

ELEC 677:  
Introduction to  
Deep Machine Learning  
Lecture 2

**Ankit B. Patel**

*Baylor College of Medicine (Neuroscience Dept.)*

*Rice University (ECE Dept.)*

8-30-2016

# More Logistics

- We've increased the limit in the class to let in wait-listers: 32 enrolled plus 5 on waitlist currently
- Check out website: [elec677.rice.edu](http://elec677.rice.edu)
- Please join the Piazza Discussion Forum (go to website and click on Discussion Forum Tab)
- Assignment #0 will be available Sep 6
- Assignment #1 will be available Sep 13

# More on Final Project

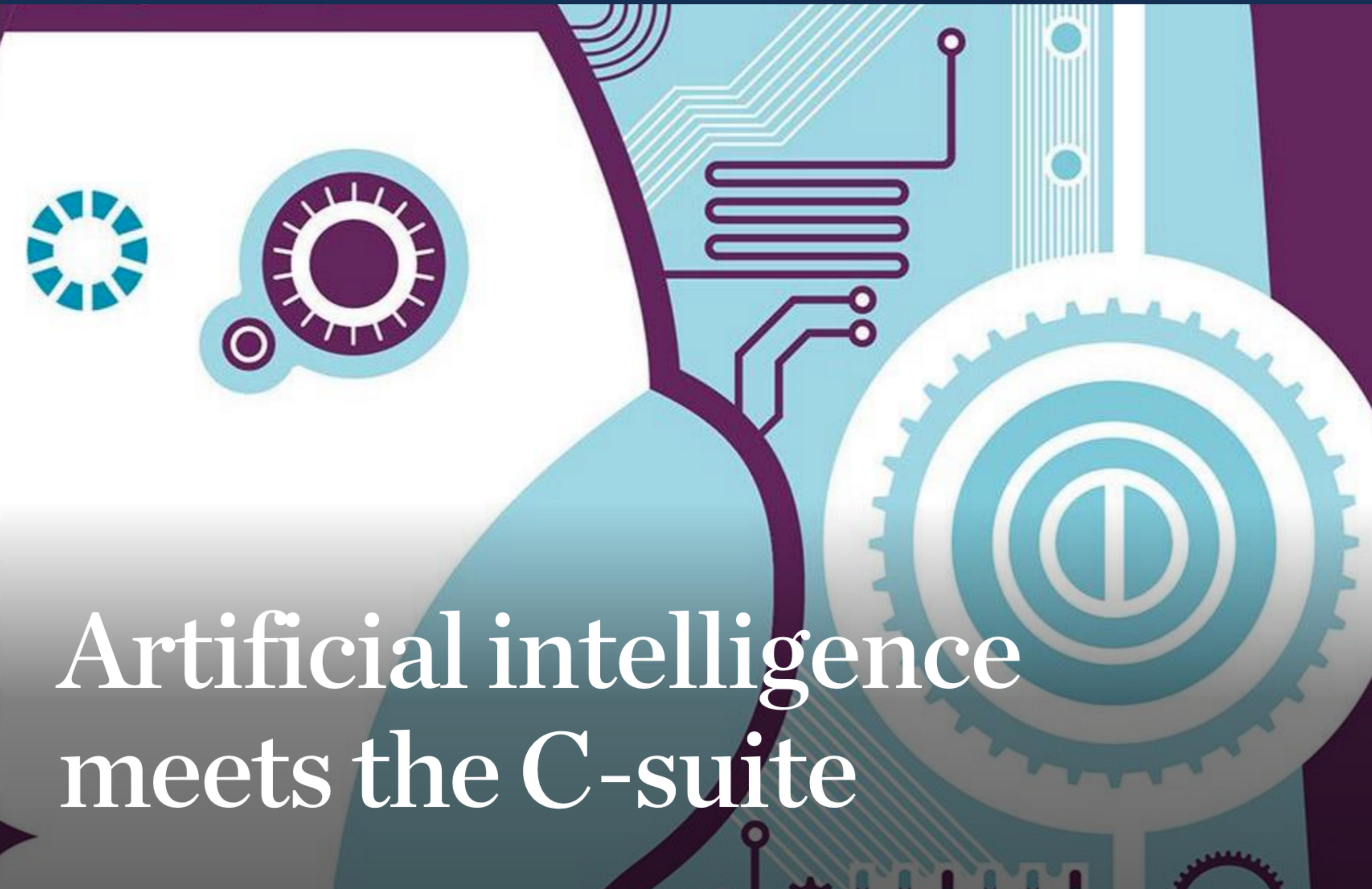
- Proposal due October 18th
- Up to two people per project: we will have a **mandatory mixer from 4:15 - 5pm** to help you all get to know each other and me
- Can combine across multiple classes if relevant. Please e-mail instructors of both courses to discuss.
- **Optional (but highly encouraged):** Choose final project topic relevant to your personal research/field. Please e-mail to discuss.

# Recent Trends + Latest News

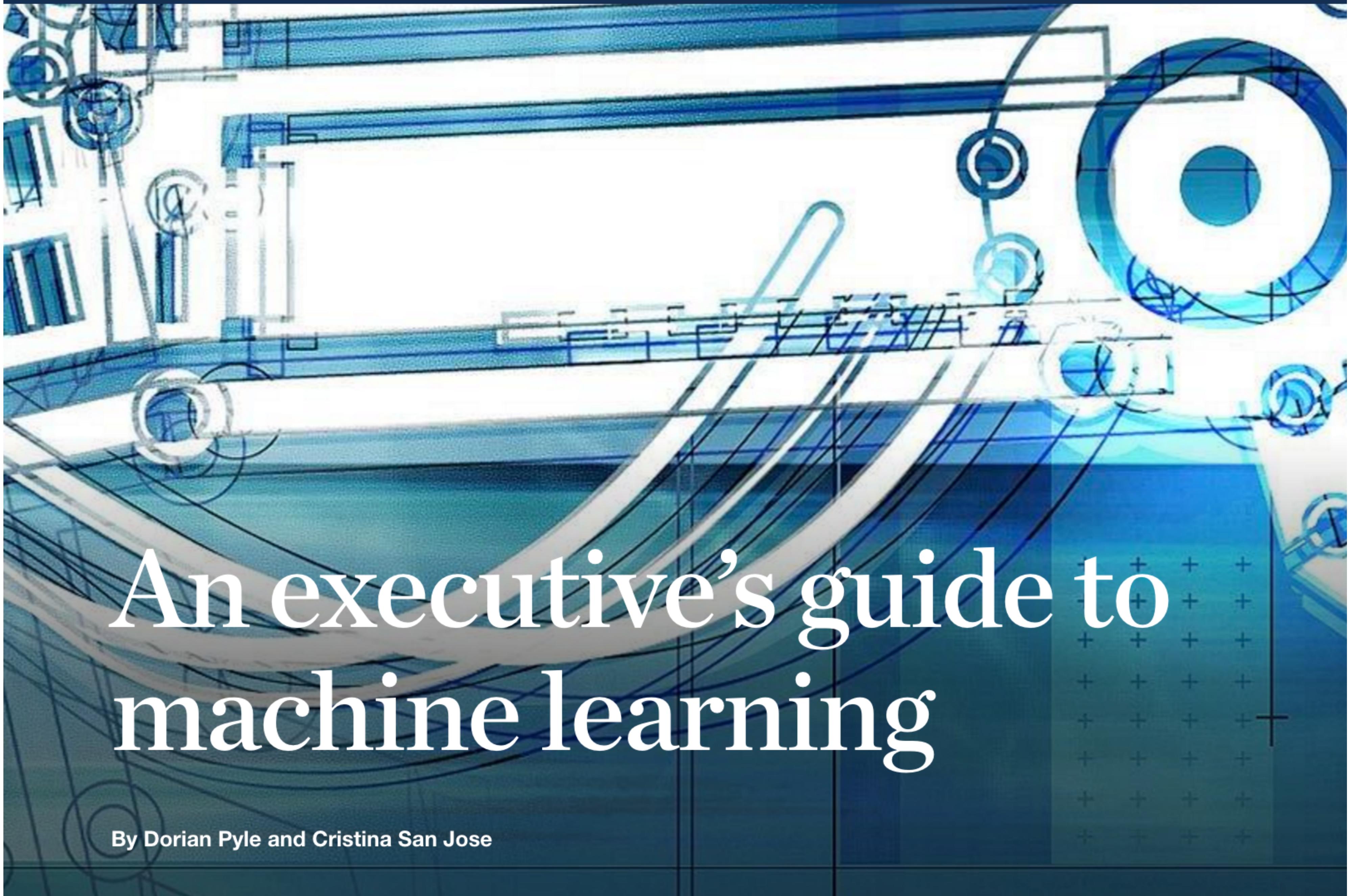
# HOW GOOGLE IS REMAKING ITSELF AS A "MACHINE LEARNING FIRST" COMPANY

If you want to build artificial intelligence into every product, you better retrain your army of coders. Check.

**F** "THE MACHINE LEARNING MODEL FEELS LIKE A  
LIVING, BREATHING THING," SAYS CHRISTINE  
ROBSON. "IT'S A DIFFERENT KIND OF  
ENGINEERING."



# Artificial intelligence meets the C-suite



# An executive's guide to machine learning

By Dorian Pyle and Cristina San Jose

# McKinsey: \$50 trillion of value to be created by AI and Robotics through 2025

by *omohundro*

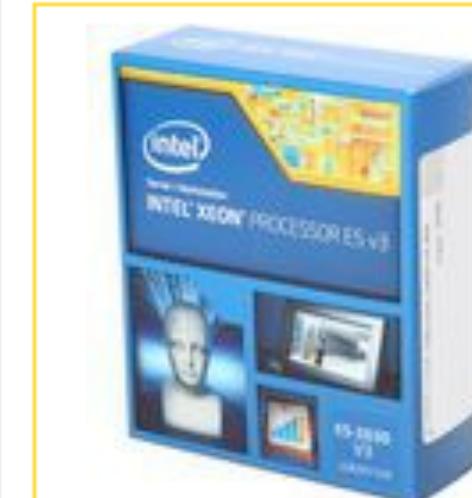
To better understand the likely social impact of AI and Robotics, it's very useful to have an estimate of the economic gains they will create in the near future. The respected consulting firm McKinsey & Company recently released the report: "[Disruptive technologies: Advances that will transform life, business, and the global economy](#)". The report estimates the [likely economic impact of 12 disruptive technologies](#) ten years from now in 2025.

To get a better sense of the scale of the forces involved, I wanted a single number that would summarize the economic impact of just AI and Robotics over the next 10 years. I took the 5 technologies that could be considered "AI and Robotics" and their ranges in "trillions of dollars of impact annually":

- Automation of knowledge work: \$5.2-6.7 trillion
- Internet of things: \$2.7-6.2 trillion
- Advanced robotics: \$1.7-4.5 trillion
- Autonomous and near-autonomous vehicles: \$.2-1.9 trillion
- 3D printing: \$.2-.6 trillion

# Drive.ai to test 'deep learning' autonomous cars on California roads

By [Jacob Kastrenakes](#) on April 23, 2016 05:00 pm [Email](#) [@jake\\_k](#)



Intel Xe  
2.4 GHz  
85W B)

Intel Xe  
Haswell

**\$684.9**

[BUY NOW](#)



Intel Xe  
2.4 GHz  
BX8066

Intel Xe  
2.4 GHz

**\$977.3**

[BUY NOW](#)



SUPER  
X10DRi  
Enhanc

SUPER  
X10DRi

**\$914.9**

[BUY NOW](#)



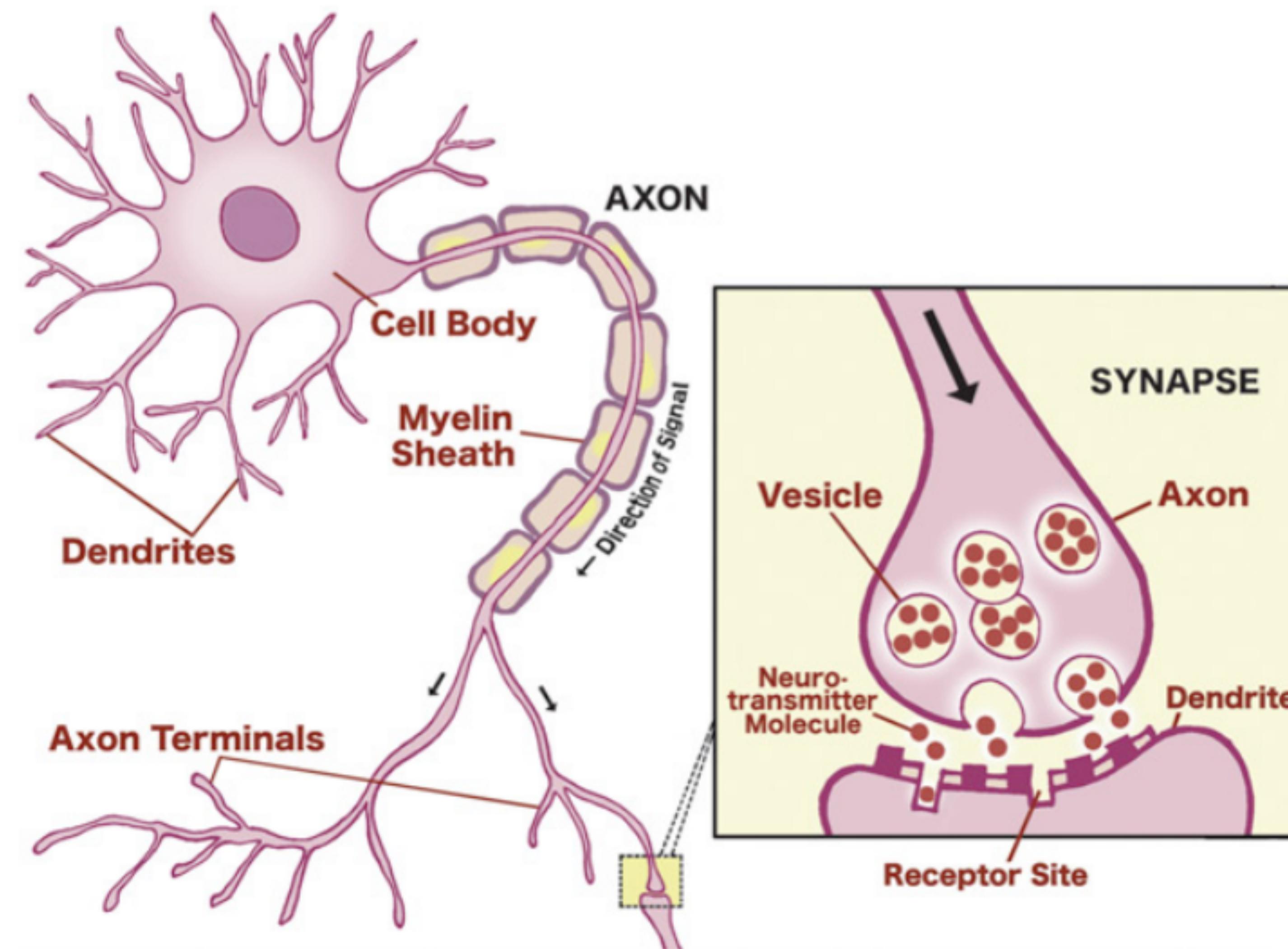
Intel Xe

# A Brief History of Neural Networks

# History

Time	Event
1943	McCulloch + Pitts introduce a simplified mathematical model of neurons as computing ANDs and ORs: capable of basic Logic but no Learning yet. They prove theorems about <b>expressive power</b> .
1949	Hebb introduces the first neurobiological <b>learning rule</b> : <i>“Neurons that fire together, wire together.”</i>
1950s	Computers are in their infancy. <i>MADALINE</i> : The <u>first NN for commercial use</u> : designed for adaptive filtering of echoes in phone lines (1959). <u>Still in use today!!</u>
1962	Rosenblatt introduces the <b>Perceptron</b> . It <u>learns</u> ! Hubel + Wiesel describe simple and complex cells in visual area V1 (inspiration for later NNs: S--> <b>template matching</b> for pattern specificity and C--> <b>pooling</b> for robustness to nuisances).
1969-1981	Minsky and Papert deal “deathblow” to Perceptrons by showing they can’t learn XOR. Later, others showed this wasn’t true for deep NNs. “NN Winter”: excessive hype and outrageous claims lead to dying interest/funding.

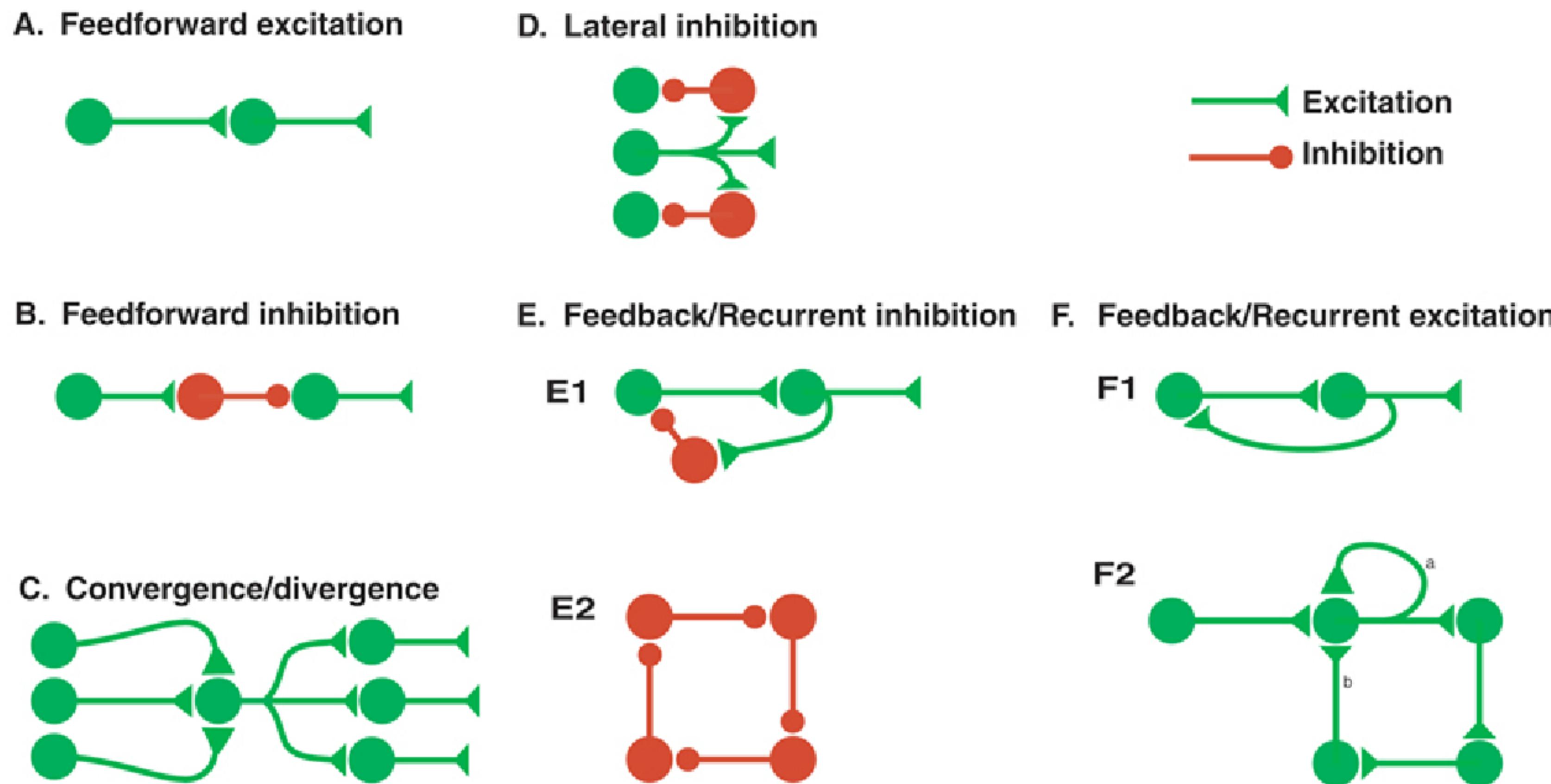
# Neuron



# How do neurons communicate?

- Animations of excitatory and inhibitory neuron:
- <http://neuroscience.uth.tmc.edu/s1/introduction.html>

# Final Project Idea: MicroNetwork Motifs



# LOGICAL CALCULUS OF IDEAS IMMANENT IN NERVOUS ACTIVITY

Warren S. McCulloch Walter H. Pitts

Presented by: Minh Nguyen

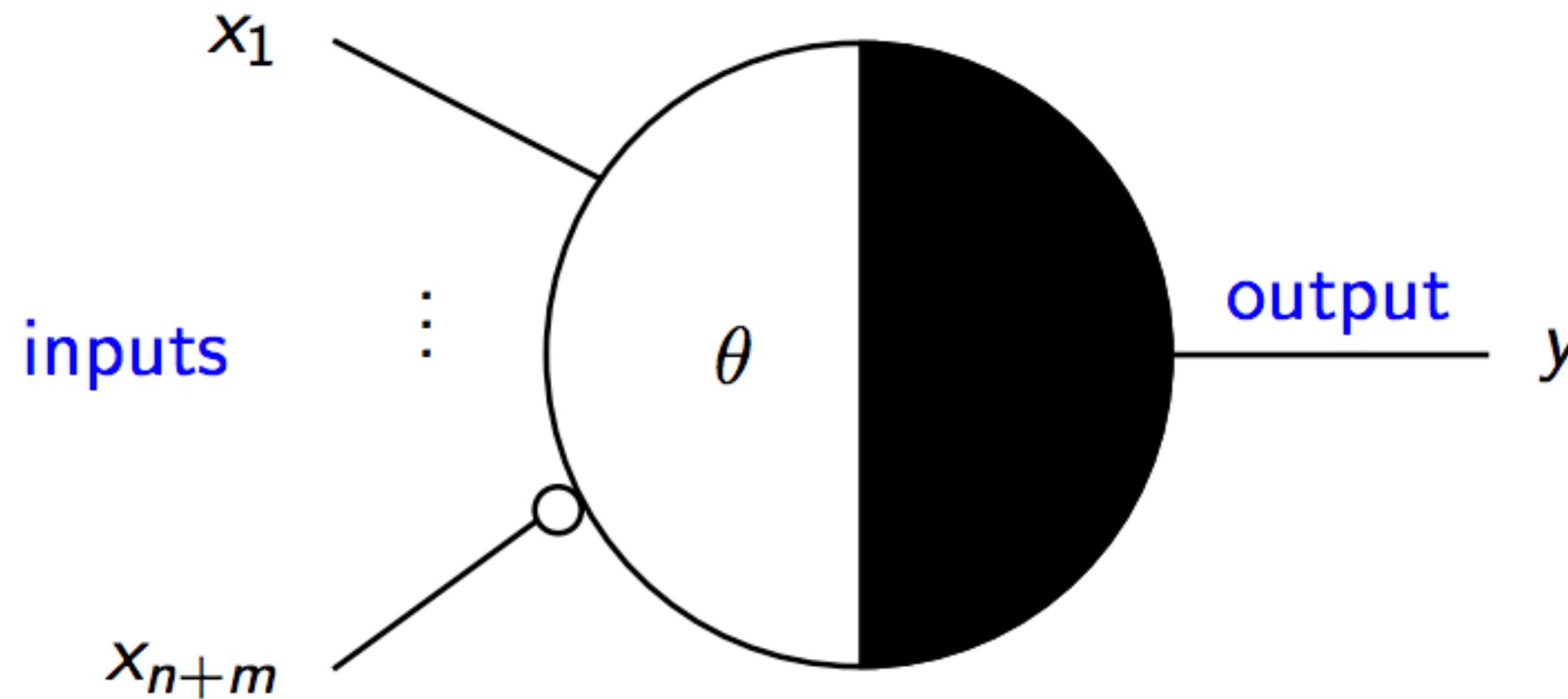
# Why is it important?

- A formalism whose refinement and generalization led to the notion of “finite automata”
- A technique that inspired the notion of logic design
- The first use of computation to address the mind-body problem
- The first modern computational theory of mind and brain

# Assumptions

1. The activity of the neuron is an “all-or-none” process.
2. A certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position of the neuron.
3. The only significant delay within the nervous system is synaptic delay.
4. The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time.
5. The structure of the net does not change with time.

# McCulloch-Pitts Neurons



Stefan Droste

- all signals binary ( $\in \{0, 1\}$ )
- threshold  $\theta \in \mathbb{N}_0$
- $n$  excitatory inputs  $x_1, \dots, x_n$
- $m$  inhibitory inputs  $x_{n+1}, \dots, x_{n+m}$
- one output  $y$  (with unrestricted fan-out)

$$y(x_1, \dots, x_{n+m}, \theta) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq \theta \text{ and } \sum_{i=n+1}^m x_i = 0 \\ 0 & \text{if } \sum_{i=1}^n x_i < \theta \text{ or } \sum_{i=n+1}^m x_i > 0 \end{cases}$$

# McCulloch-Pitts Nets

A McCulloch-Pitts Net is a directed graph  $G$  with McCulloch-Pitts neurons as nodes and edges marked as either excitatory or inhibitory.

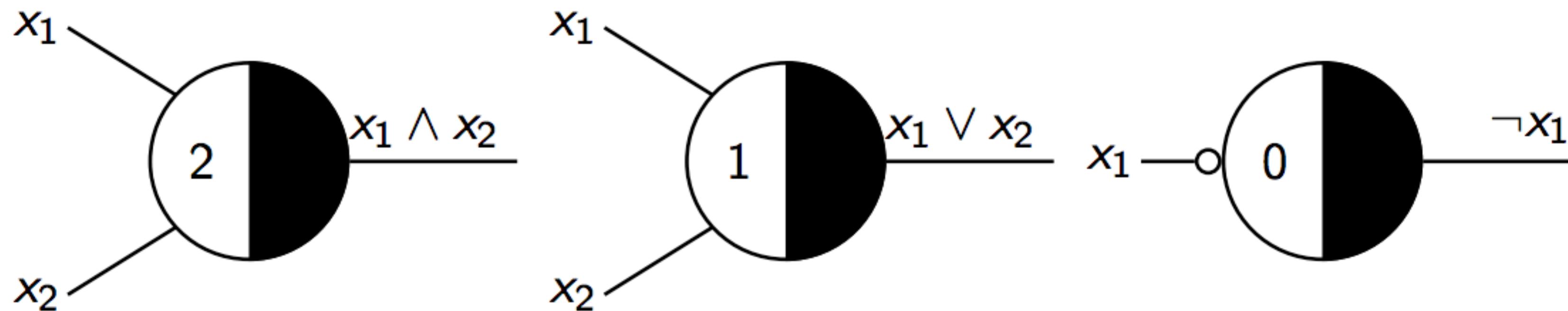
If  $G$  is acyclic it is called a feed-forward net.  
Otherwise,  $G$  is called recursive net.

Stefan Droste

In recursive McCulloch-Pitts nets, each neuron computes an output in 1 time step.

# Expressive Power of McCulloch-Pitts Nets

Feed-forward McCulloch-Pitts nets can compute any Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ .



Stefan Droste

Recursive McCulloch-Pitts nets can simulate any deterministic finite automaton (DFA).

## **Interesting Question:**

What are the problems/limitations of  
expressive power? (1-2 min)

## **Interesting Question:**

What are the problems/limitations of expressive power? (1-2 min)

## **Answer:**

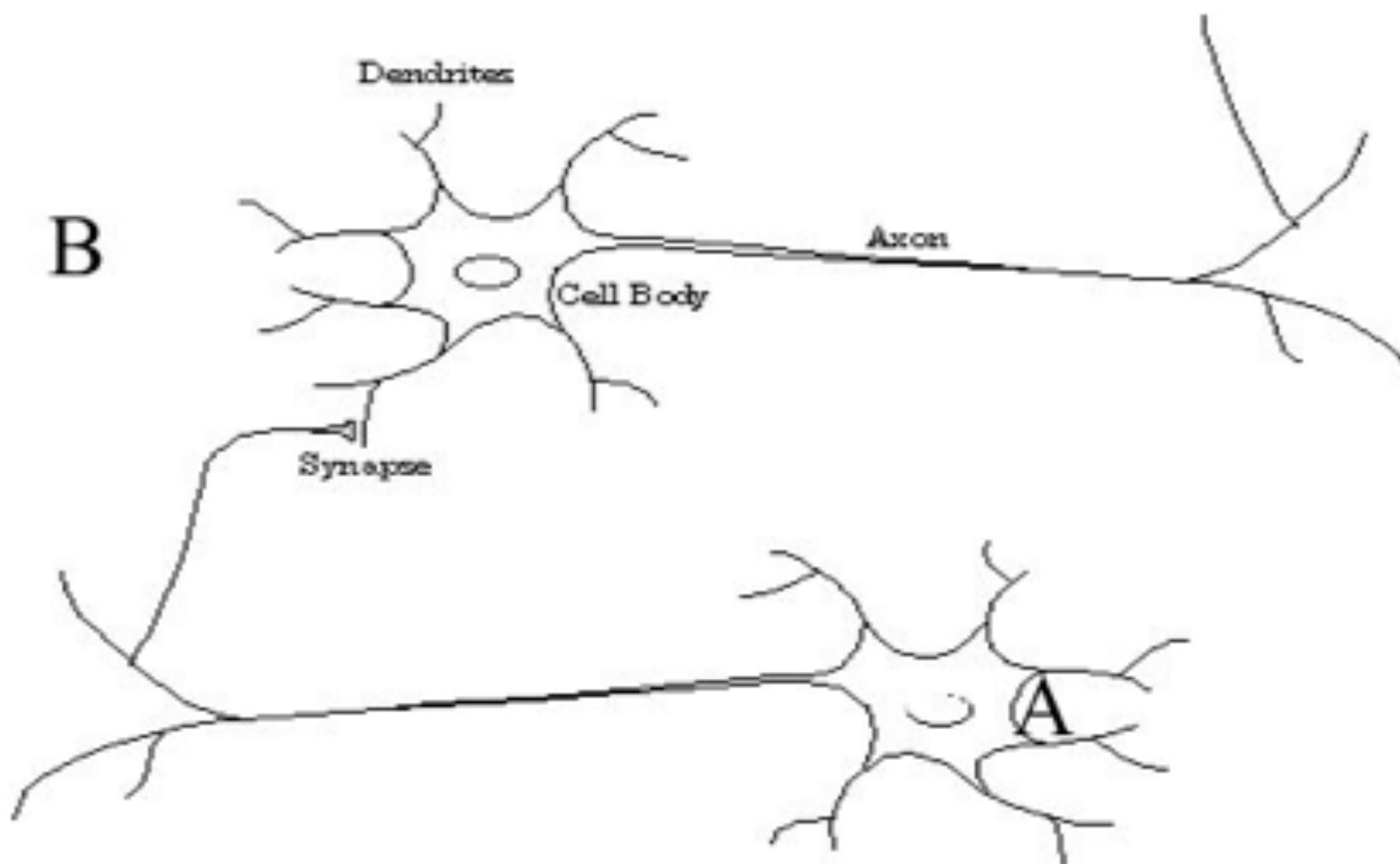
Expressability does not imply Learnability

# Ideas

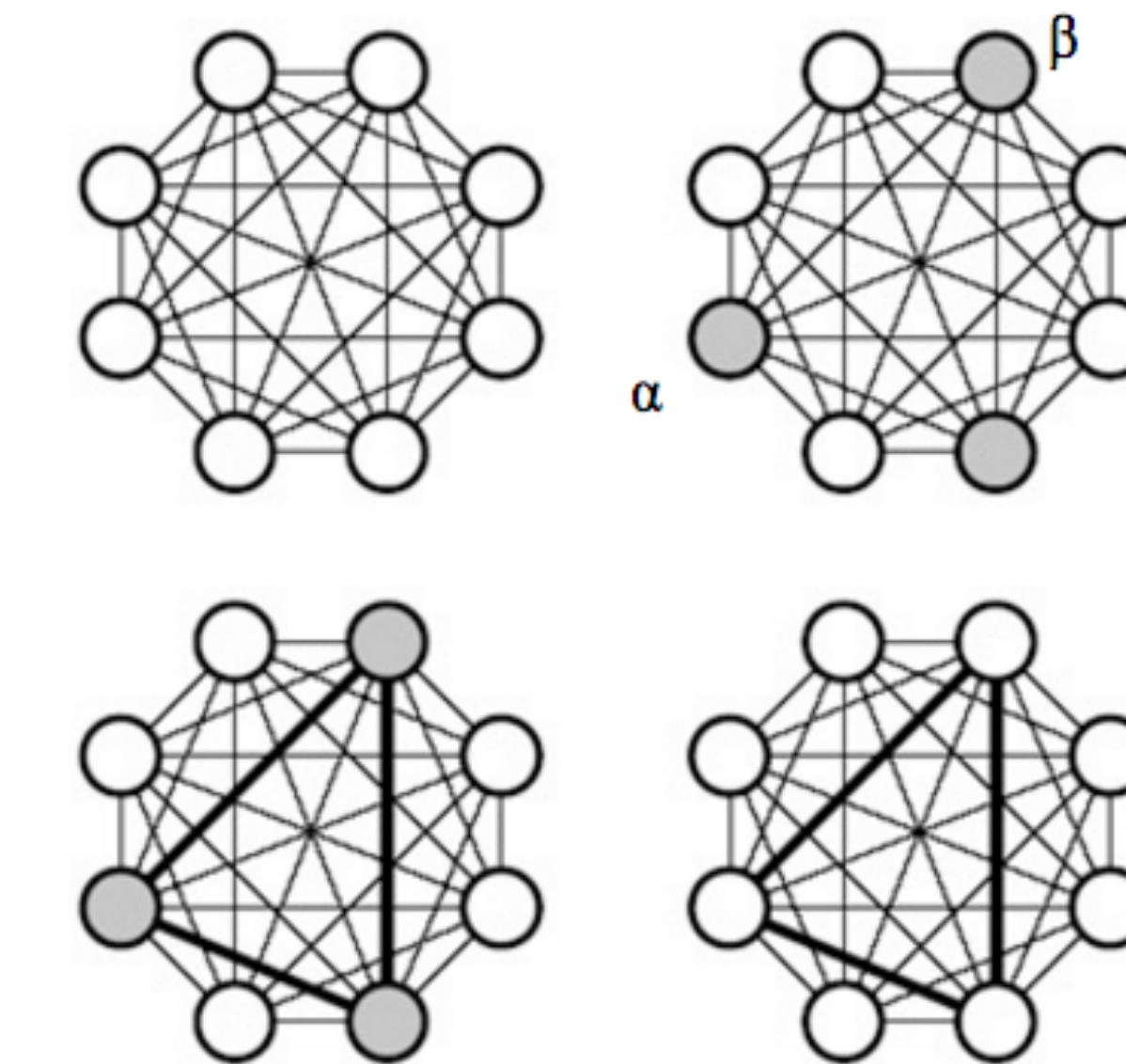
- The activities of any neural <-> Bits (True or False)
- Physiological relations among nervous activities <->  $f(\text{Bits})$
- The utility of the representation <-> how to play around with  $f(\text{Bits})$

# How can Neuron's learn? Hebb's postulate

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.”



D. O. Hebb, 1949

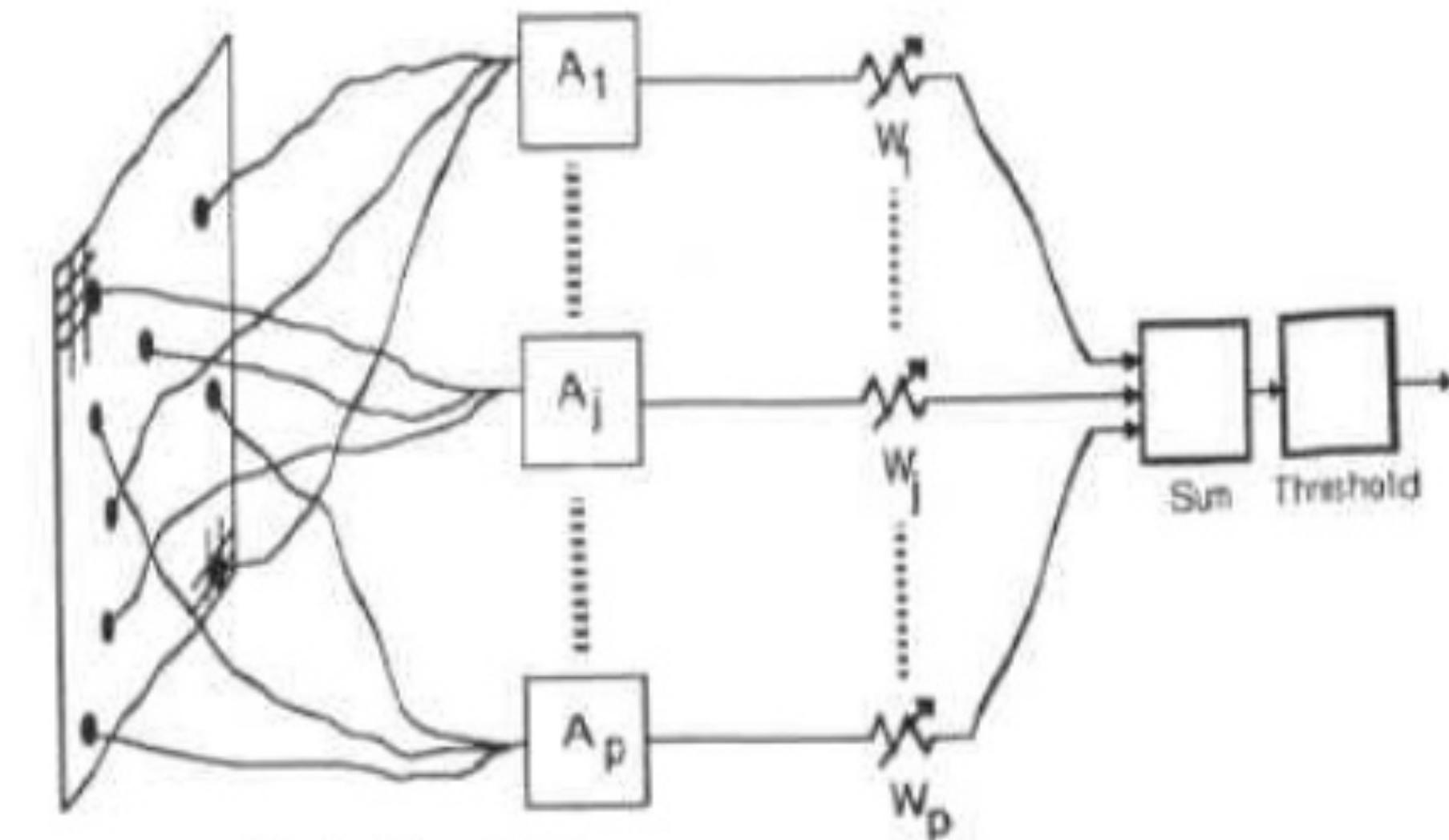


# How do neurons process visual input?

- Hubel and Wiesel's incredible discovery (1962):
  - <https://www.youtube.com/watch?v=IOHayh06LJ4>
- Awarded Nobel Prize in Physiology and Medicine

# The Perceptron (Rosenblatt)

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{else} \end{cases}$$



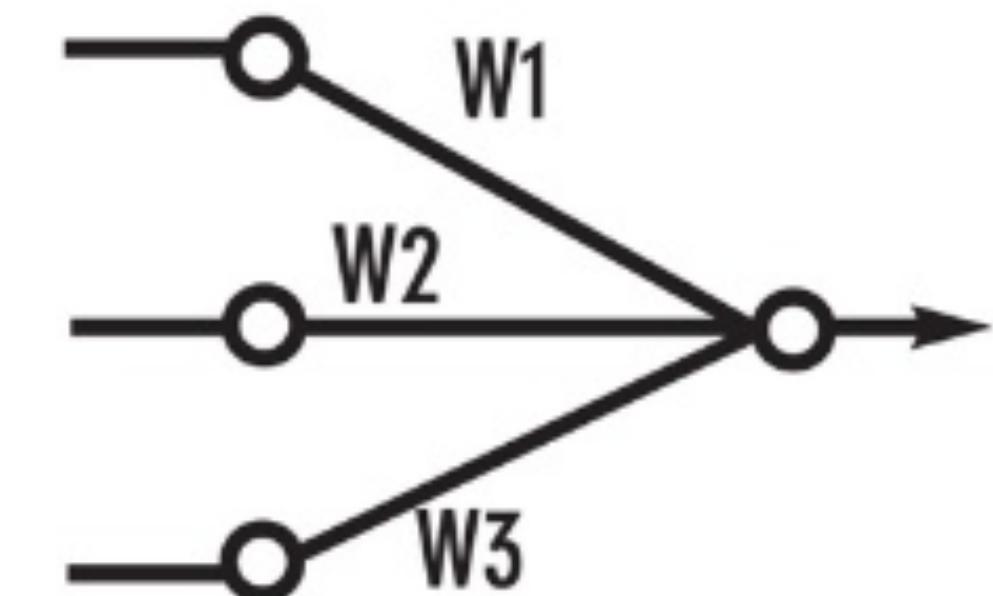
Original Perceptron

(From *Perceptrons* by M. L. Minsky and S. Papert, 1969, Cambridge, MA: MIT Press. Copyright 1969 by MIT Press.



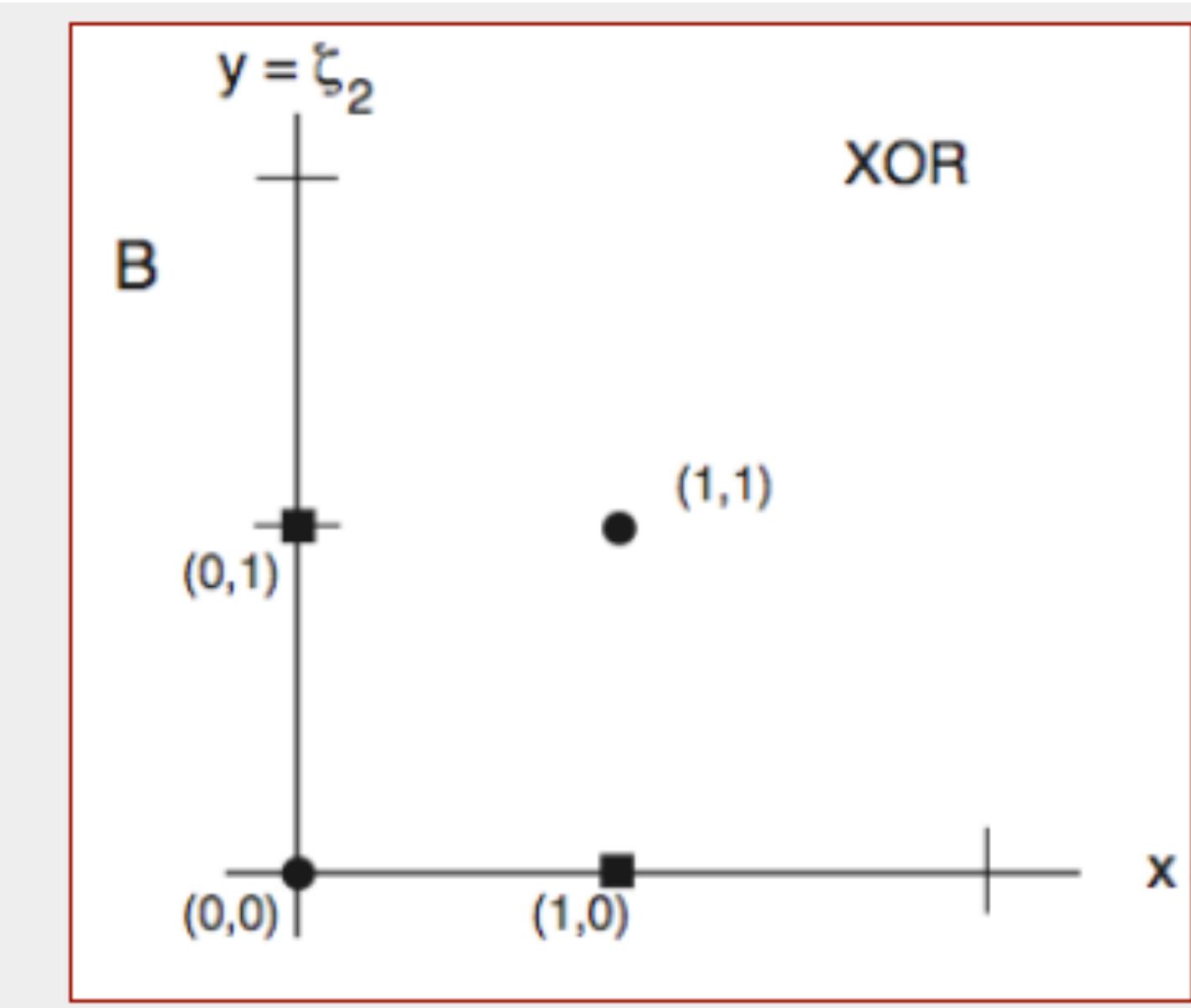
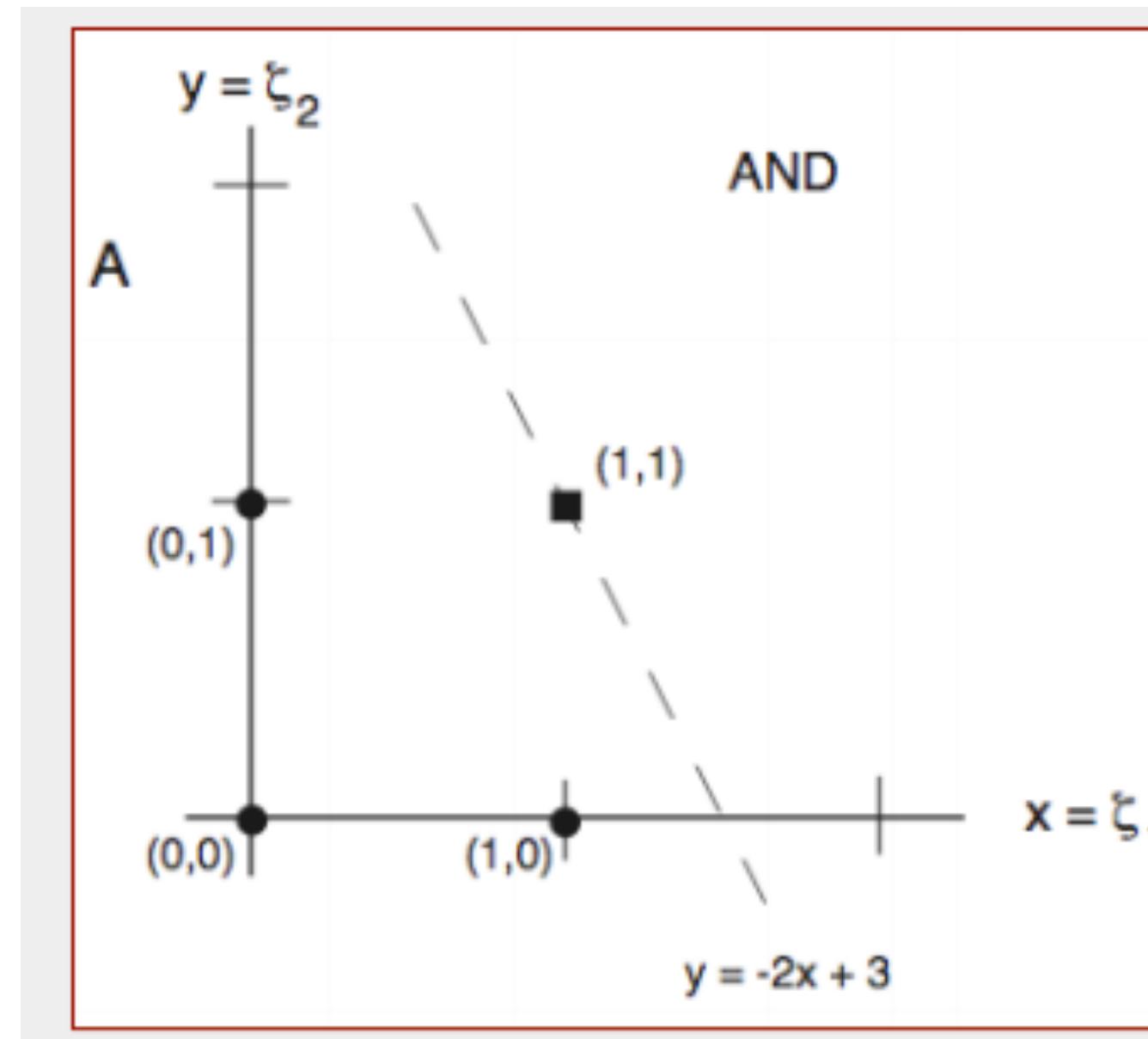
Frank Rosenblatt  
(1928-1971)

Simplified model:



# Minsky & Papert Deal a “Deathblow” to the Perceptron: The XOR Problem

Linear Separable or Not?



## **Interesting Question:**

How can you prove that the XOR Problem  
is not linearly separable? (4 min)

# Answer: Proof by Contradiction

$$w_1x + w_2y = w_10 + w_20 < \theta$$

$$w_1x + w_2y = w_10 + w_21 \geq \theta$$

$$w_1x + w_2y = w_11 + w_20 \geq \theta$$

Greater than zero.

$$w_1x + w_2y = w_11 + w_21 < \theta$$

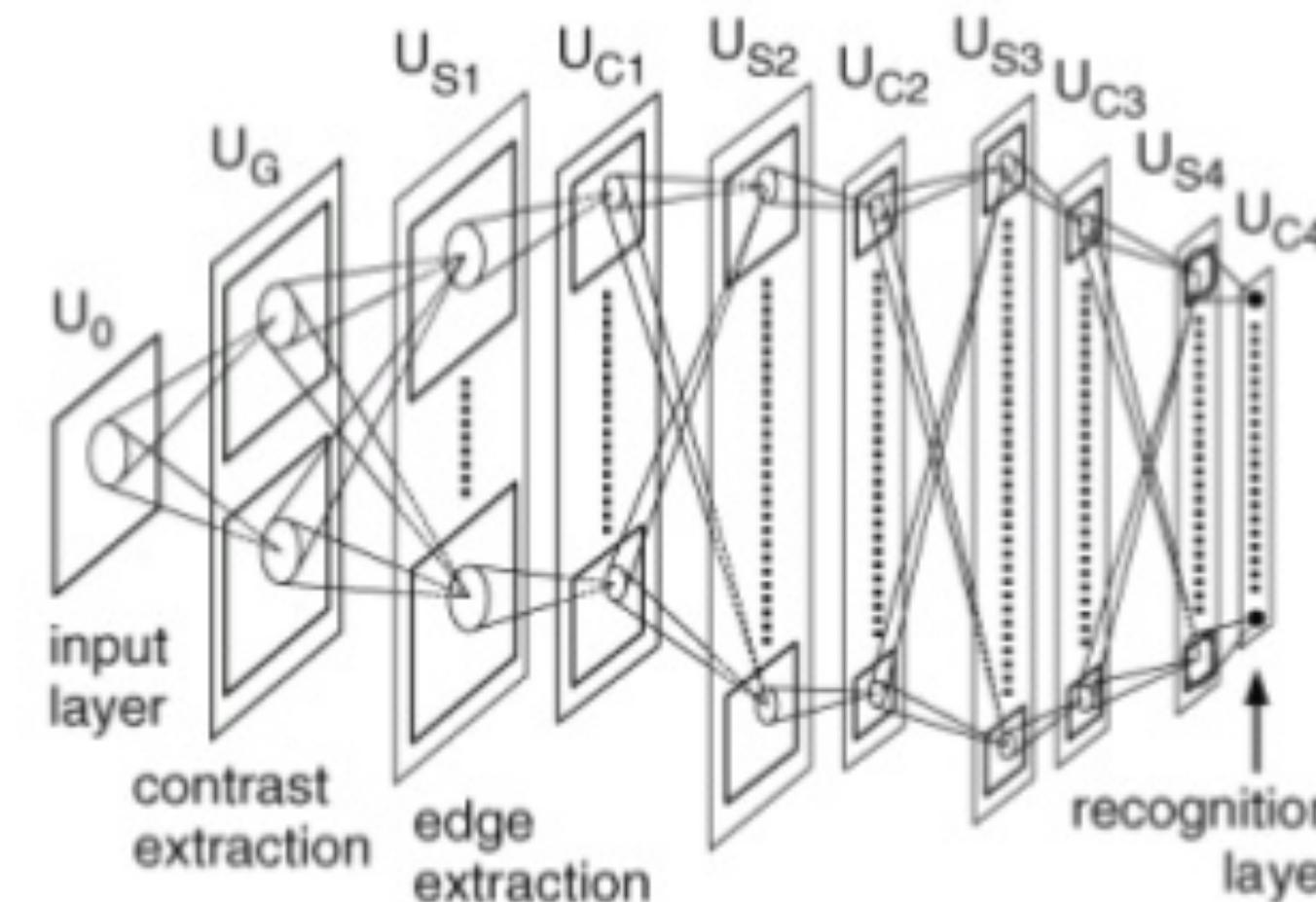
Impossible, given above.

# History

1970 +- 10 years	<b>Backpropagation</b> (Gradient Descent for nested functions via Chain Rule) explored in various contexts (both NN-specific and general).
1979	Fukushima introduces the <i>Neocognitron</i> . It foreshadows current deep NNs: <b>convolutional layers</b> , <b>weight replication</b> , and <b>WTA-subsampling</b> . However its <u>unsupervised</u> .
1982	<b>Hopfield Nets</b> : RNNs with binary units that can serve as an content-addressable associative memory (given a partial/wrong pattern, it will complete/correct it to the most similar memory). Revival of interest ensued.
1986	Werbos, Rumelhart + Hinton Williams <u>revive</u> Backprop for NNs. Hinton and Smolensky introduce <b>(Restricted) Boltzmann machines (RBMs)</b> .
1989	LeCun applies Backprop to Fukushima's Neocognitron to do supervised learning. This is the first incarnation of modern <b>convolutional neural nets (CNNs)</b> . Subsequently used by US Post Office for address reading.
1990	Baird introduces the idea of <b>data augmentation</b> : adding transformed/deformed versions of original input data to increase size of training dataset.
1991-5	Fundamental Problem of DL: <b>vanishing/exploding gradients</b> plague deep FFNNs and RNNs. Hochreiter + Schmidhuber introduce LSTM RNNs (1995-7) to address this. LSTMs are critical to solving problems requiring a real memory (e.g. widely spaced events in time are important.)

# Neocognitron: Precursor to Modern Convnets

Fukushima (1980). Hierarchical multilayered neural network

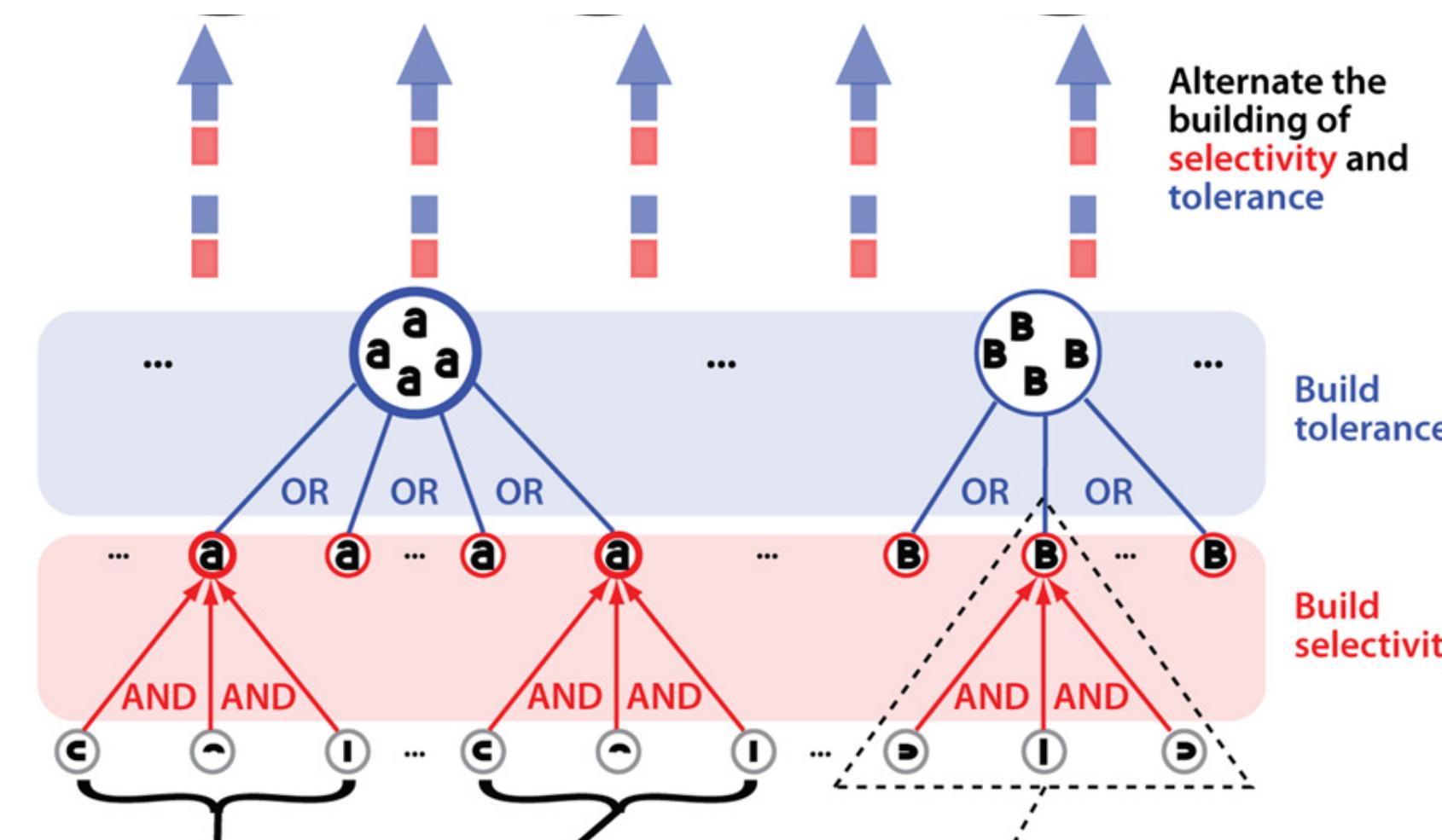


**S-cells** work as feature-extracting cells. They resemble simple cells of the primary visual cortex in their response.

**C-cells**, which resembles complex cells in the visual cortex, are inserted in the network to allow for positional errors in the features of the stimulus. The input connections of C-cells, which come from S-cells of the preceding layer, are fixed and invariable. Each C-cell receives excitatory input connections from a group of S-cells that extract the same feature, but from slightly different positions. The C-cell responds if at least one of these S-cells yield an output.

# Key Idea: Alternating Selectivity and Invariance

- ▶ Hubel and Wiesel's discovery of simple/complex cells and their special properties of *selectivity and tolerance/invariance*



DiCarlo, J. J. et al. How does the brain solve visual object recognition? *Neuron* (2012).

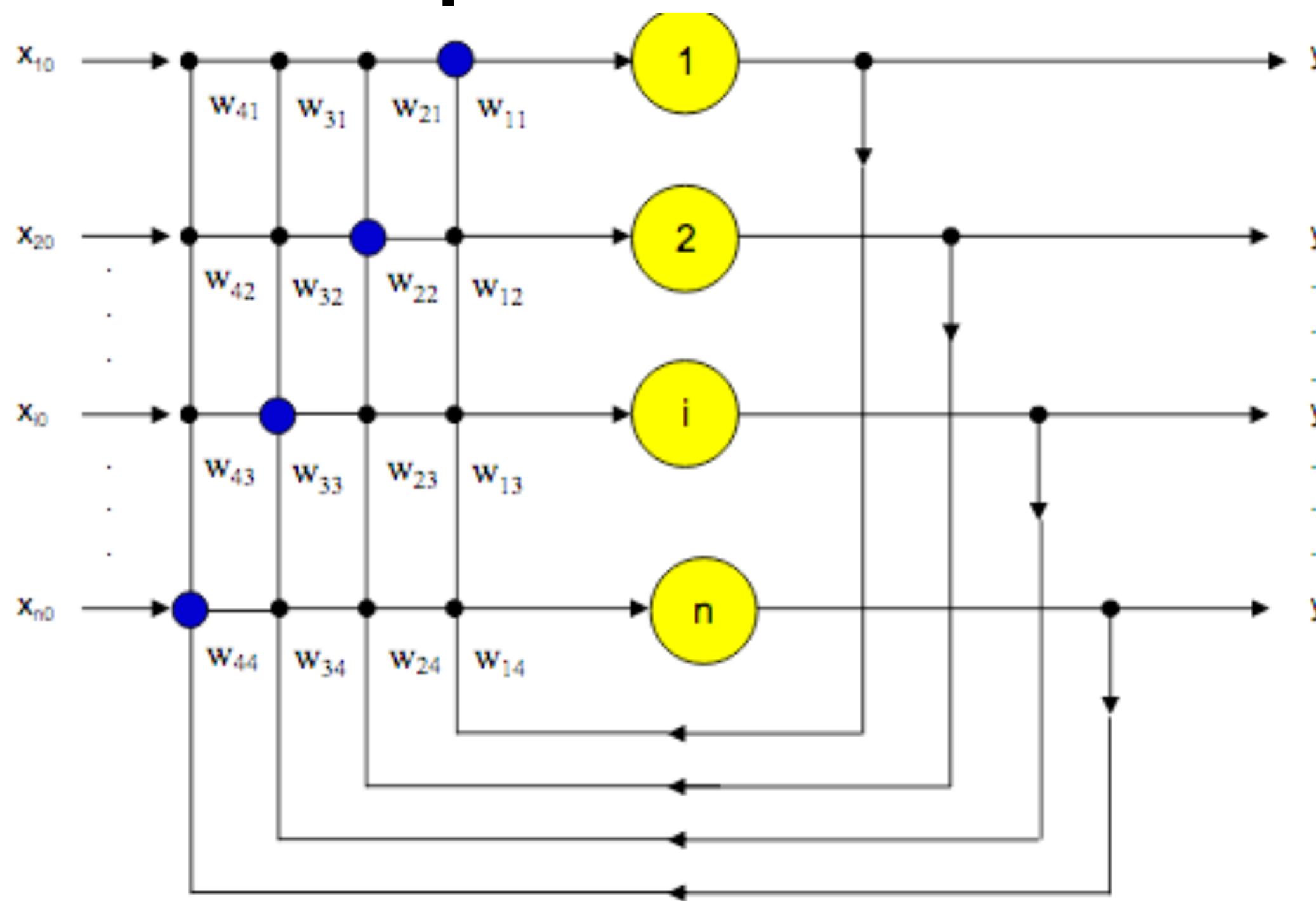
## Key Inspiration from Neuroscience

Build up feature selectivity and tolerance over multiple layers in a hierarchy ⇒  
ML architectures: Neocognitron, HMAX, SIFT, and modern **Deep Convnets**

# Hopfield Nets

- A feedback neural network has feedback loops from its outputs to its inputs. The presence of such loops has a profound impact on the learning capability of the network.
- After applying a new input, the network output is calculated and fed back to adjust the input. This process is repeated until the outcome becomes constant.
- John Hopfield (1982)
  - Associative Memory via artificial neural networks
  - Optimisation

# Hopfield Nets



$$y_j(t) = \text{sgn}(x_j(t)); \text{ sgn} = \begin{cases} +1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

It is a dynamic system:  
 $x(0) \rightarrow y(0) \rightarrow x(1) \rightarrow y(1) \dots \rightarrow y^*$

$$x_j(t) = \sum_{i=1}^n w_{ji} y_i(t-1)$$

# Hopfield Nets

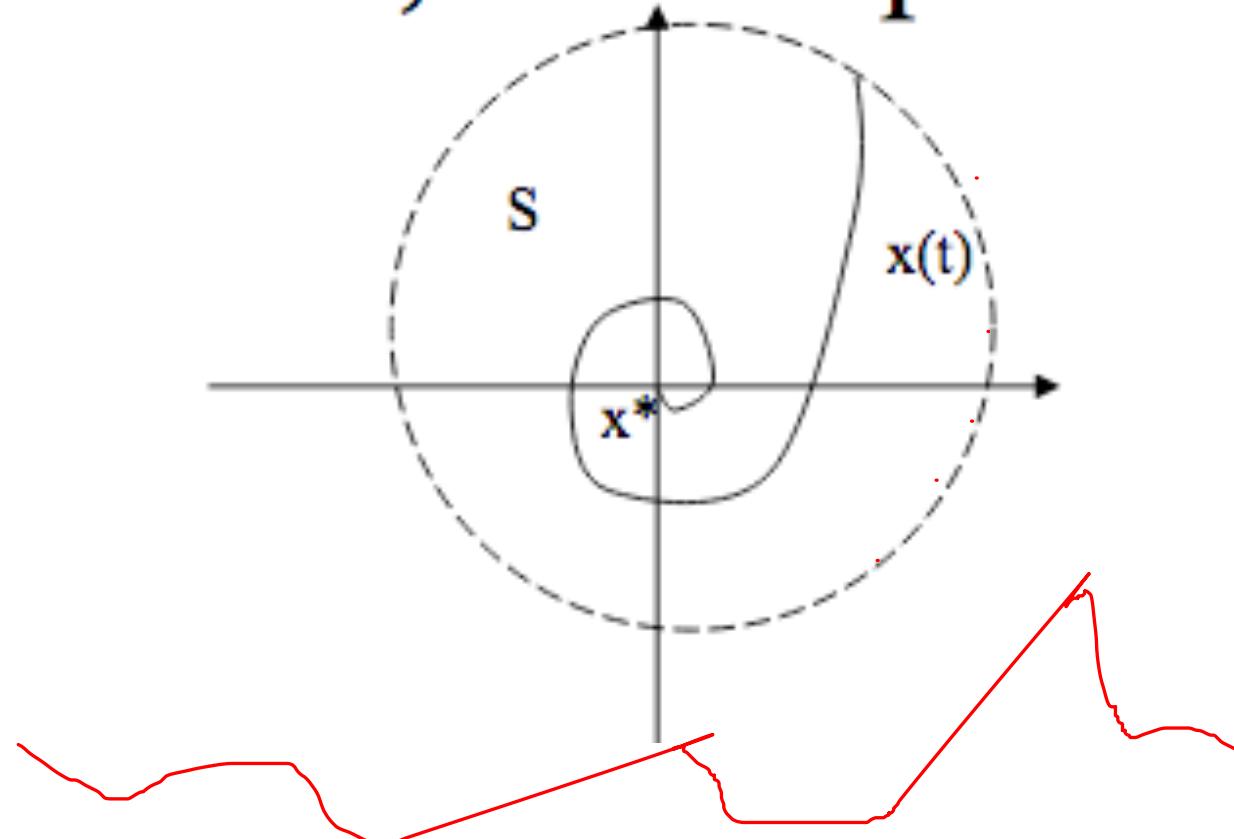
- Attractor

If a state  $x(t)$  in a region  $S$  and  $t \rightarrow \infty$ ,  $x(t) \rightarrow x^*$

$S$  is the attractive region.

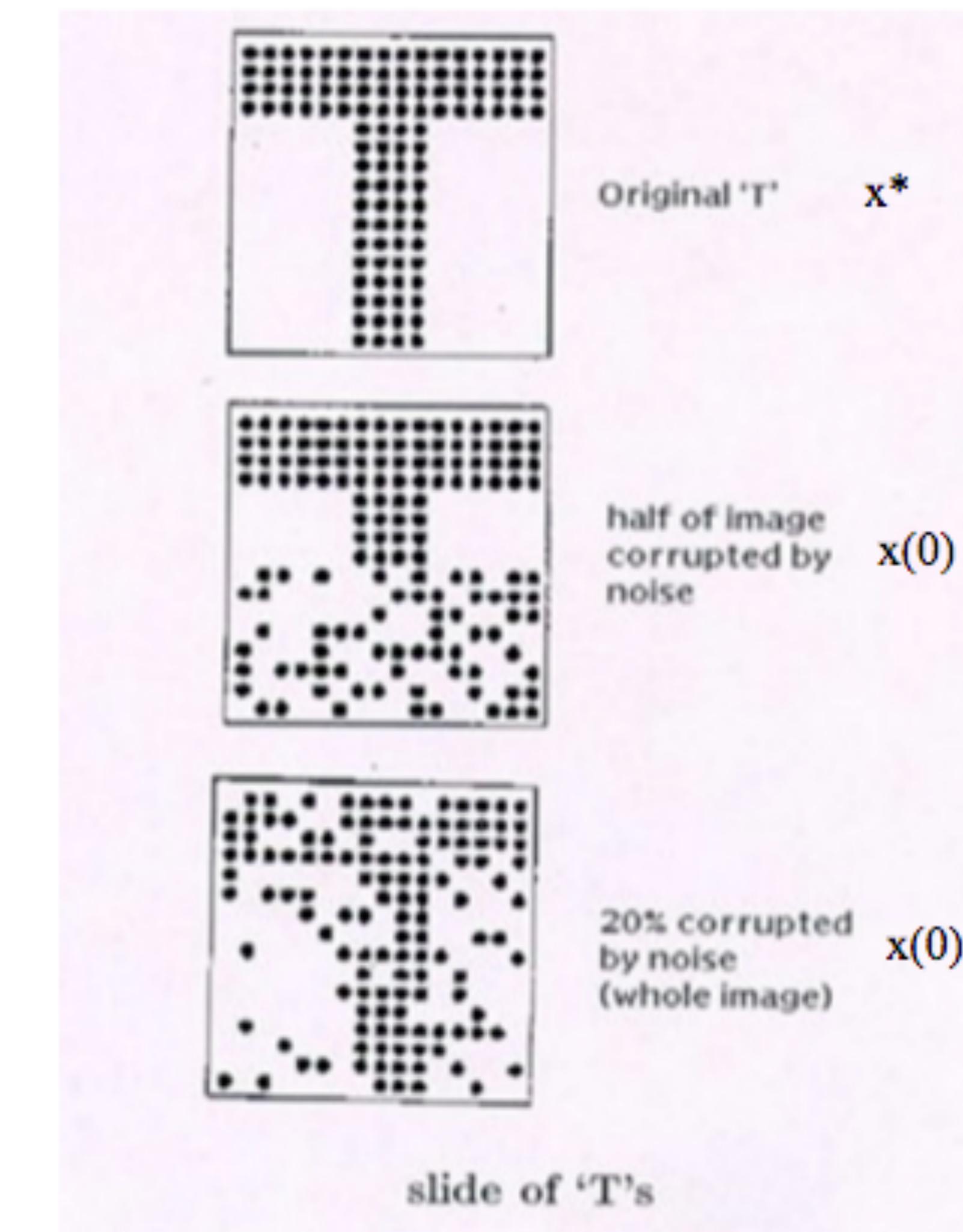
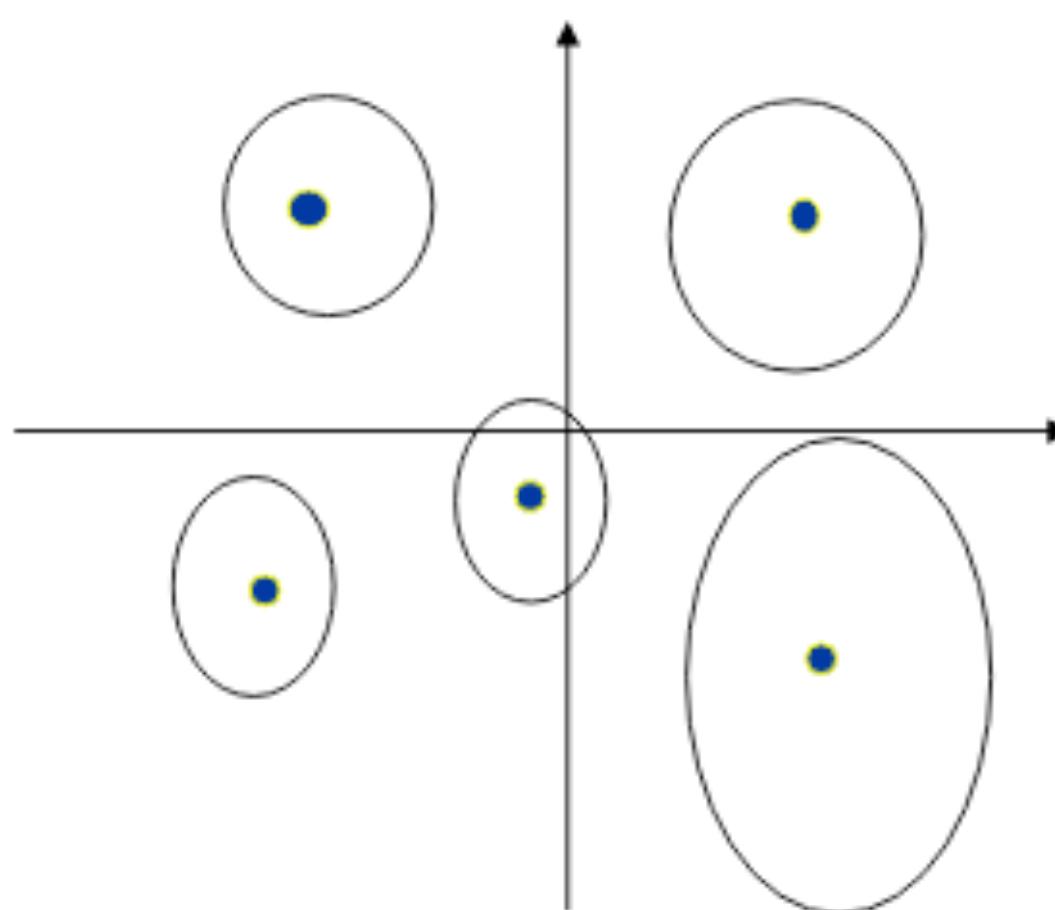
if  $x^* = \text{desired state}$ ,  $x^*$  is an attractor

if  $x^* \neq \text{desired state}$ ,  $x^*$  is a spurious attractor.



# Hopfield Nets

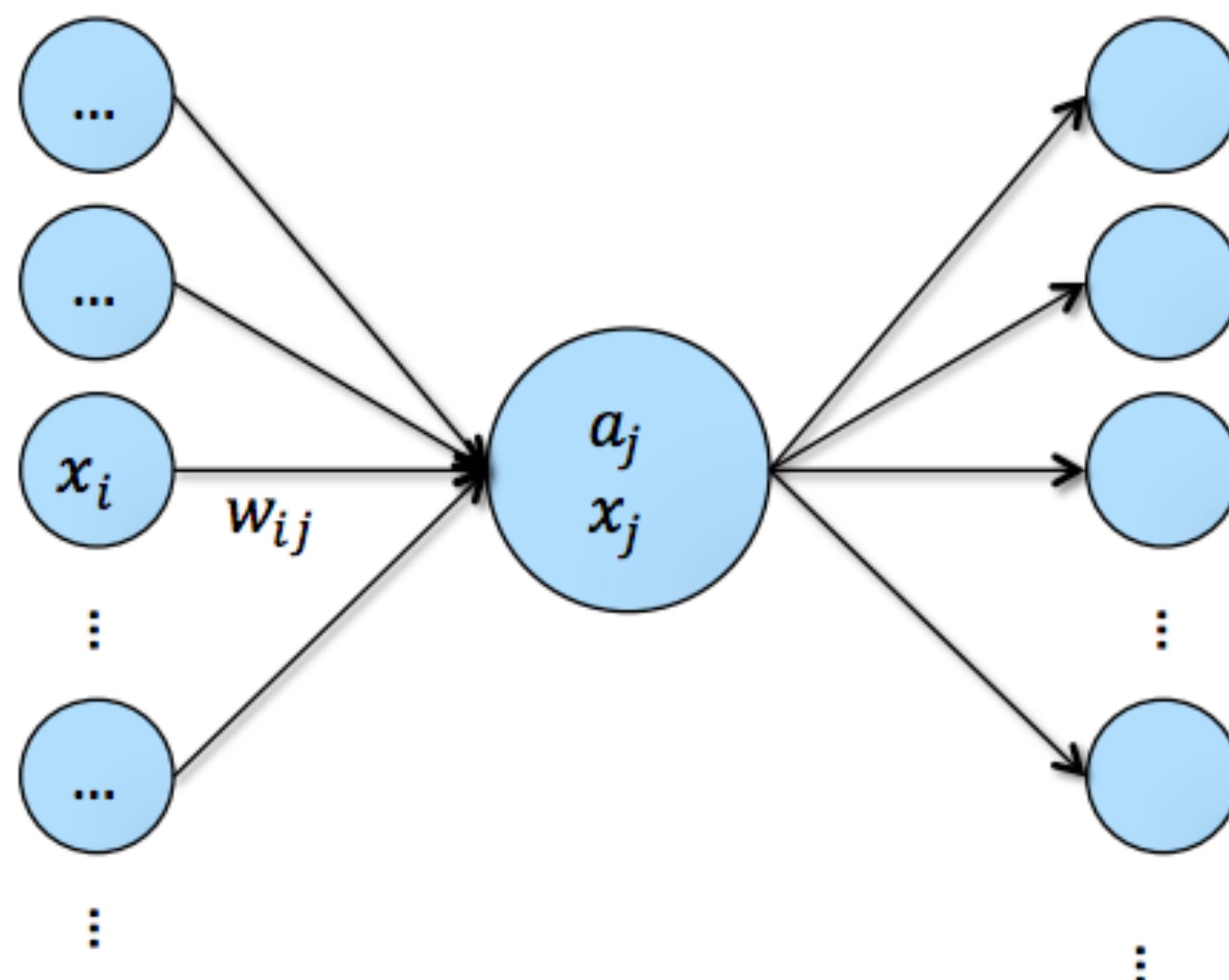
- Nature of associative memory
  - part of information given
  - the rest of the pattern is recalled
- Hopfiled networks can be used as associative memory.
  - Design weight W so that  $X^*$ =the memorised patterns.
  - Can store more than one. Capacity increases



# How to Learn NNs? History of the Backpropagation Algorithm (1960-86)

- Introduced by Henrey J. Kelley (1960) and Arthur Bryson (1961) in control theory, using Dynamic Programming
- Simpler derivation using **Chain Rule** by Stephen Dreyfus (1962)
- General method for Automatic Differentiation by Seppo Linnainamaa (1970)
- Using backdrop for parameters of controllers minimizing error by Stuart Dreyfus (1973)
- Backprop brought into NN world by Paul Werbos (1974)
- Used it to learn representations in hidden layers of NNs by Rumelhart, Hinton & Williams (1986)

# The Backpropagation Algorithm



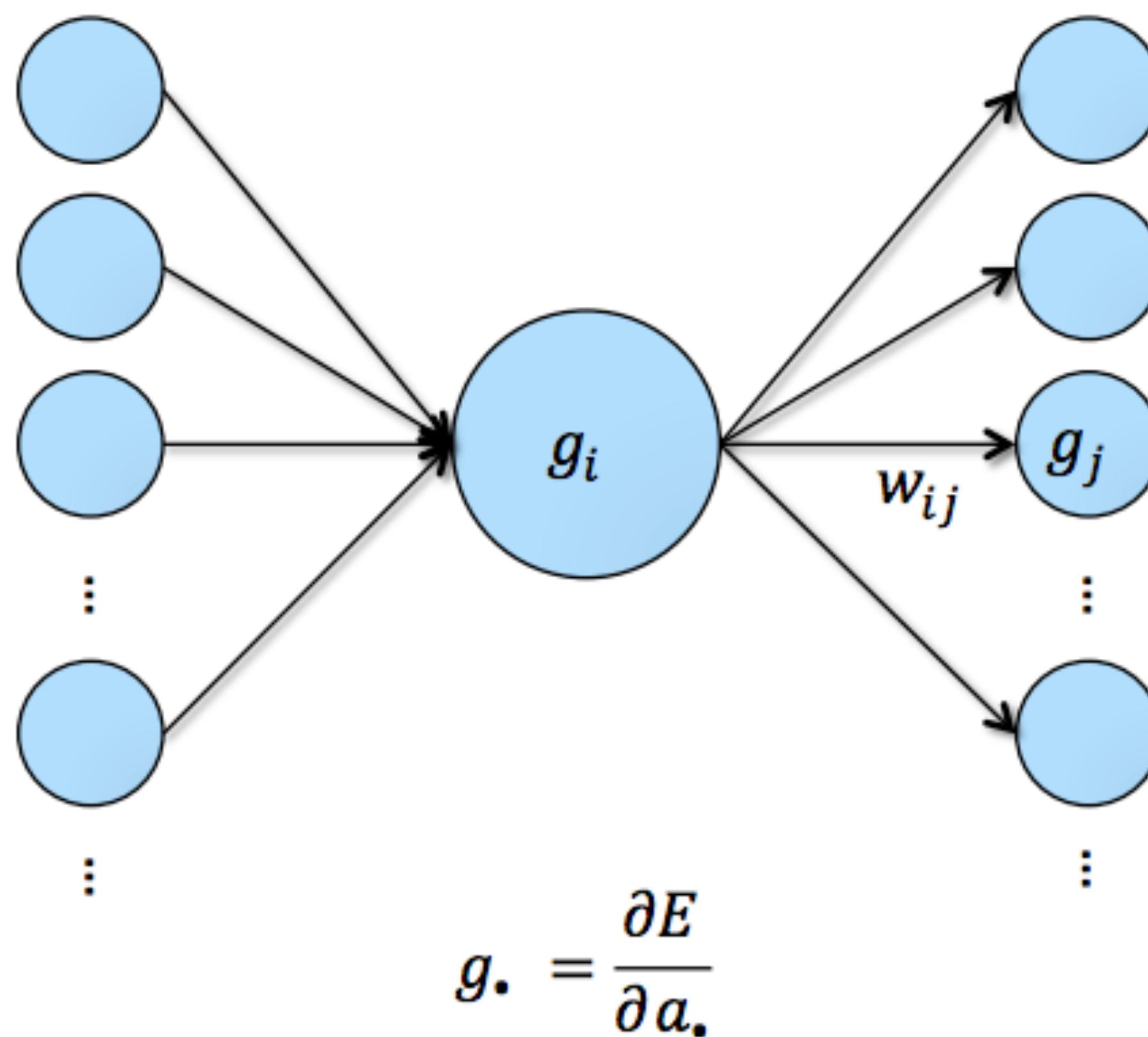
**Net Input**

$$a_j = \sum_{i \in Pre(j)} w_{ij} x_i$$

**Activation + Nonlinearity**

$$x_j = f(a_j)$$

# The Backpropagation Algorithm



Chain rule

$$g_i = f'(a_i) \sum_{j \in Post(i)} w_{ij} g_j$$

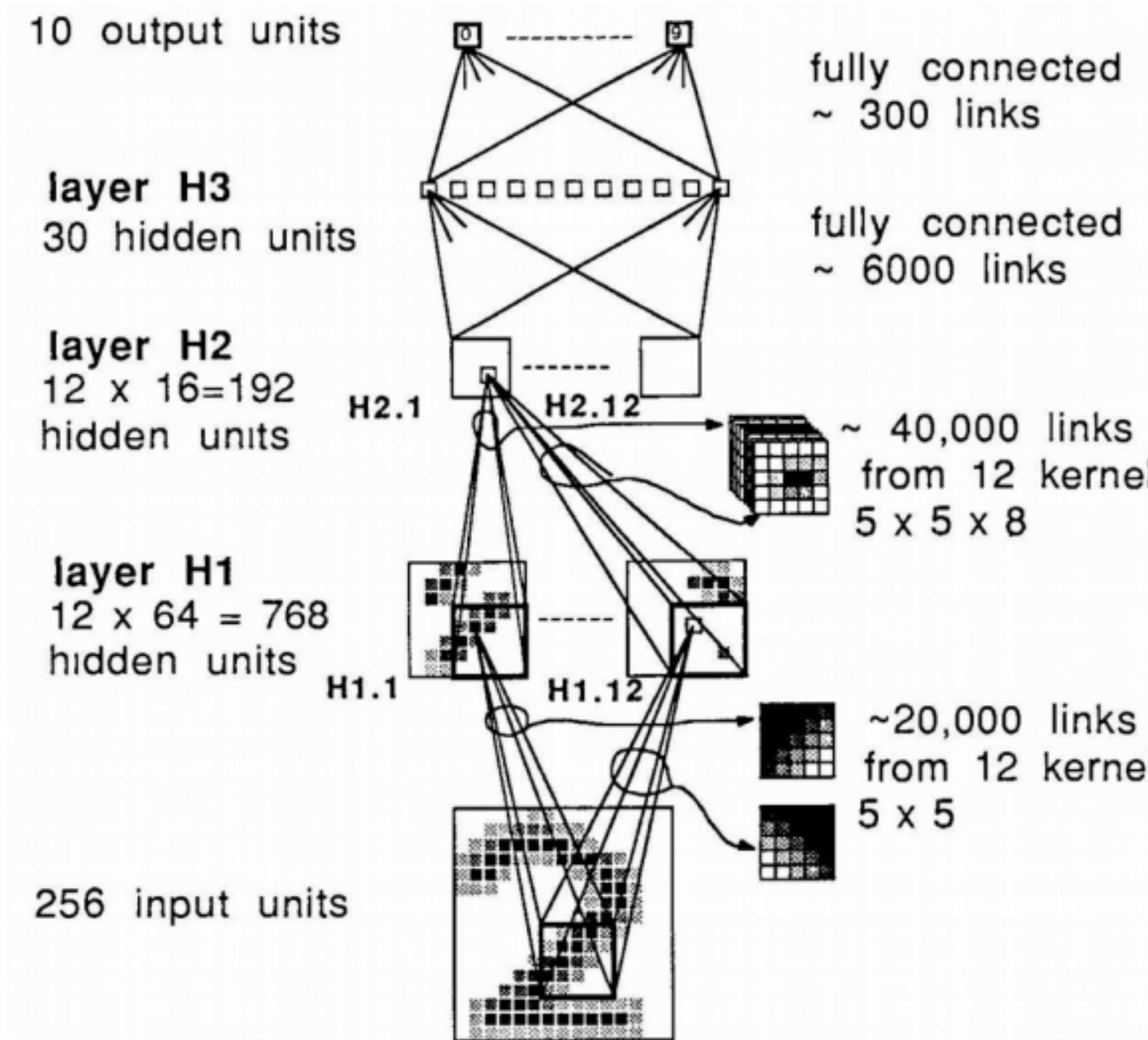
$$\frac{\partial E}{\partial w_{ij}} = x_i g_j$$

# Digit Recognition for the US Post Office (Lecun 1989)

Trained with Backprop.

USPS Zipcode digits: 7300 training, 2000 test.

Convolution with stride. No separate pooling.



80322-4129 80206

40004 14310

37878 05153

55502 75216

35460 44209

1011915485726803226414186  
6359720299299722510046701  
3084111591010615406103631  
1064111030475262009979966  
8912054708557131427955460  
2018730187112993089970984  
0109707597331972015519055  
1075318255182814358090943  
1787541655460354603546055  
18255108503047520439401

# Long Short-Term Memory Recurrent Neural Networks (1992)

- Input Gate
- Forget Gate
- Cell
- Output Gate
- Hidden state output

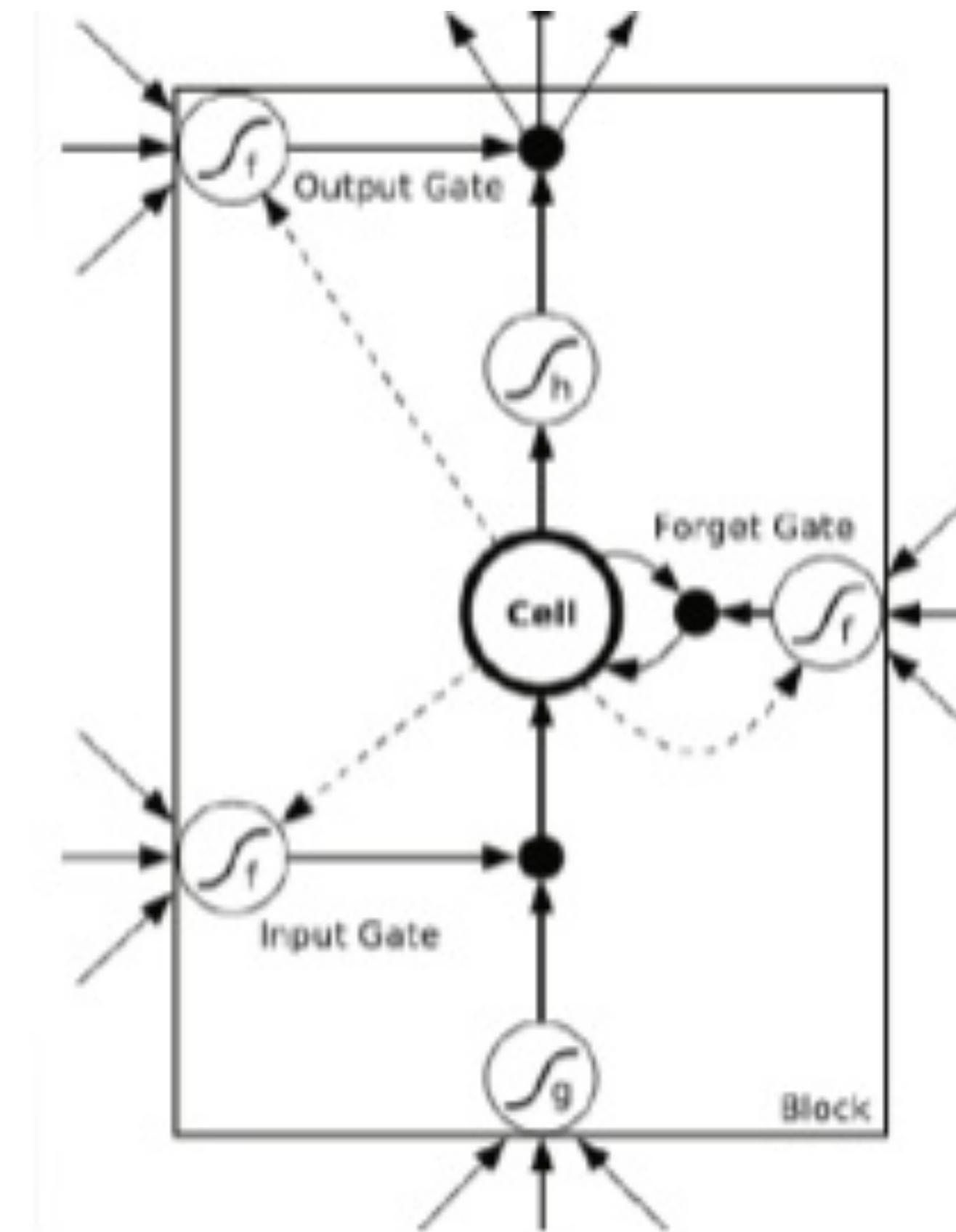
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \tanh(c_t)$$



# History

1992	Weng introduces the <i>Cresceptron</i> and <b>Max-Pooling</b> . Critical for current DNNs. <b>Wavelets</b> and <b>Multi-resolution</b> analysis of images are introduced (but no learning).
1999-2004	Riesenhuber Poggio introduce <b>HMAX</b> : a hierarchical model of visual cortex. Lowe introduces <b>SIFT</b> , <b>scale-invariant feature transform</b> , that is partially inspired by selectivity/invariance properties described in HMAX. SIFT is deep (multi-scale descriptor) but there is no learning.
2001	Breimann introduces <b>Random Decision Forests (RFs)</b> : a combination of <b>Bagging</b> and randomly dropping features in order to prevent <u>overfitting</u> that was so common in decision trees. RFs are employed in many modern day image segmentation tasks in Medical Imaging and for pose estimation in 3D Gaming (Kinect). They also do “unreasonably” well in many ML competitions.
2006-7	Hinton and Salaktdinov use unsupervised pre-training + supervised fine-tuning to train Deep <b>RBMs</b> , setting new records on MNIST.

# History

2006-7	Hinton and Salaktdinov use unsupervised pre-training + supervised fine-tuning to train Deep <b>RBM</b> s, setting new records on MNIST.
2006-10	GPU implementations of BP that are up to 50x faster than CPU are introduced, and can process lots more data in the same time. Emphasized that advances in hardware and Big Data may be more important than specific NN architecture.
2011	Martens and Sutskever introduce <i>Hessian-Free Optimization</i> : a training algorithm for RNNs that can alleviate vanishing gradient problem.
2012 - present	Krizhevsky and Hinton show that CNNs can win ImageNet competition. A resurgence of interest in DL: Lots more Contests and Benchmarks advanced. And the hype is back as well! A new <i>Theory of Deep Learning (developed here at Rice)</i> that shows <b>the probabilistic origin of deep CNNs and RFs</b> .  Generative model $\Rightarrow$ Probabilistic Inference $\Rightarrow$ NN relaxation

# Graphical Processing Units (GPUs) revolutionize Deep Learning

- GPUs first introduced in 2006 for DL
- Order of magnitude increase in speed of training
- Nvidia is the major player; Intel and others lagging behind



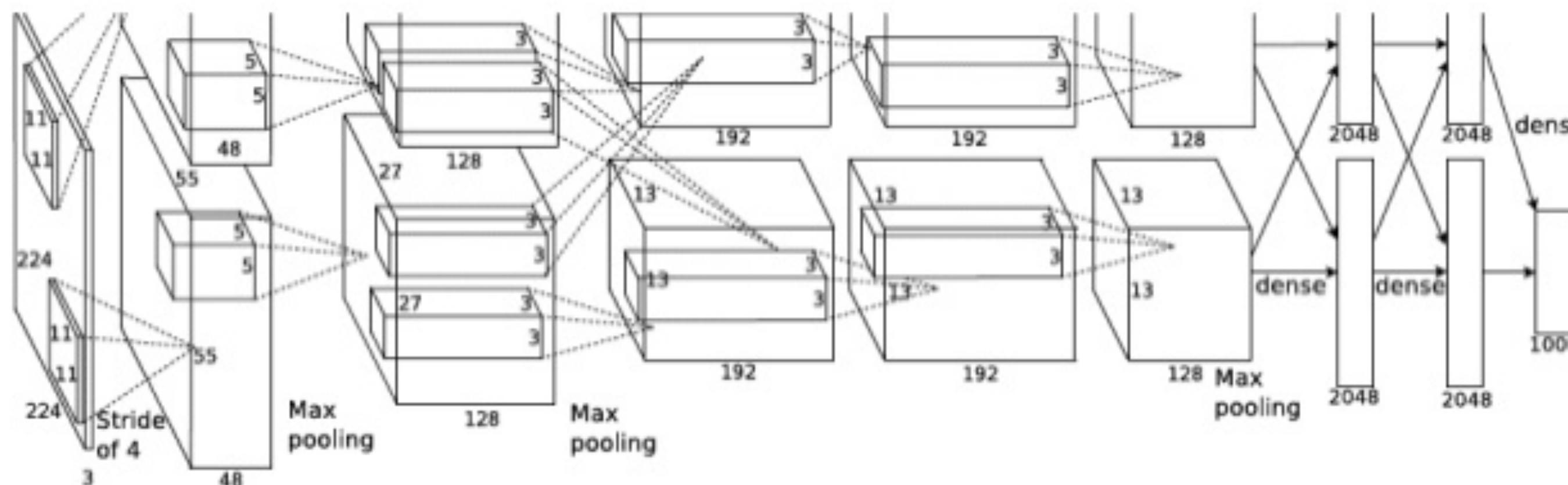
# ImageNet Dataset (2011): The Largest Hand-Labeled Dataset in the World



# Convnets dominate ImageNet Challenge (2012)

## AlexNet

- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data ( $10^6$  vs.  $10^3$  images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week

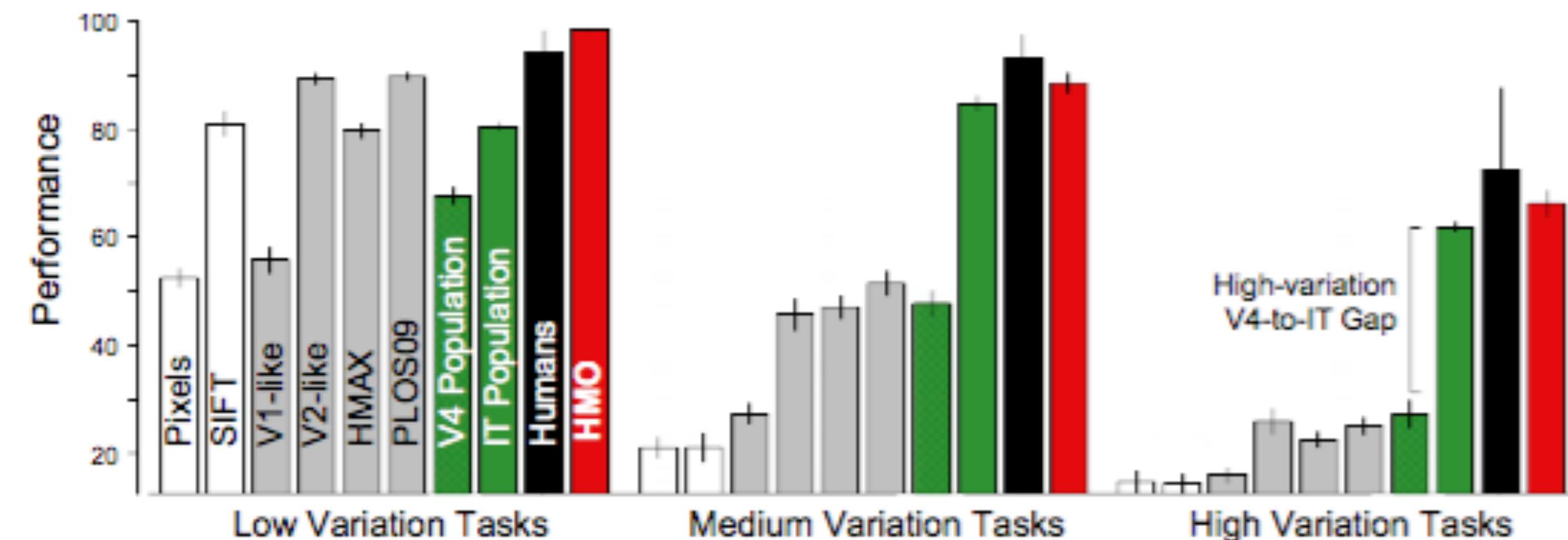


A. Krizhevsky, I. Sutskever, and G. Hinton,  
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

# Deep Learning: Recent Applications to Neuroscience

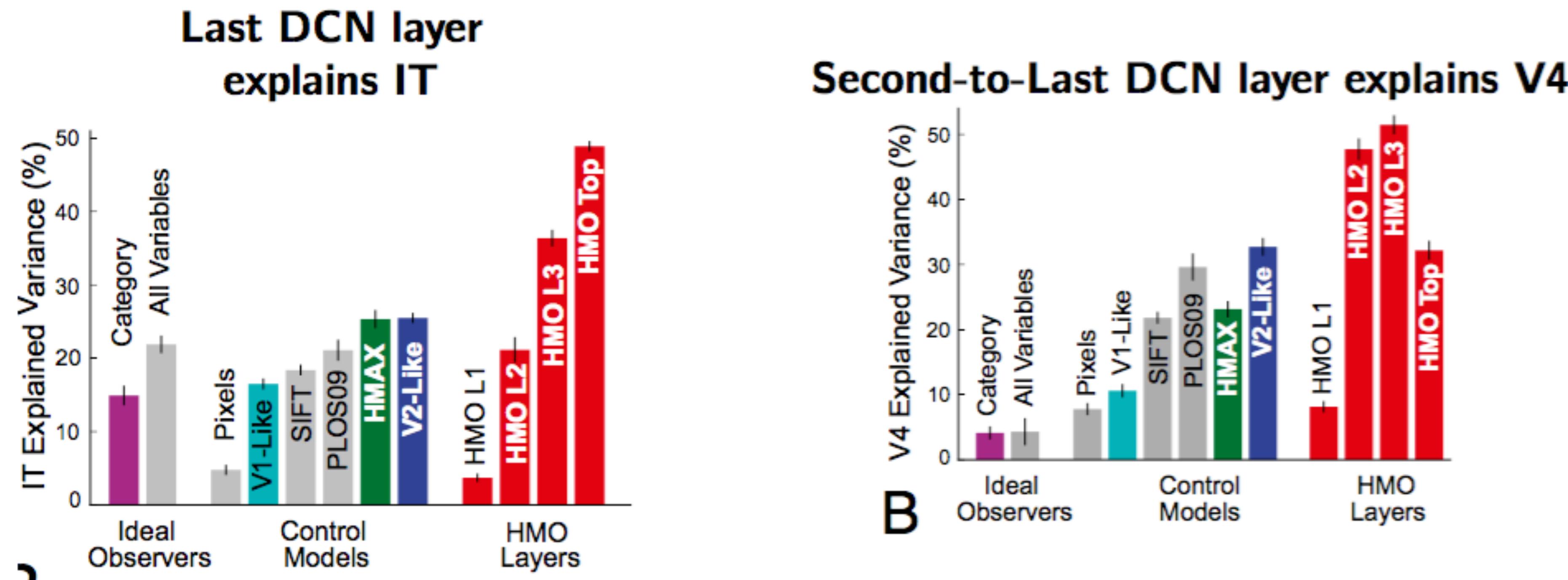
**Recent Finding:** Best model  
of neural activations in  
ventral stream (V1 to IT) is a  
DCN

- ▶ Low nuisance variation: all models perform well
- ▶ High nuisance variation: IT and DCN perform best by far



D. Yamins, H. Hong, C. Cadieu, E. Solomon, D. Seibert, and J. J. DiCarlo.  
*Performance-optimized hierarchical models predict neural responses in higher visual cortex.* PNAS 2014

# Deep Learning: Recent Applications to Neuroscience



D. Yamins, H. Hong, C. Cadieu, E. Solomon, D. Seibert, and J. J. DiCarlo.

*Performance-optimized hierarchical models predict neural responses in higher visual cortex.* PNAS 2014

## The Implication

A better understanding of Deep Convolutional Nets may lead to  
a better understanding of the Visual Cortex

# Mixer!

- Introduce yourself to me so I can get to know you a little bit more personally (1-2min)
- Introduce yourself to at least 3 people you don't already know
- What is your research area(s) of interest? What are the important problems in your field(s)?
- How can Deep Learning impact your research area?