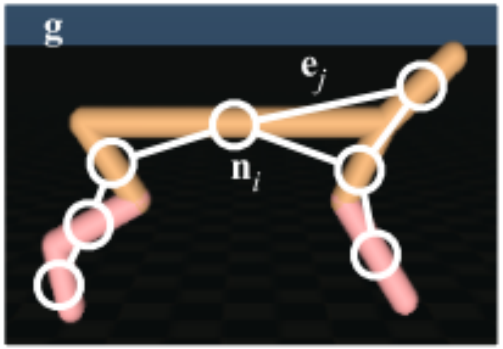
- Mean:
  - Pooling layer for more general features
- Direct pooling modules
  - Learn graph level representation directly from nodes with different node selection strategies.
  - EX: Simple Node Pooling: max/mean/sum/attention operation for global graph representation
  - EX: SortPooling: sorts the nodes embeddings according to the structural roles of the nodes and fed embedding to CNN
- Hierarchical pooling modules
  - Direct pooling method only pay attention to nodes itself
  - Hierarchical pooling methods would investigate the property of graph structure.
  - EX: gPool: use a project vector to learn projection scores for each node and select nodes with top-k scores.

Applications

Physics - Robotics

- Encode objects as nodes and edges as interactions to create physics simulation
- Ex: Using GNN to develop controls for robotic systems using graph based physic simulation
- Automates the search for control in a large search space

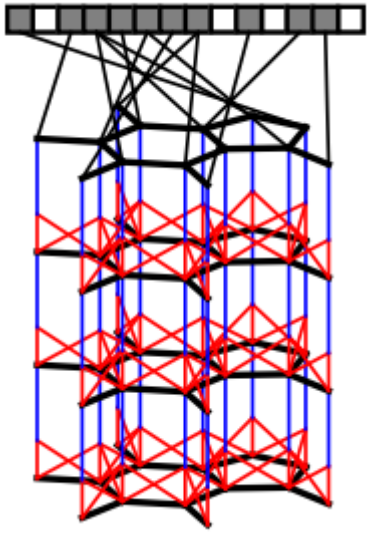




Graph Networks as Learnable Physics Engines for Inference and Control

Chemistry - Molecular Fingerprints

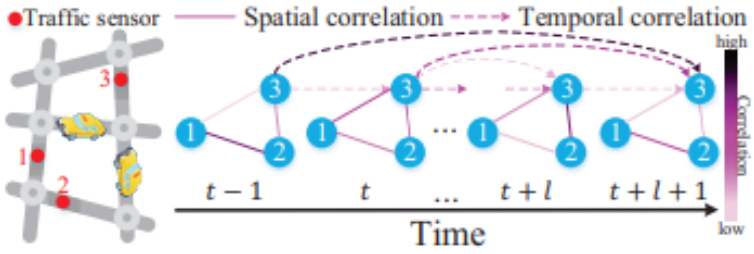
- Graphs encoding of structure of molecules
- Atoms are nodes and edges are chemical bonds
- Applying GNNs to molecular graphs can result in more accurate fingerprints
- Very important for the pharmaceutical industry to develop drugs



Convolutional Networks on Graphs for Learning Molecular Fingerprints

Traffic Networks

- Traffic networks are dynamic and have complex dependencies
- Optimizing traffic flow based on NN and GNN is an active area of research
- Useful for routing for navigation services and ride sharing services



(a) Xiamen (95 sensors) (b) PeMS (325 sensors)

GMAN: A Graph Multi-Attention Network for Traffic Prediction

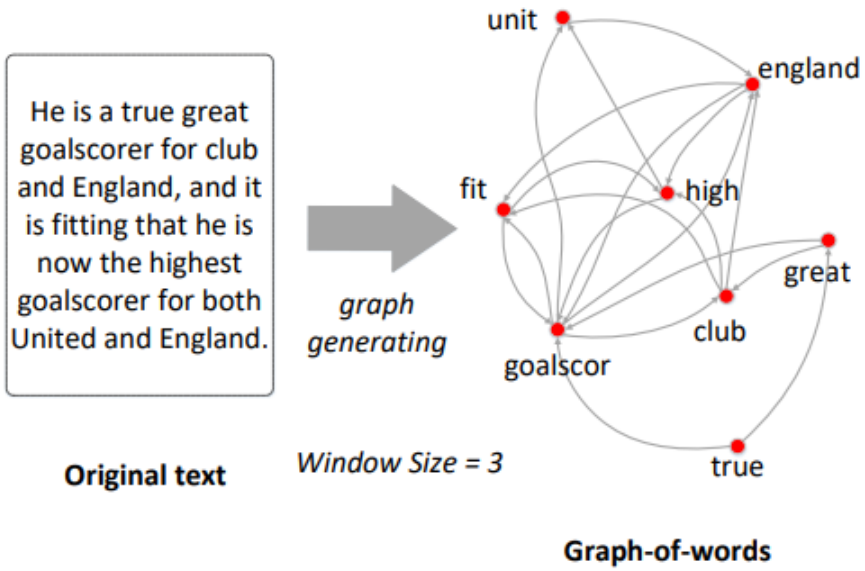
Computer Vision - Semantic Segmentation

- Traditional CNNs are successful at identifying key objects in large ROIs
- GNN techniques are suitable for classifying pixels
- Allows for much stricter boundaries on important objects in frame
- Potentially useful for self-driving cars

Semantic Object Parsing with Graph LSTM

Text - Natural Language Processing

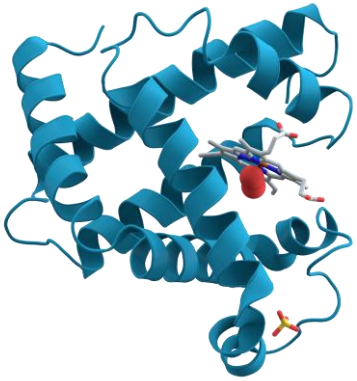
- GNNs are being used to gain more information from text
- Relationships can be drawn between words farther from each other than traditional NN methods
- Important in any application where a human needs to be understood by a machine

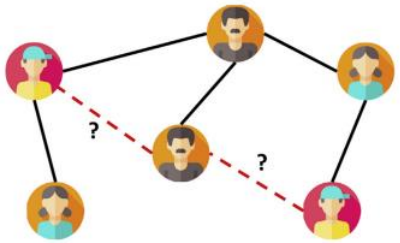


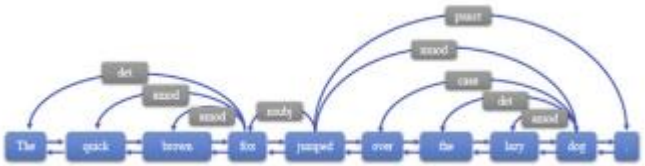
Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN

Other Applications

- Protein Interface Prediction
- Stock Market Prediction
- Social Networking Analysis
- Image Classification
- Text Prediction







Current Challenges in GNNs

- Generally lack robustness and resilience to adversarial attacks
- Still largely a “black box”
- Well-labeled and easily usable datasets are lacking compared to traditional machine learning

Conclusions

- GNNs offer a unique way of performing training and inference on graph-based data
- Unlocks machine learning to problems not easily represented in 2D kernel or 1D space
- Still fairly novel compared to CNNs and are not easily interpreted
- Far-reaching in various scientific fields

GNNs / References

- Graph neural networks: A review of methods and applications. https://arxiv.org/pdf/1812.08434.pdf
- https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications
- https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
- https://distill.pub/2021/gnn-intro/