Neural Networks

Lee Spector (lspector@hampshire.edu)

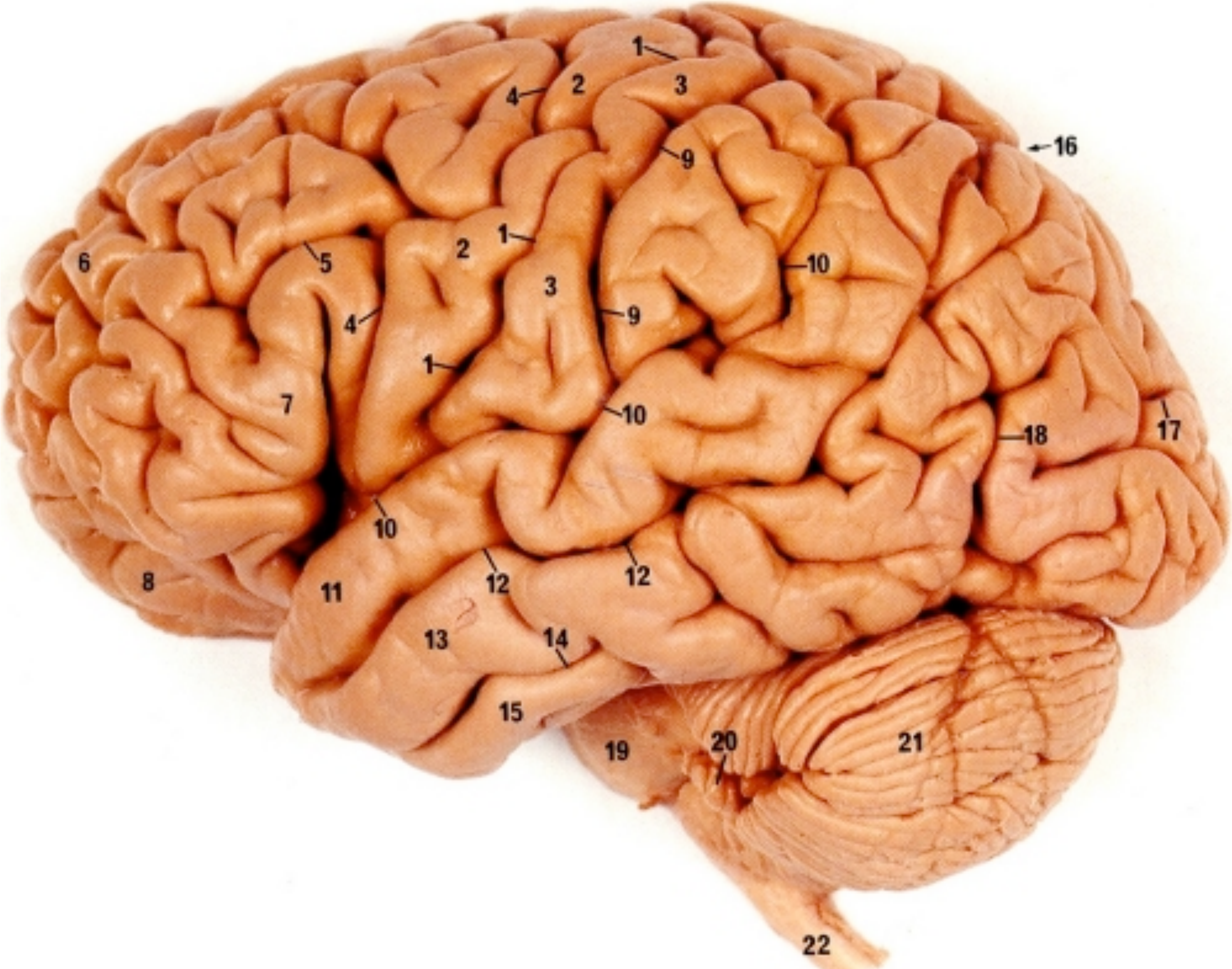
CS 263

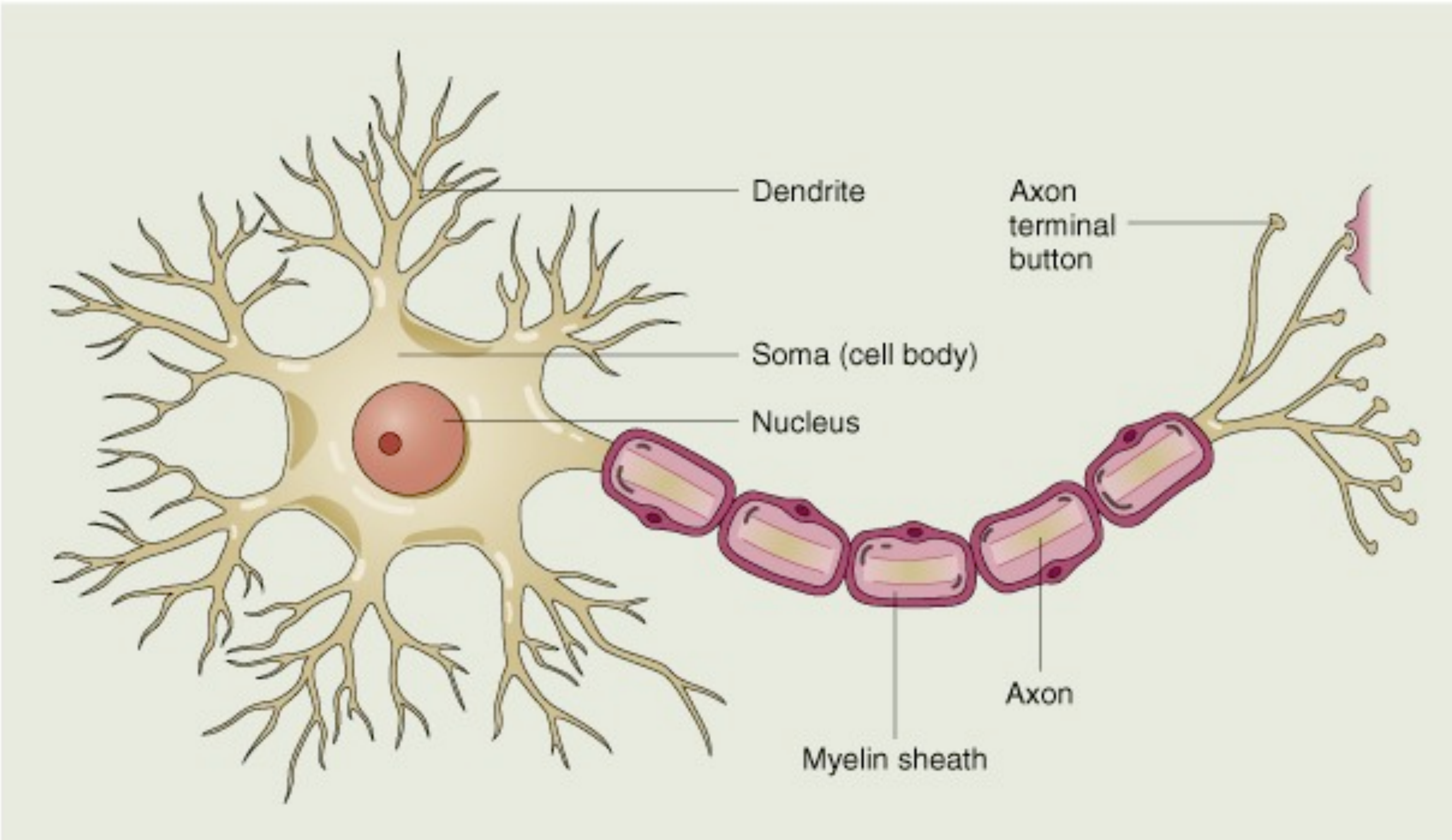
Brain Architecture

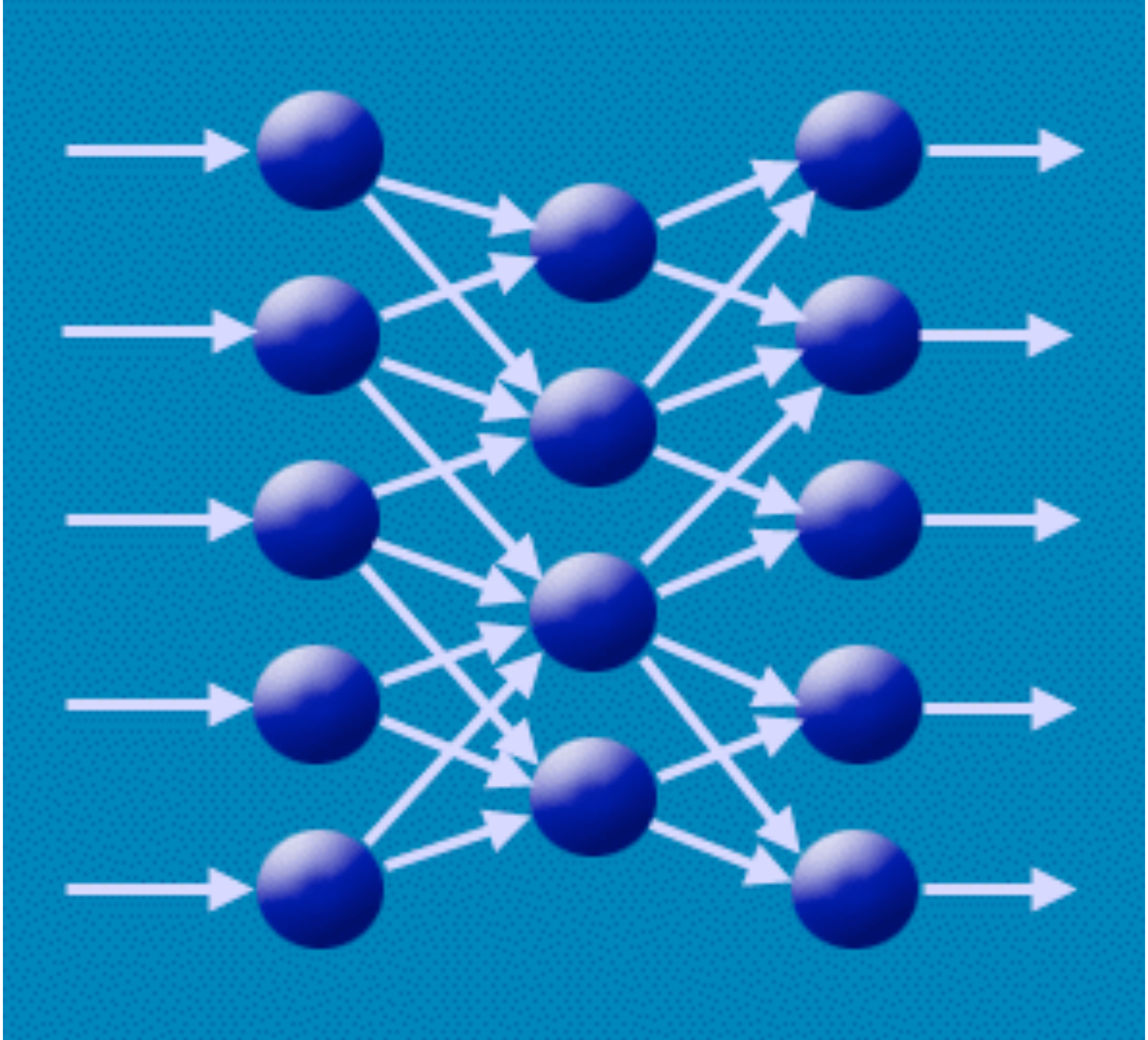
- Many AI models are based on the Von Neumann Architecture.
- The brain is a Massively Parallel computing machine. It is also slow (ms. range), "noisy" and exhibits fault-tolerance.
- The idea of connectionism (a.k.a. "neural network research") is, roughly, to build systems on the brain's model, rather than on Von Neumann's model.

What is going on in your brain right now?

- As you look at this?
- As you listen to me?
- As you think of what your answer is?


1
4
2
3
16
9
6
5
2
1
10
3
4
9
1
7
10
18
17
10
12
12
8
11
13
14
15
19
20
21
22


Dendrite
Axon
terminal
button
Soma (cell body)
Nucleus
Axon
Myelin sheath



Human Neurons

- With your right hand, hold up fingers indicating your threshold.
- When you get that much activation within 1 second FIRE:
  - Say “Bang!”
  - Poke your target’s shoulder to stimulate
  - Pat your target’s shoulder to inhibit

(demonstrate)

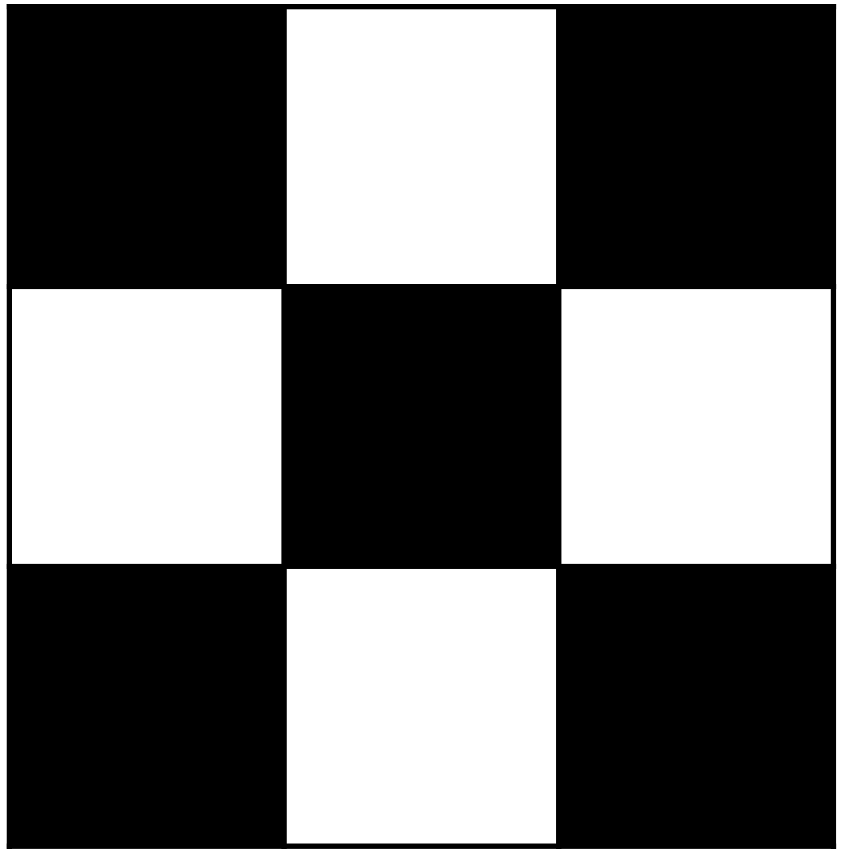
Sensory Neurons

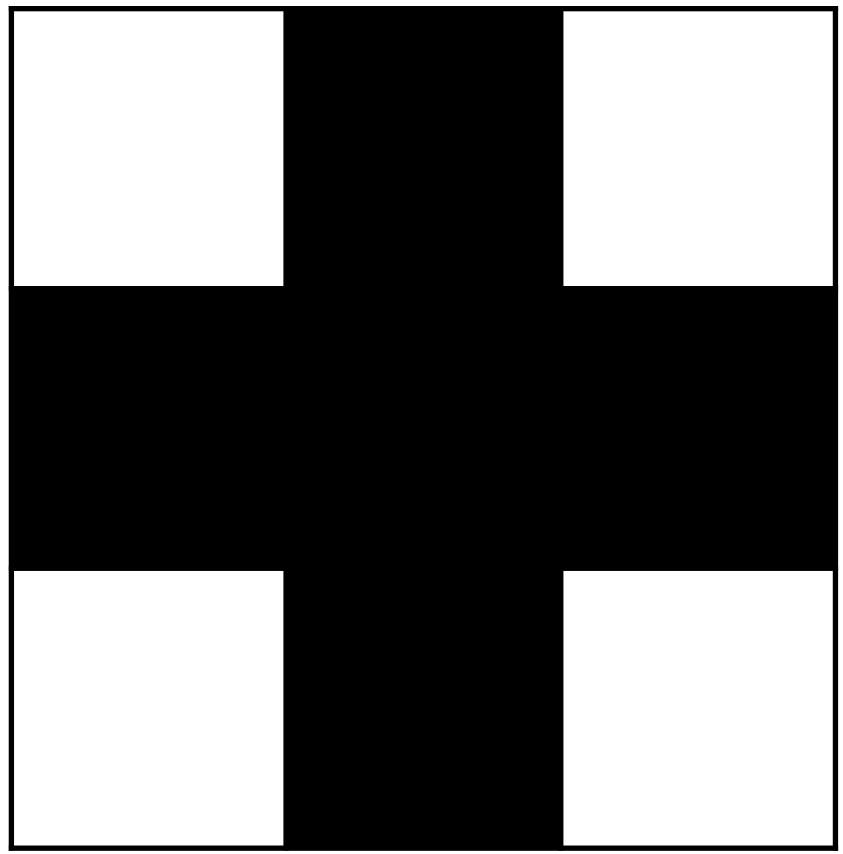
- Fire every 2 seconds if you are stimulated.

Motor Neurons

- Make a loud noise when you fire

1	2	3
4	5	6
7	8	9





What is the neural network for...

- Your memory of your best friend’s face?
- Your memory of waking up this morning?
- The taste of chocolate ice cream?
- My voice?
- Doing multiplication?
- Being happy?

PDP

- Connectionist systems are examples of "Parallel Distributed Processing."
- The "PDP books" are historically important and provide a good general introduction.
- The key idea of PDP is "the notion that intelligence emerges from the interactions of large numbers of simple processing units." (PDP1, ix)
- Connectionism is a microstructural enterprise (PDP1, 12).
- An *non-connectionist* example of PDP: the cellular automaton "game" of Life. Surprising complexity can result from the interactions of the small set of simple rules.

Connectionist Learning

- In many connectionist systems, connections have strengths, such that the stronger the connection, the more activation passes through it.
- Such networks can be trained by adjusting the strengths of the connections to improve the system's performance.

Connectionist Learning

- There are a variety of paradigms for connectionist system learning.
- The simplest learning rule is due to Hebb (1949): "When unit A and unit B are simultaneously excited, increase the strength of the connection between them." (PDP1, 36).
- More elaborate learning rules have been used; some are more powerful.

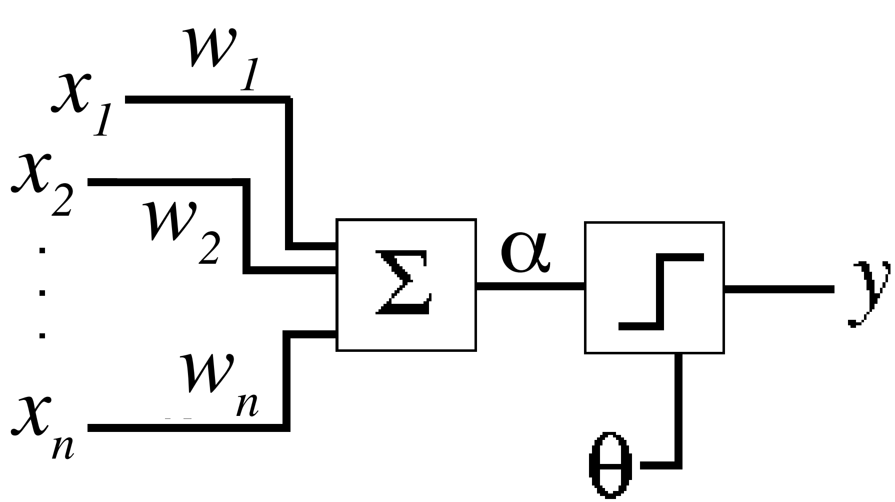
Perceptrons

[Rosenblatt, 1962]

- Early neurally-based PDP model.
- n inputs ($x_1 ... x_n$), one output (y)
- Each input has a numeric weight w_i
- The output has a threshold θ.

Perceptrons

$$\alpha = \sum_{i=1}^{n} w_i x_i \qquad\qquad y = \begin{cases} 1 & \text{if} \quad \alpha \geq \theta \\ 0 & \text{otherwise} \end{cases}$$



Perceptron Logic

- Perceptron for n-input AND gate:
  all weights = 1
  $\theta = n - 1/2$

- Perceptron for n-input OR gate:
  all weights = 1
  $\theta = 1/2$

- Perceptron for 1-input NOT gate:
  weight = -1
  $\theta = -1/2$

Perceptron Retina

- Perceptron for template-style letter "A" recognizer in a 5x5 array (inputs are either -1 or 1).
- Weights, in 2D arrangement:

-1	-1	1	-1	-1
-1	1	-1	1	-1
1	1	1	1	1
1	-1	-1	-1	1
1	-1	-1	-1	1

- θ = 25 - epsilon

Perceptron Learning

- X+ = positive training examples
- X- = negative training examples
- Set all weights to arbitrary initial values (perhaps 0)
- Update the vector of weights W for each example:

  $W_{k+1} = W_k$, if X_k is correctly classified by W_k
  $W_k - c_k X_k$, if X_k is in X- but is misclassified by W_k
  $W_k + c_k X_k$ if X_k is in X+ but is misclassified by W_k

Fundamental Training Theorem

Given a set of training samples X and any training sequence for it, if c_k is taken as a positive constant, and at least one solution vector of weights exists for X, then the standard training procedure will find a solution after a finite number of steps.

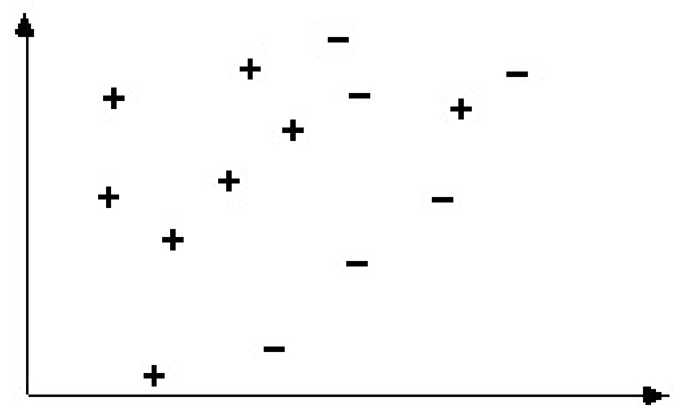
Connectedness

- The connectedness problem: to tell whether all of the “on” pixels in an image are connected.
- Minsky and Papert showed that with reasonable limits on perceptron structure, a single layer of perceptrons can't solve the connectedness problem.

Limits of Perceptrons

More generally, input vectors can be considered as points in an n-dimensional space. A single-layer perceptron can solve a problem only if there exists an n-1-dimensional straight line (plane, etc.) that separates the input points corresponding to positive cases from those corresponding to negative cases (that is, when the two sets of points are “linearly separable”).

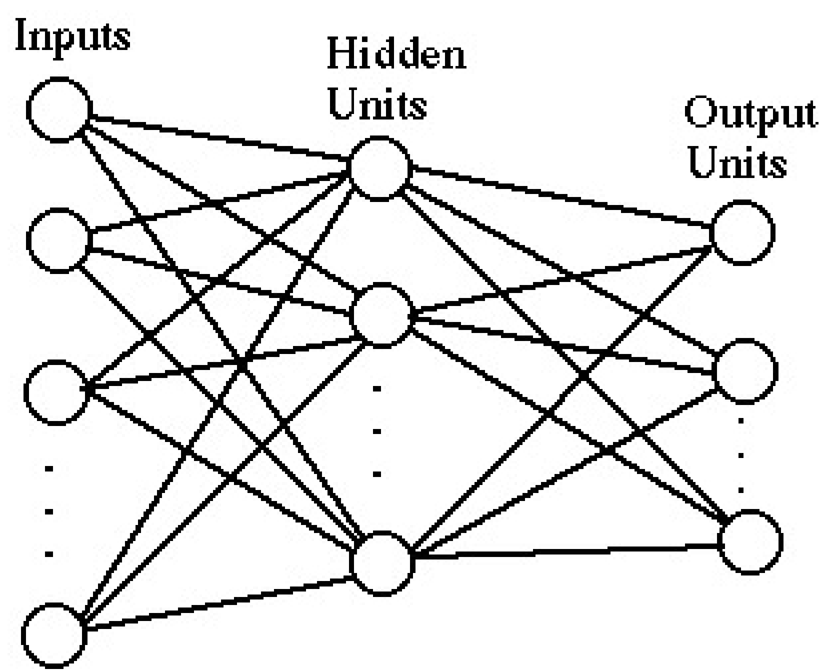
Linear Separability



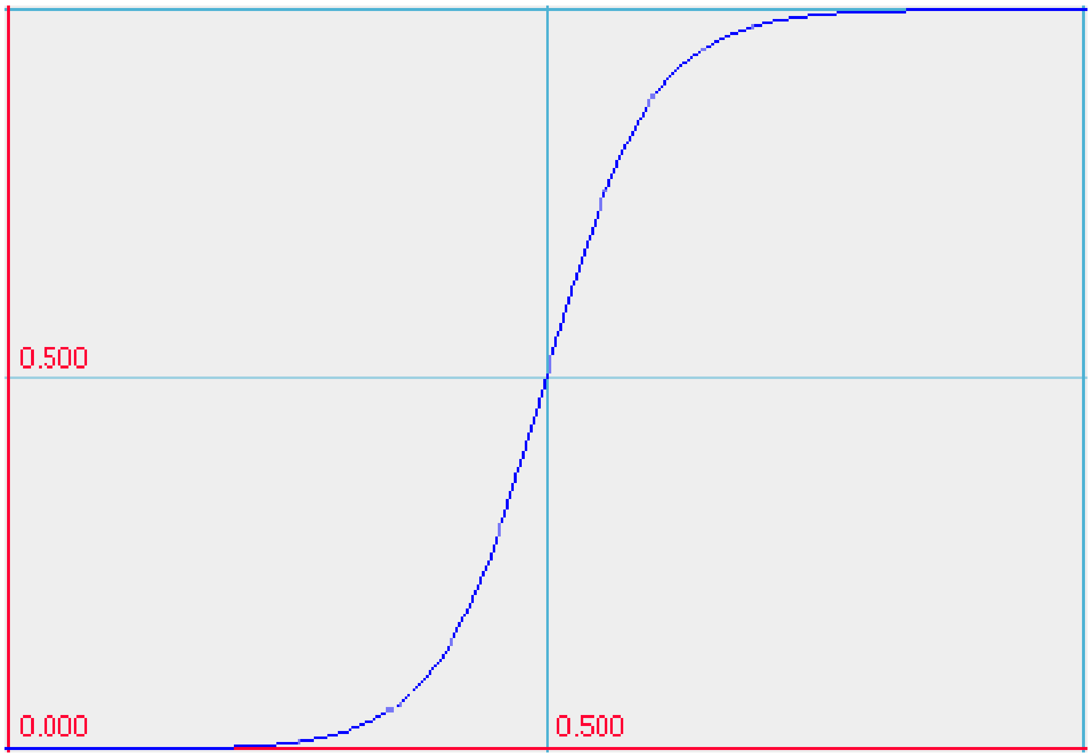
Note: Multiple layers of perceptrons CAN separate these sets.

Feedforward Networks

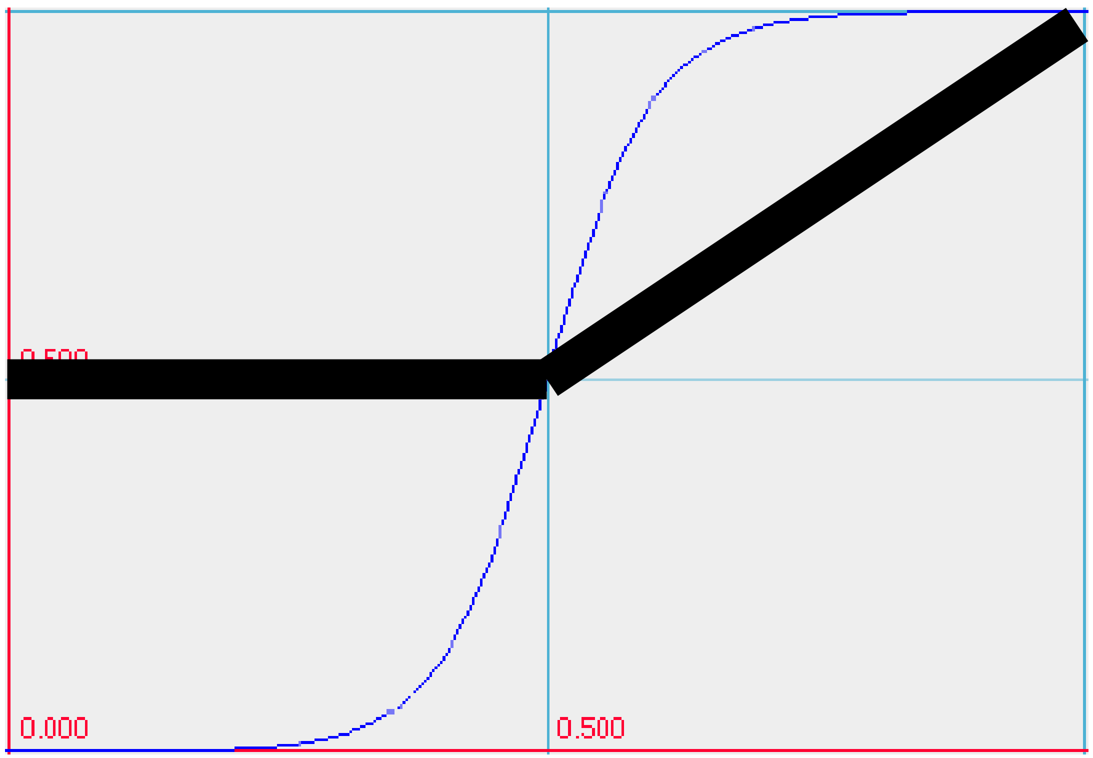
Multiple layers, including "hidden" layers.



Sigmoid Activation



ReLU Activation



Rectified Linear Unit

Training with Backpropagation of the Error

- Process each training example by feeding it (forward) through the net.
- Compute the error vector: the output vector minus the correct ("target") output vector.
- Use the error vector to adjust the weights on the connections to the output units, and then to adjust the weights on the connections to the previous level, etc.
- The "delta rule" is used to adjust the weights. See Nilsson for details.

Pros & Cons

- Features of connectionist systems:
  - fault tolerance
  - automatic similarity computations
  - learning
  - generalization
- Limitations:
  - complexity
  - opacity
  - timing
  - sequencing
  - symbol manipulation

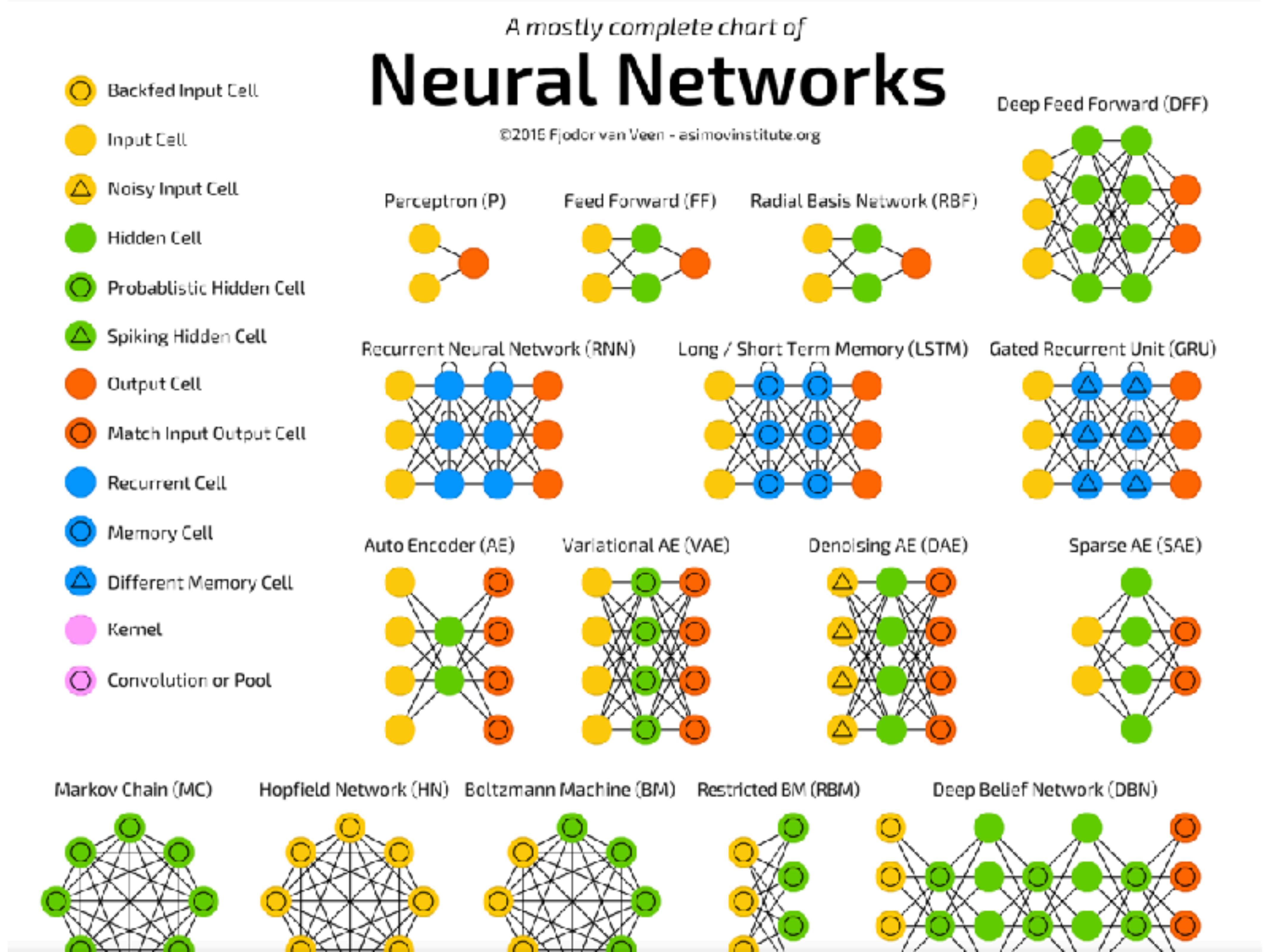
Neural Nets and AI

Andy Clark (in Margaret Boden's 1990 *The Philosophy of Artificial Intelligence*) suggests an analogy:

symbolic AI : connectionism ::

  Newtonian physics : quantum mechanics

Clark questions the explanatory role of connectionist theories on cognitive science, but says that they can be explanatory through post-processing techniques such as cluster analysis.


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)