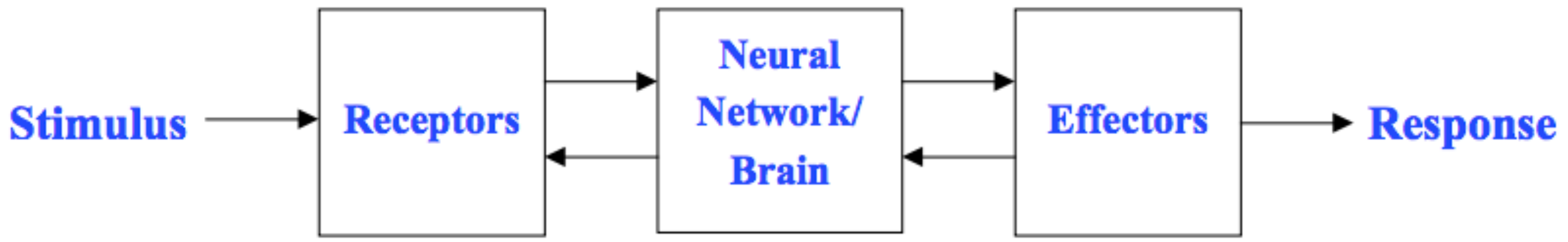
Perceptron/Multi-Layer Perceptron

Outline

- Perceptron algorithm.
- Multi-layer perceptron.
- Universial approximation theorem.
- Other networks/Retricted Boltzmann machine

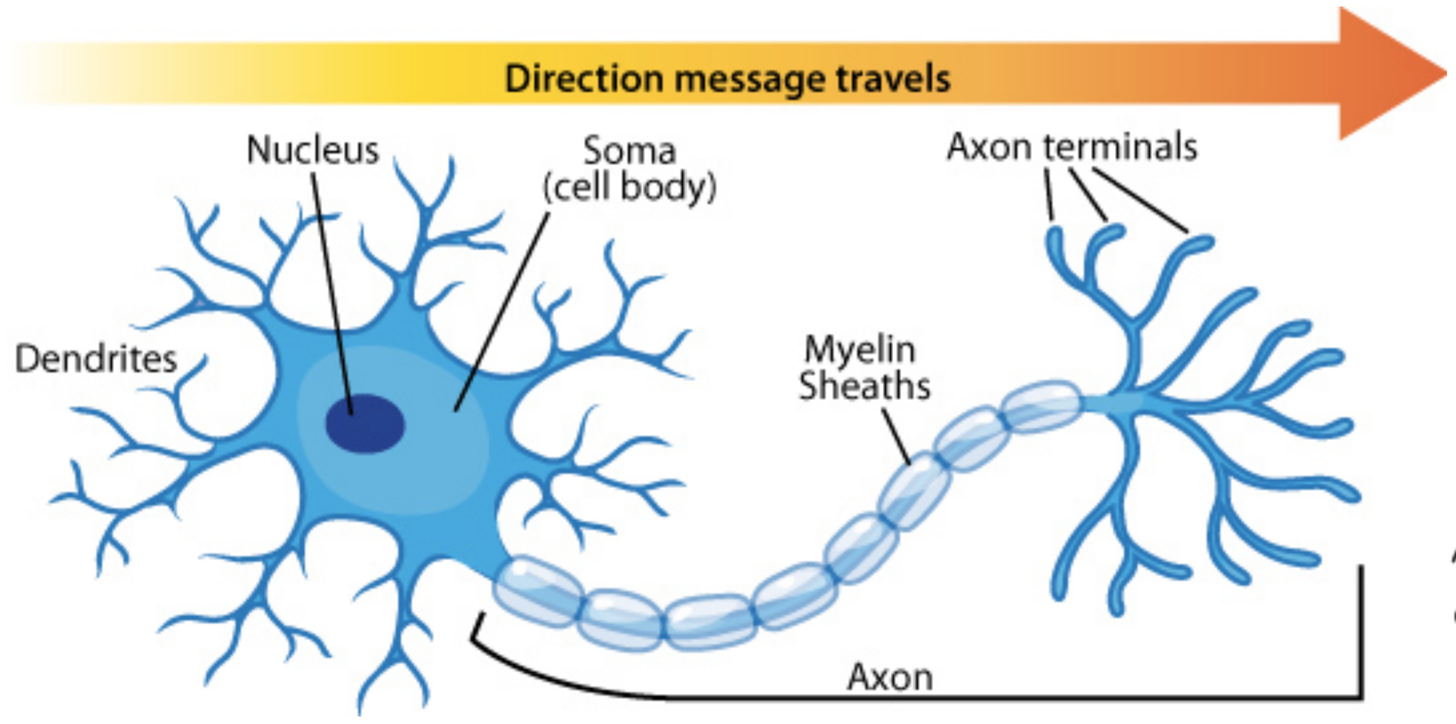
Nervous System



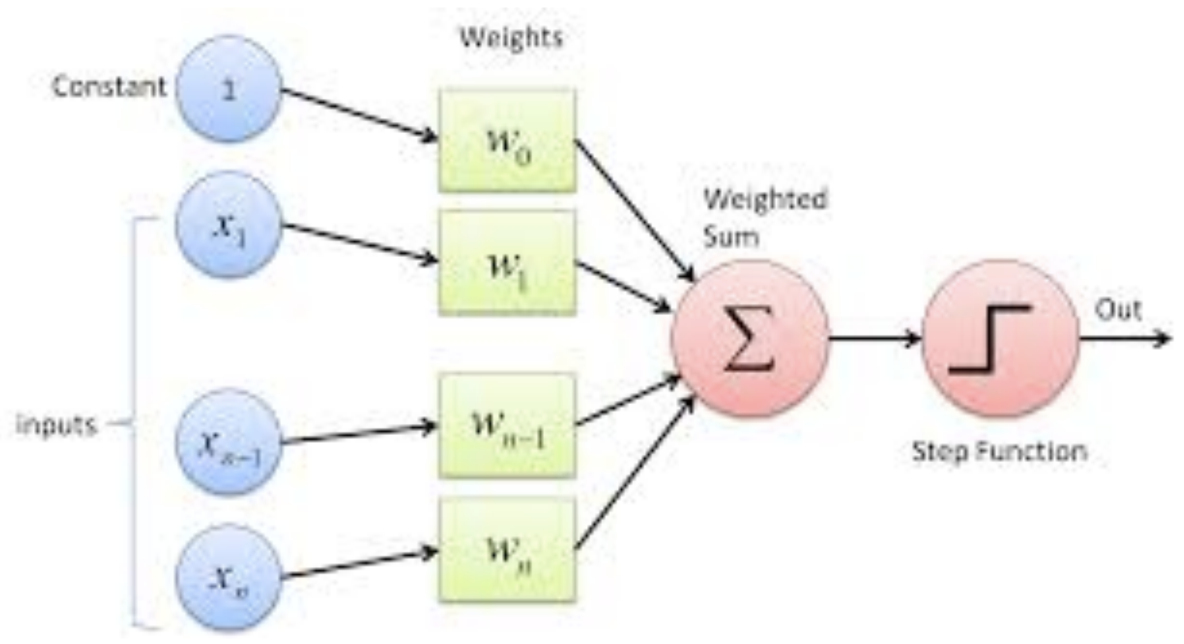
Levels of Brain Organization

1. Molecules and Ions
2. Synapses
3. Neuronal microcircuits
4. Dendritic trees
5. **Neurons**
6. **Local circuits**
7. Inter-regional circuits
8. Central nervous system

Perceptron



$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\right)$$



$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0. \end{cases}$$

Implementation of Logical NOT, AND, and OR

NOT

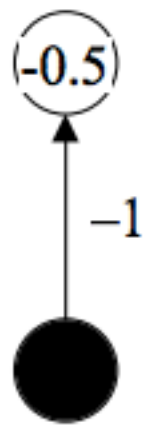
in	*out*
0	1
1	0

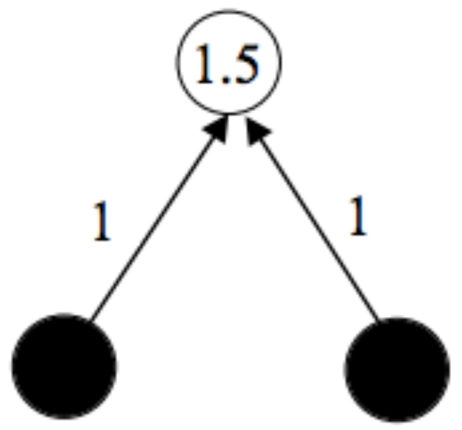
AND

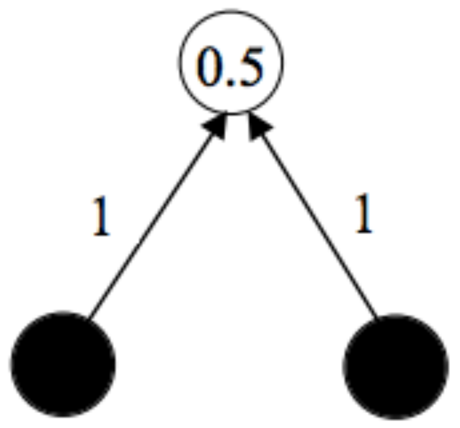
in_1	in_2	*out*
0	0	0
0	1	0
1	0	0
1	1	1

OR

in_1	in_2	*out*
0	0	0
0	1	1
1	0	1
1	1	1







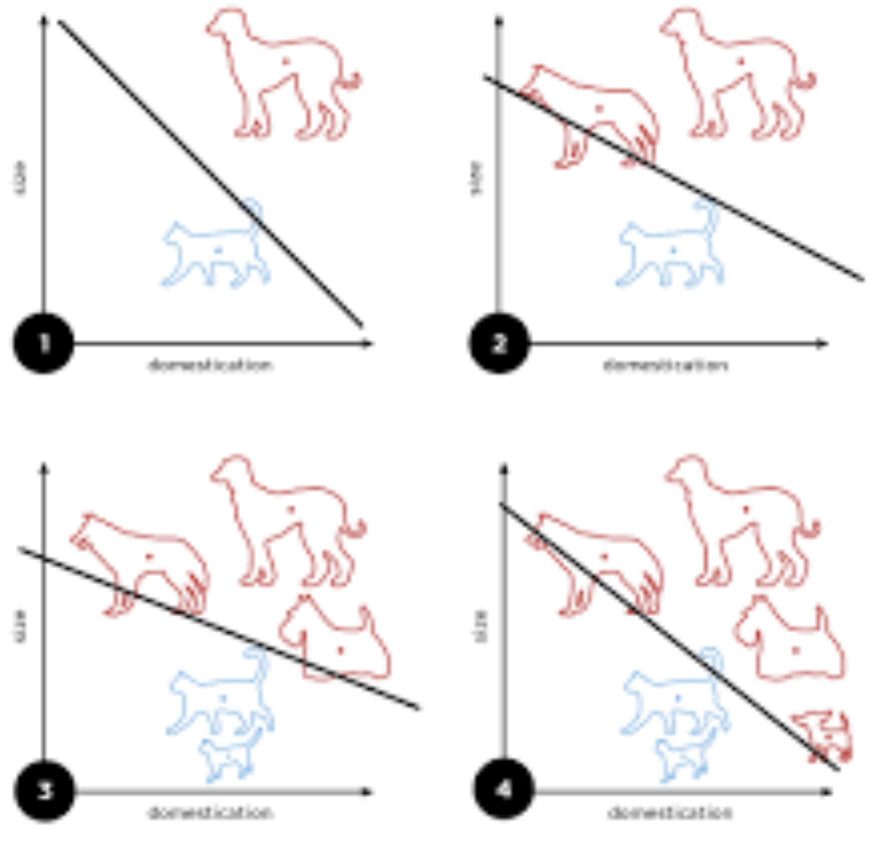
Perceptron (bio)

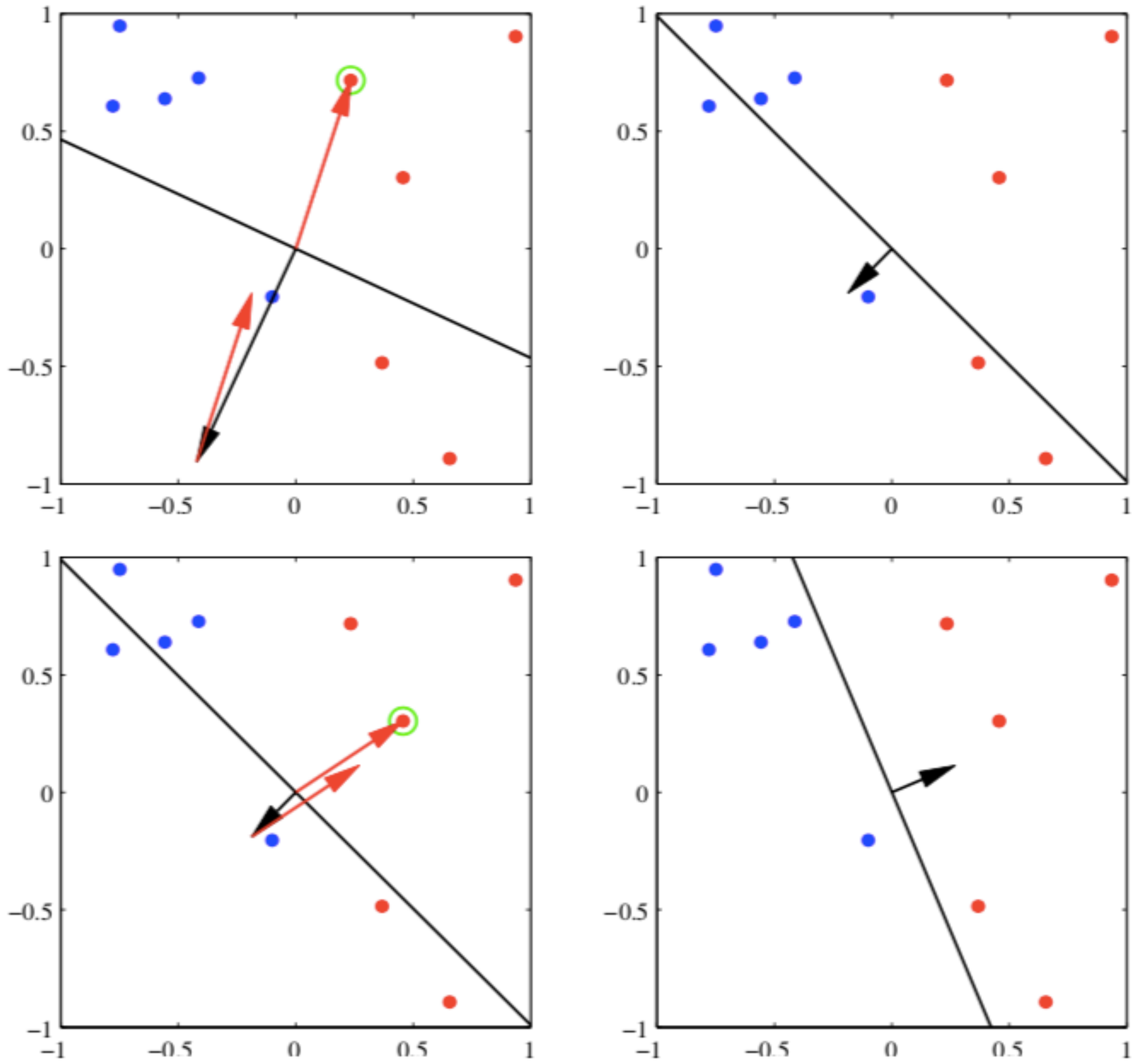
- Different types of neurons.
- Dendrites can do more than linear weight and bias.
- Synapses are complex dynamical system.
- Rate coding vs spike coding

Perceptron training: 1st try

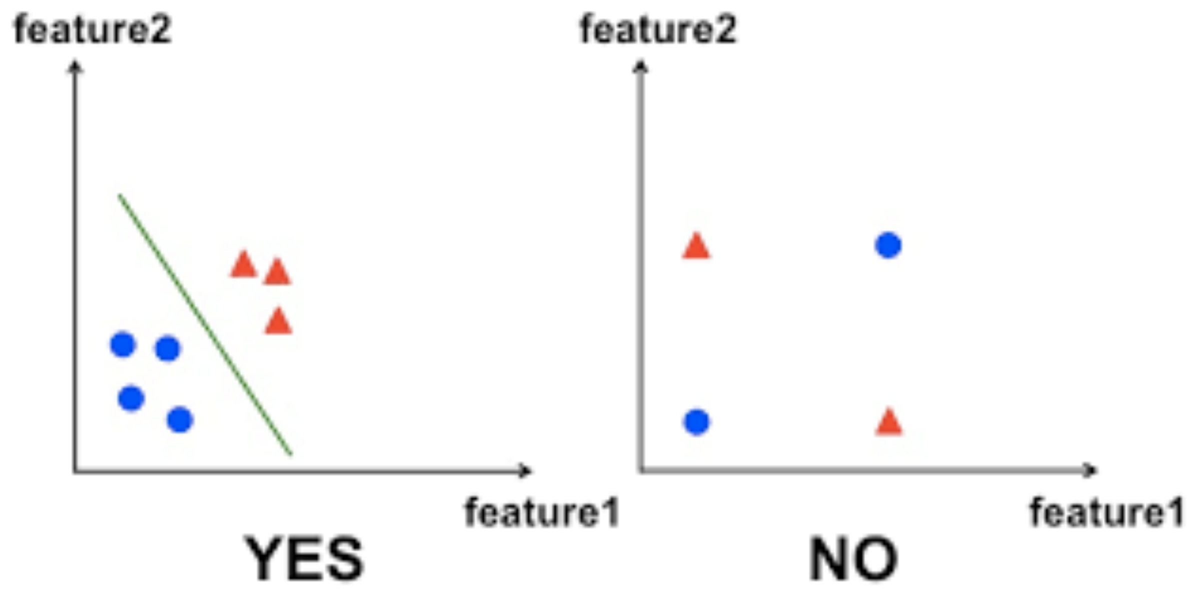
- Define a cost function and use the gradient descent!
- This doesn’t work well!
- The error is piecewise constant.
- Thus the gradient on w is almost zero everywhere!

Perceptron training: 2nd try perceptron criterion

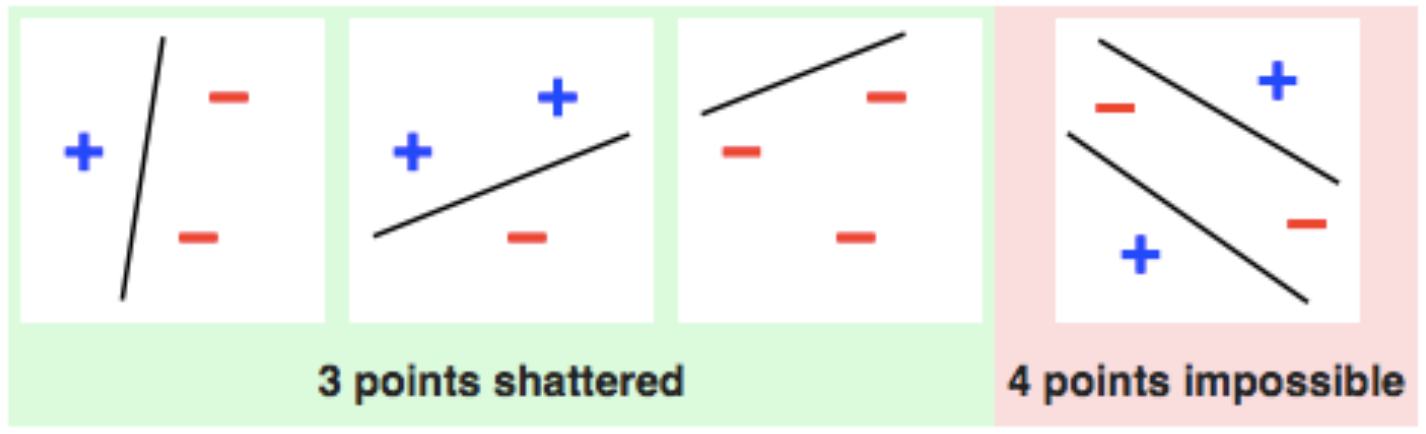




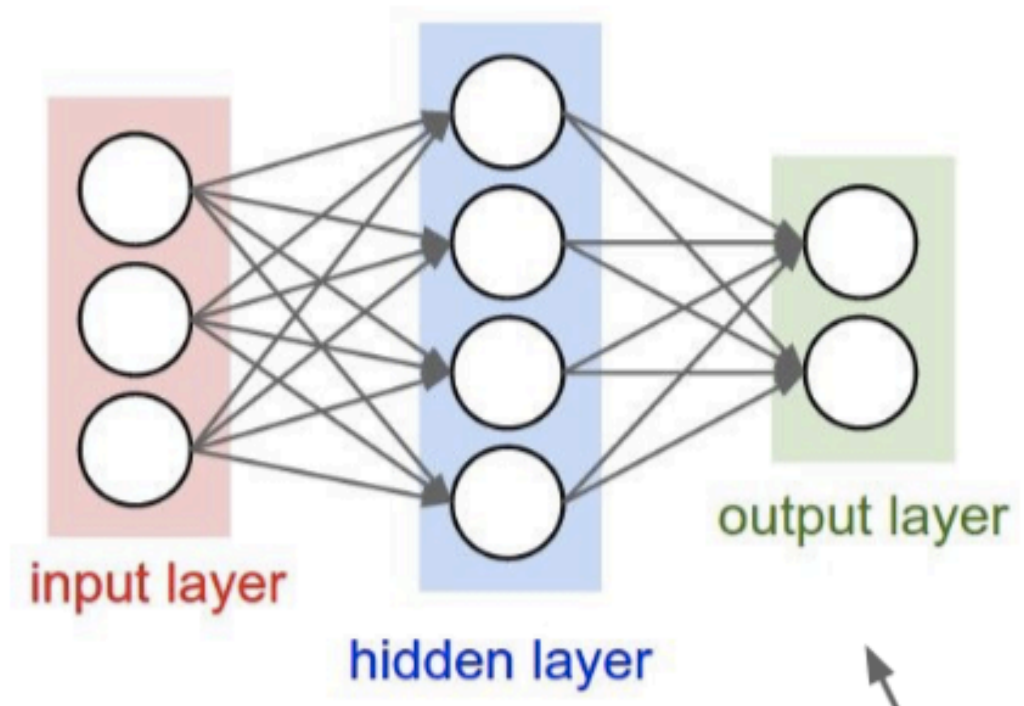
Perceptron is limited, Vapnik-Chervonenkis dimension (VC)

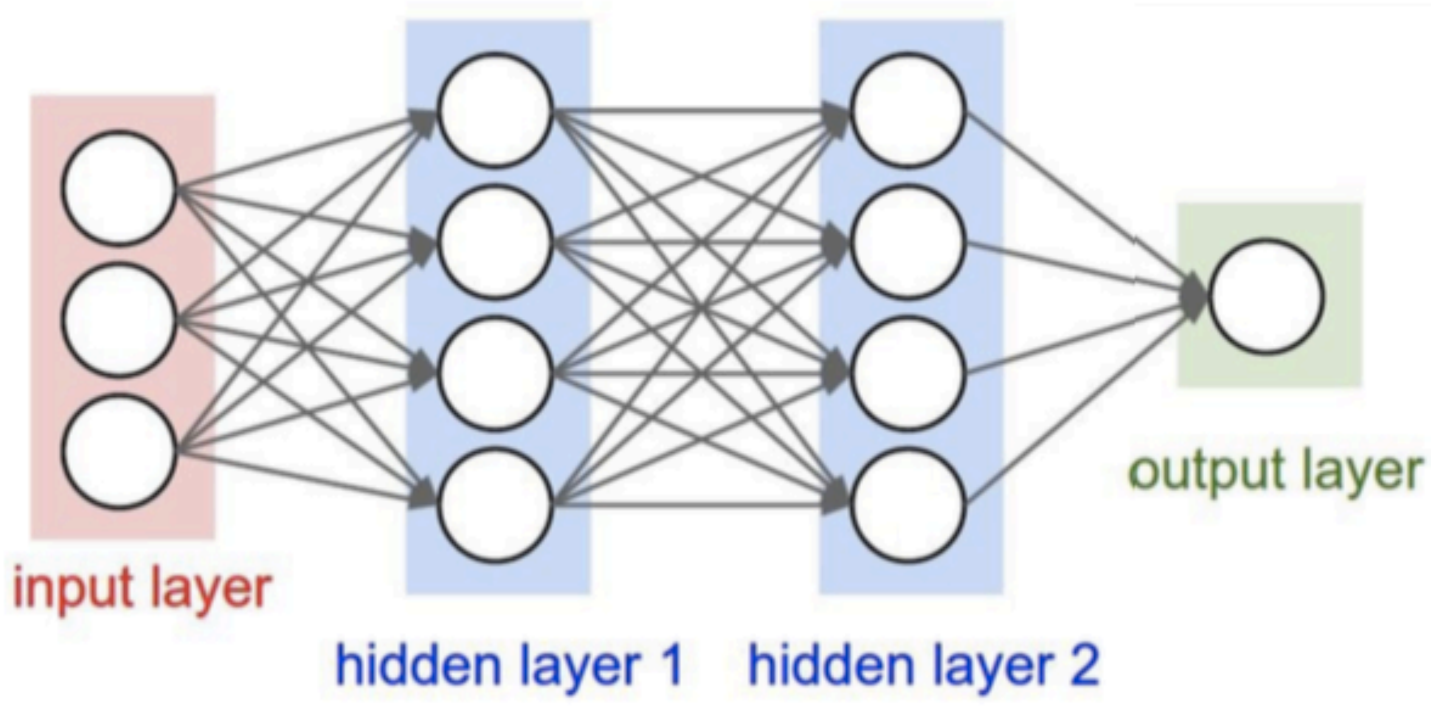


VC dimension basically states the power of your classifier.



Multilayer Perceptron





"2-layer Neural Net", or
"1-hidden-layer Neural Net"

"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

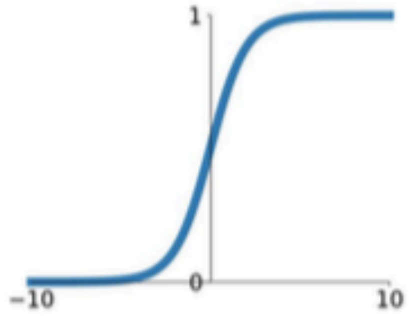
Evaluation is fast!

Activation functions

Activation functions

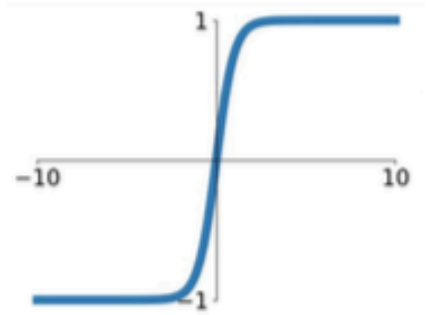
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



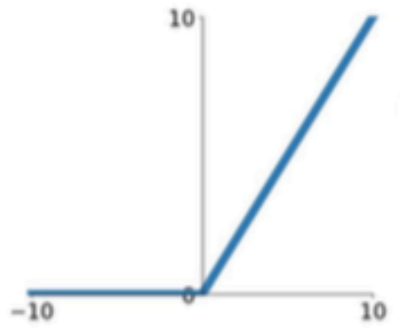
tanh

$\tanh(x)$



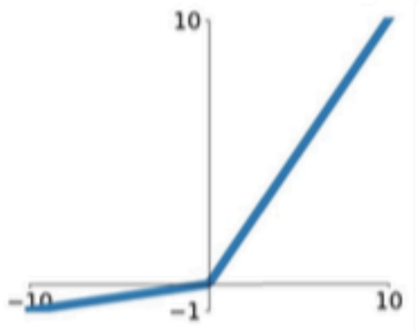
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

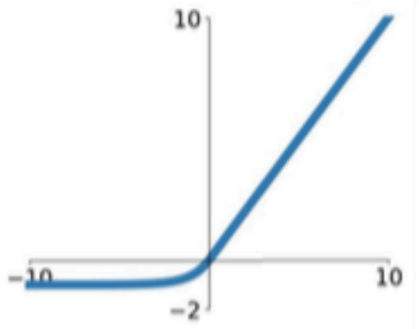


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

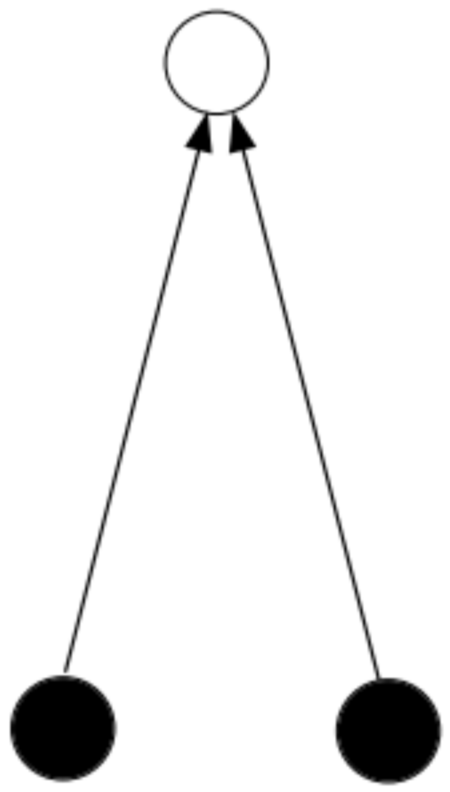
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



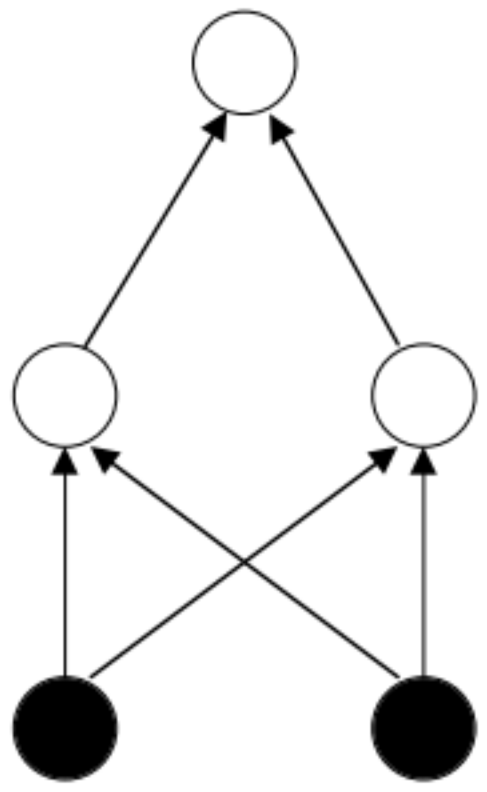
Examples of Network Architectures

Single Layer Feed-forward



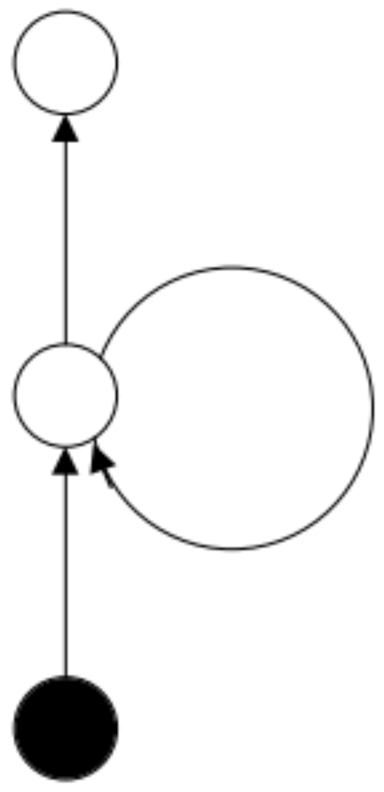
Single-Layer Perceptron

Multi-Layer Feed-forward



Multi-Layer Perceptron

Recurrent Network

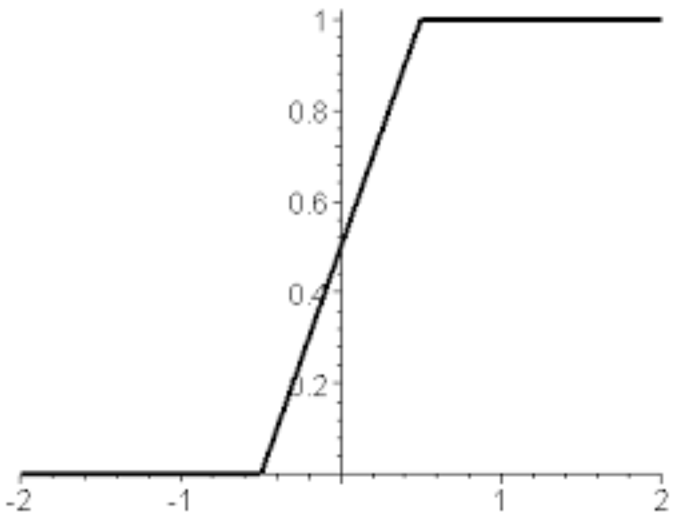


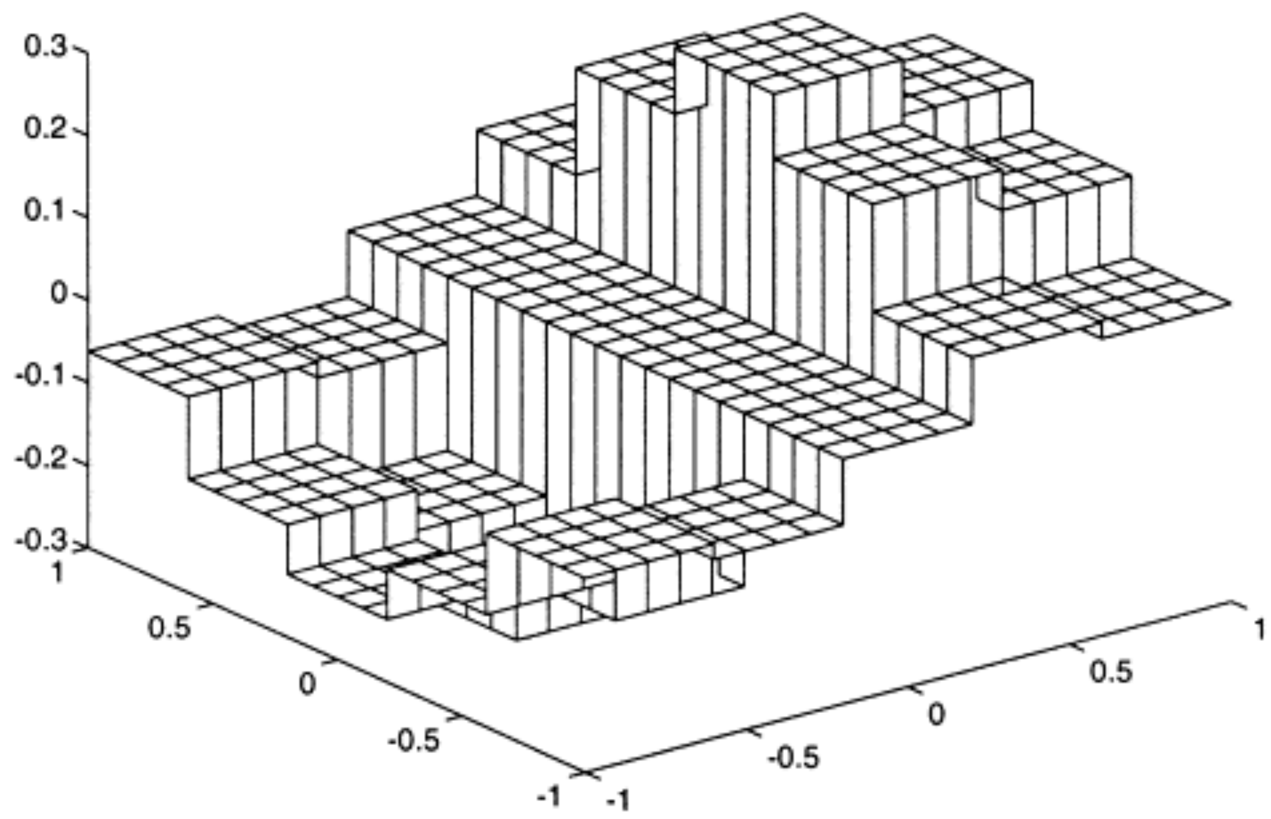
Simple Recurrent Network

Universal approximation theorem(the statement)

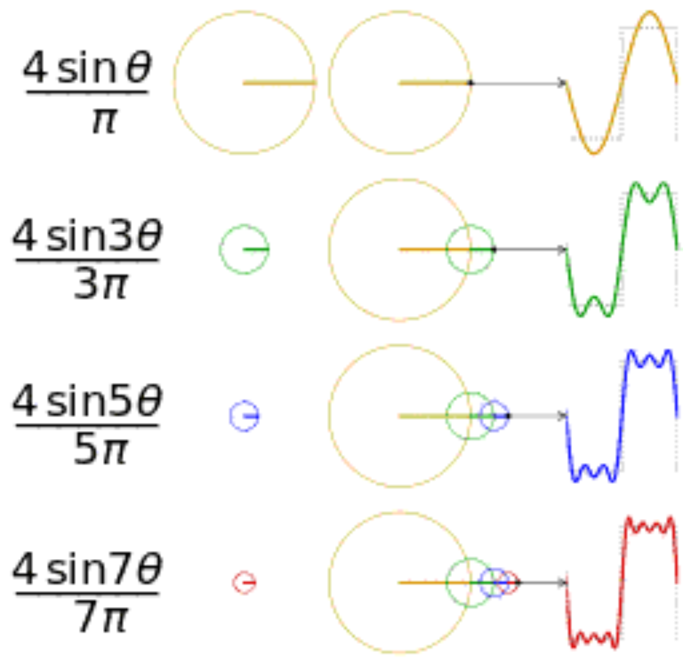
- With a nonconstant, bounded, and continuous activation function.
- A multilayer forward perceptron can be an approximation of a function f on a compact set.
- If it’s not working, then
- not enough neurons.
- not learn enough, a.k.a. more training.
- lack of a determinstic relationship between input and output.

Universal approximation theorem(prove by hand waving)

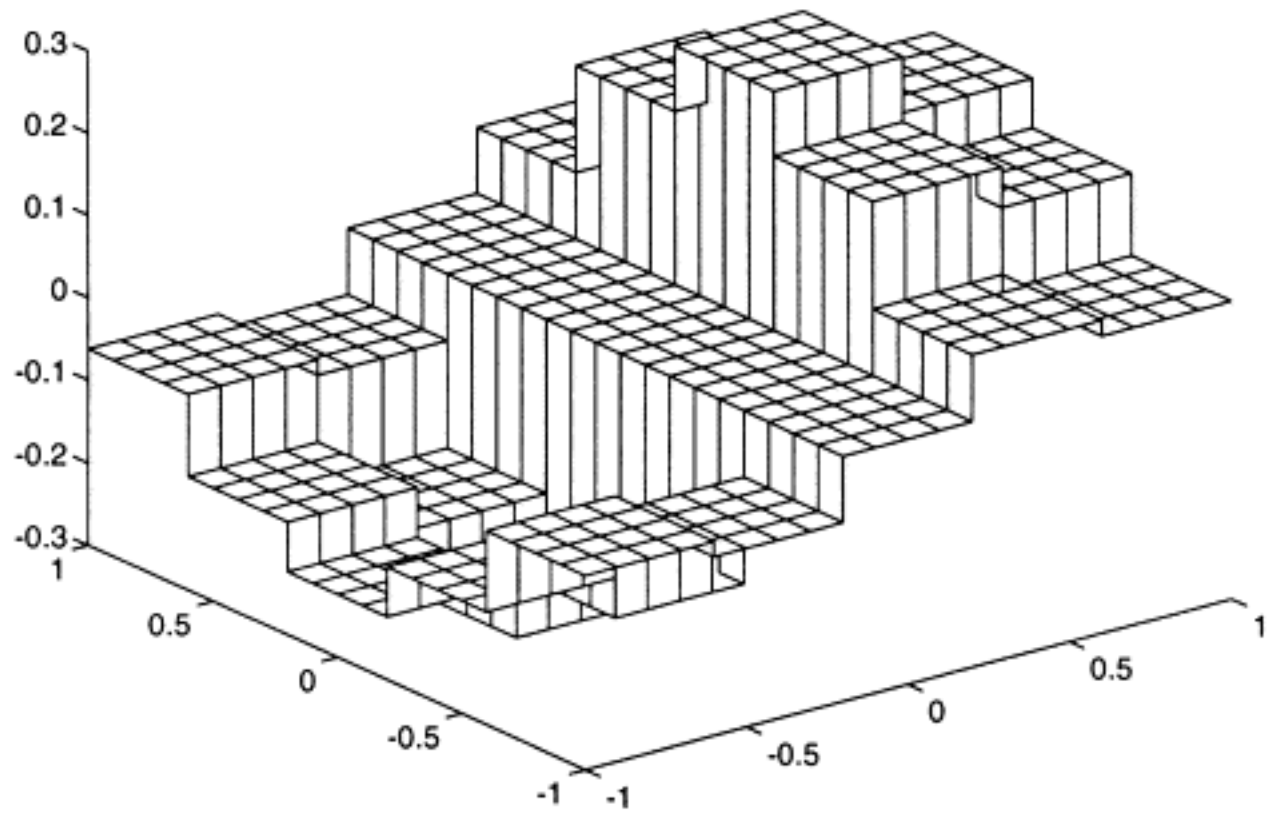




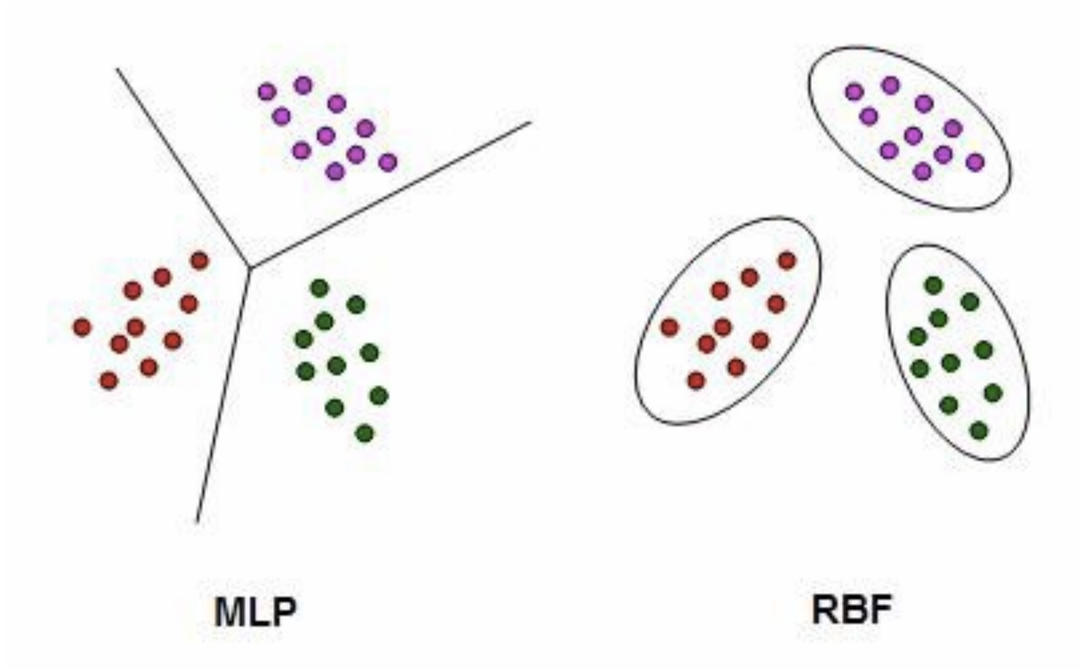
Universal approximation theorem(prove by hand waving+FT)



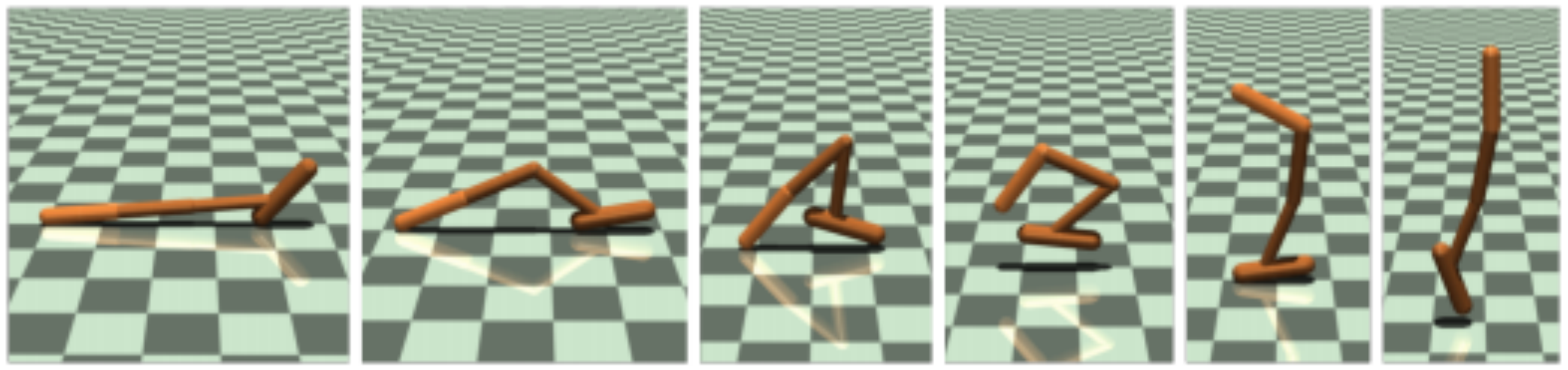
$\frac{4\sin\theta}{\pi}$
$\frac{4\sin3\theta}{3\pi}$
$\frac{4\sin5\theta}{5\pi}$
$\frac{4\sin7\theta}{7\pi}$



Radial basis function(RBF) neural network



$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \varphi(\|\mathbf{x} - \mathbf{x}_i\|),$$



Towards Generalization and Simplicity in Continuous Control, 2018

RBFs

1. **Gaussian Functions:**

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

width parameter $\sigma > 0$

2. **Multi-Quadric Functions:**

$$\phi(r) = \left(r^2 + \sigma^2\right)^{1/2}$$

parameter $\sigma > 0$

3. **Generalized Multi-Quadric Functions:**

$$\phi(r) = \left(r^2 + \sigma^2\right)^{\beta}$$

parameters $\sigma > 0,\ 1 > \beta > 0$

Self organize map (SOM)

- Unsupervised system /competitive learning.
- Only one output neuron is activated at any one time.
- Such competition can be induced/implemented by having lateral inhibition connections (negative feedback paths) between the neurons.
- The result is that the neurons are forced to organize themselves. For obvious reasons, such a network is called a Self Organizing Map (SOM).

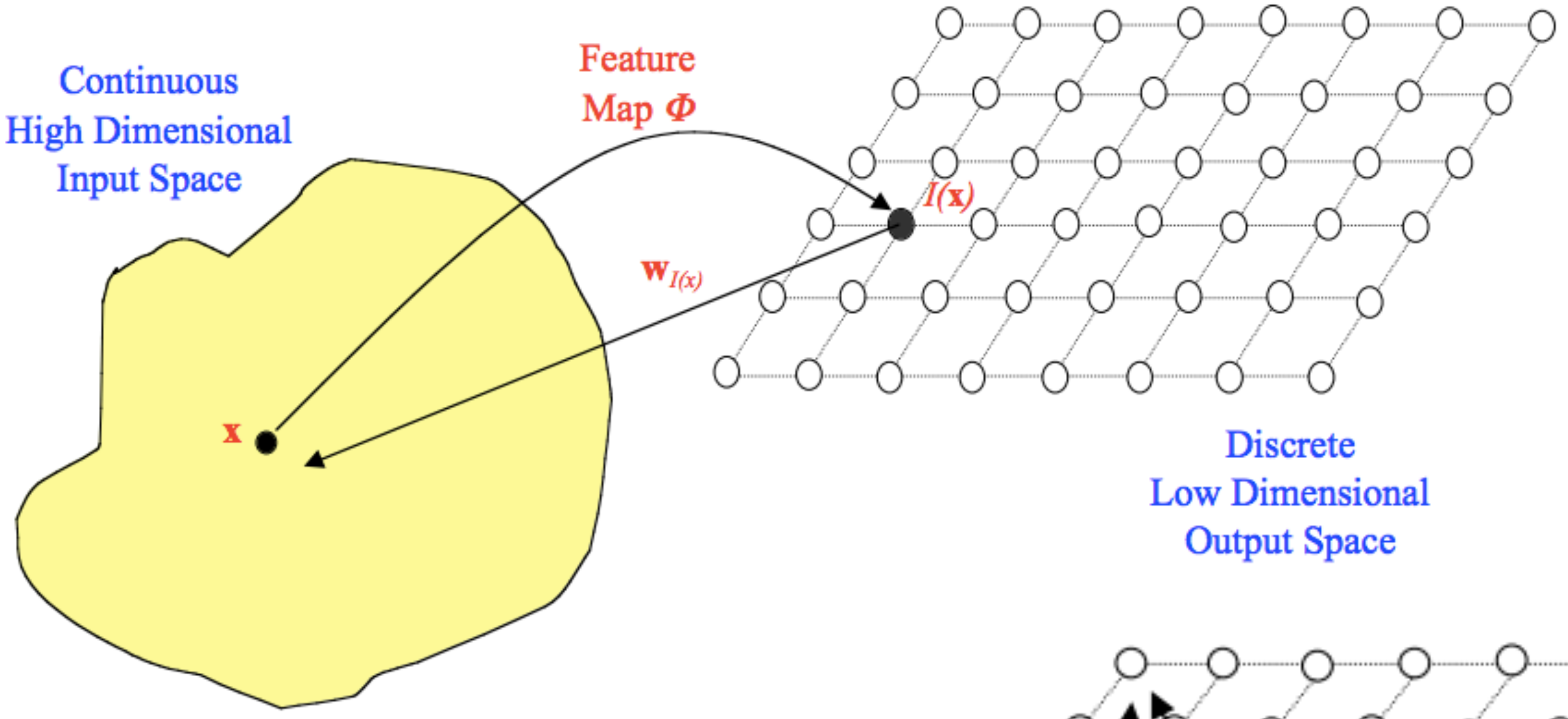
SOM (bio)

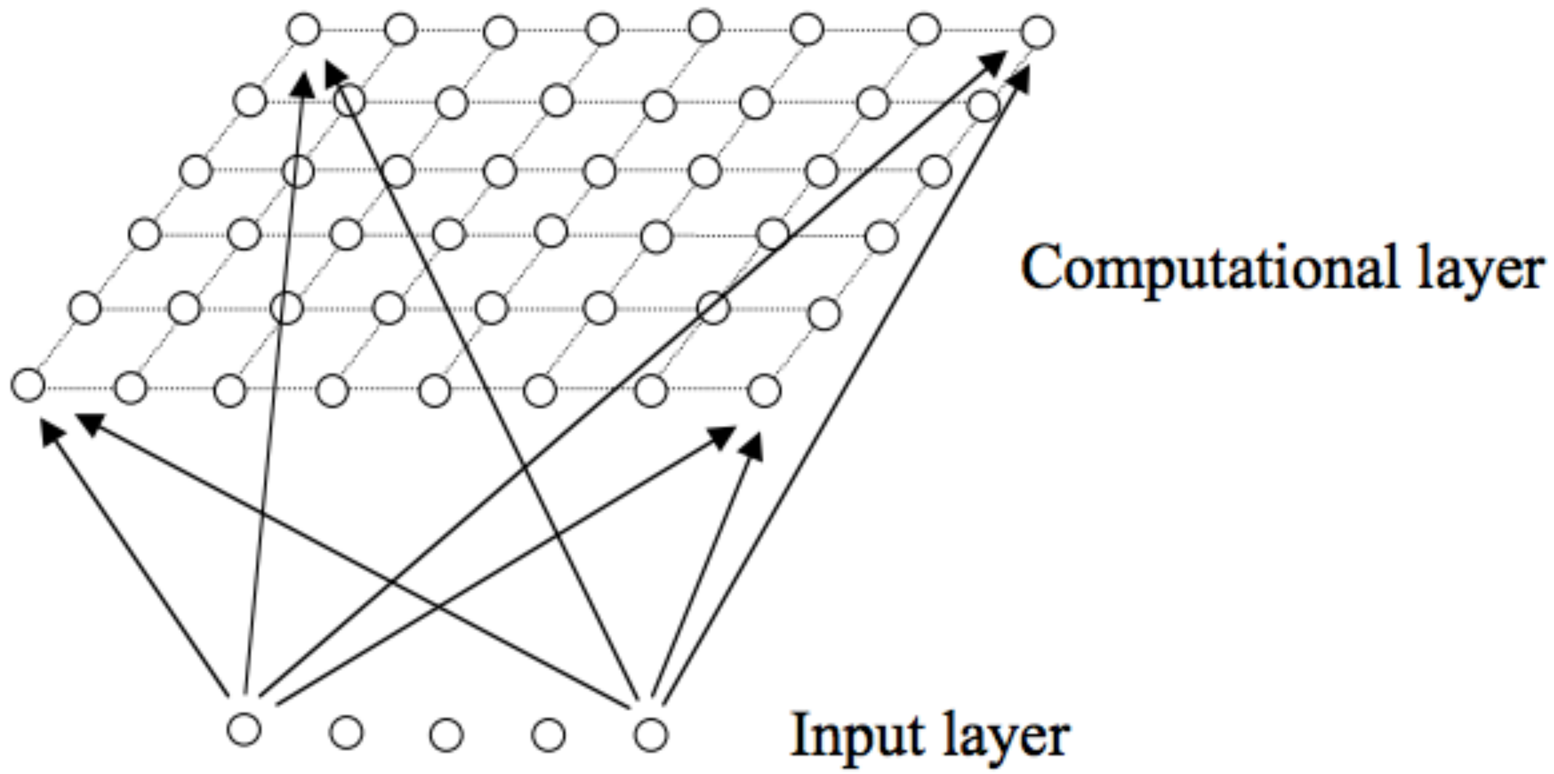
- Neurobiological studies indicate that different sensory inputs (motor, visual, auditory, etc.) are mapped onto corresponding areas of the cerebral cortex in an orderly fashion.

- At each stage of representation, or processing, each piece of incoming information is kept in its proper context/neighbourhood.

- Neurons dealing with closely related pieces of information are kept close together so that they can interact via short synaptic connections.

SOM (Kohonen Net.)





SOM (learn)

- Initialization: All the connection weights are initialized with small random values.
- Competition: For each input pattern, the neurons compute their respective values of a discriminant function which provides the basis for competition. The particular neuron with the smallest value of the discriminant function is declared the winner.
- Cooperation: The winning neuron determines the spatial location of a topological neighbourhood of excited neurons, thereby providing the basis for cooperation among neighbouring neurons.
- Adaptation: The excited neurons decrease their individual values of the discriminant function in relation to the input pattern through suitable adjustment of the associated connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

SOM (metric)

- We can then define our discriminant function to be the squared Euclidean distance between the input vector x and the weight vector wj for each neuron j.

- In other words, the neuron whose weight vector comes closest to the input vector (i.e. is most similar to it) is declared the winner.

$$d_j(\mathbf{x}) = \sum_{i=1}^{D} (x_i - w_{ji})^2$$

SOM (cooperative process)

- In neurobiological studies we find that there is lateral interaction within a set of excited neurons.
- There is a topological neighbourhood that decays with distance.
- If Sij is the lateral distance between neurons i and j on the grid of neurons, we take

$$T_{j,I(\mathbf{x})} = \exp(-S^2_{j,I(\mathbf{x})} / 2\sigma^2)$$

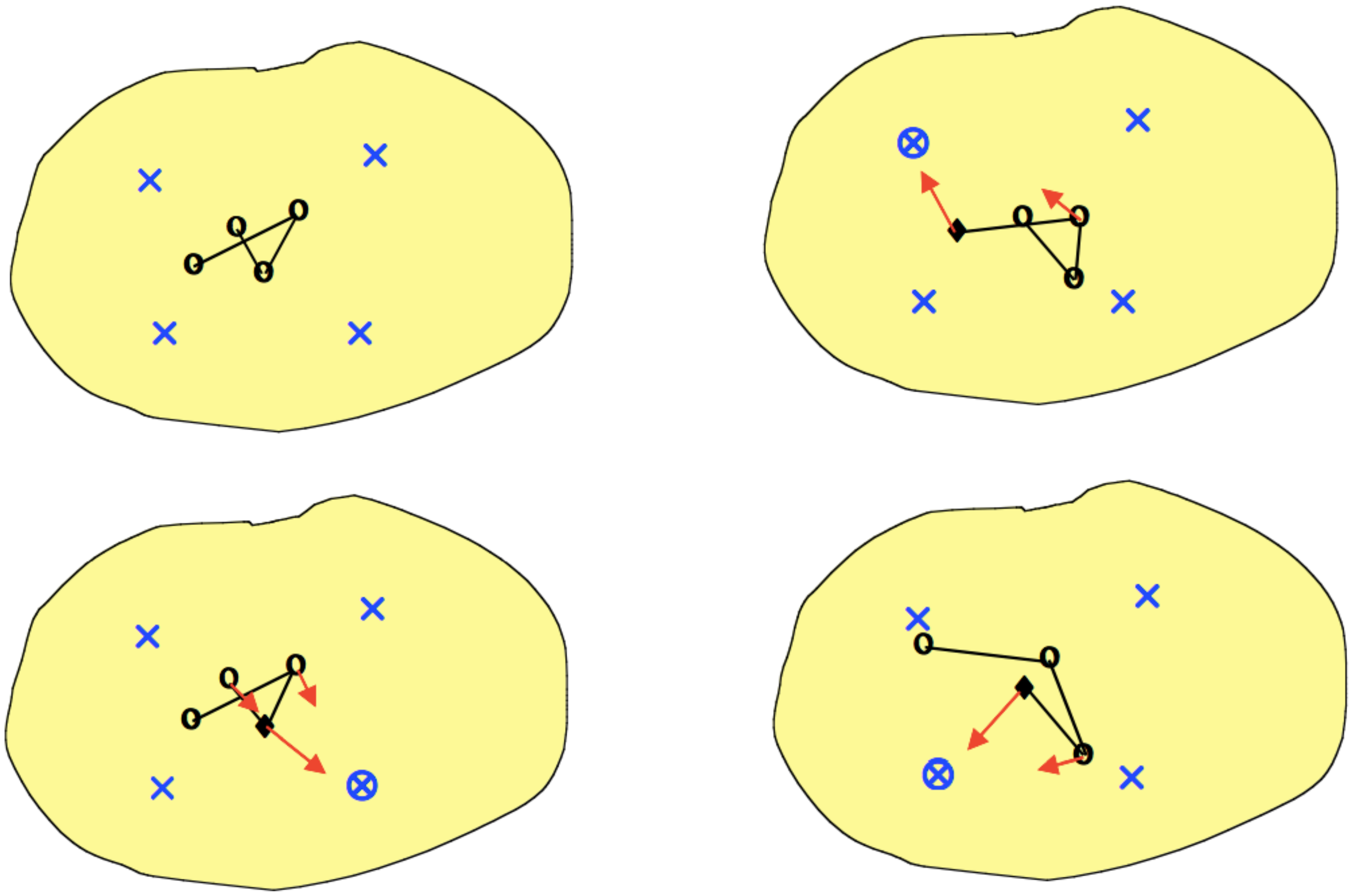
- as our topological neighbourhood, where I(x) is the index of the winning neuron.
- it is symmetrical about that neuron.
- it is translation invariant.
- And we need a time decay $\sigma(t) = \sigma_0 \exp(-t / \tau_\sigma)$

SOM(adaptive)

$$\Delta w_{ji} = \eta(t) \,.\, T_{j,I(\mathbf{x})}(t) \,.\, (x_i - w_{ji})$$

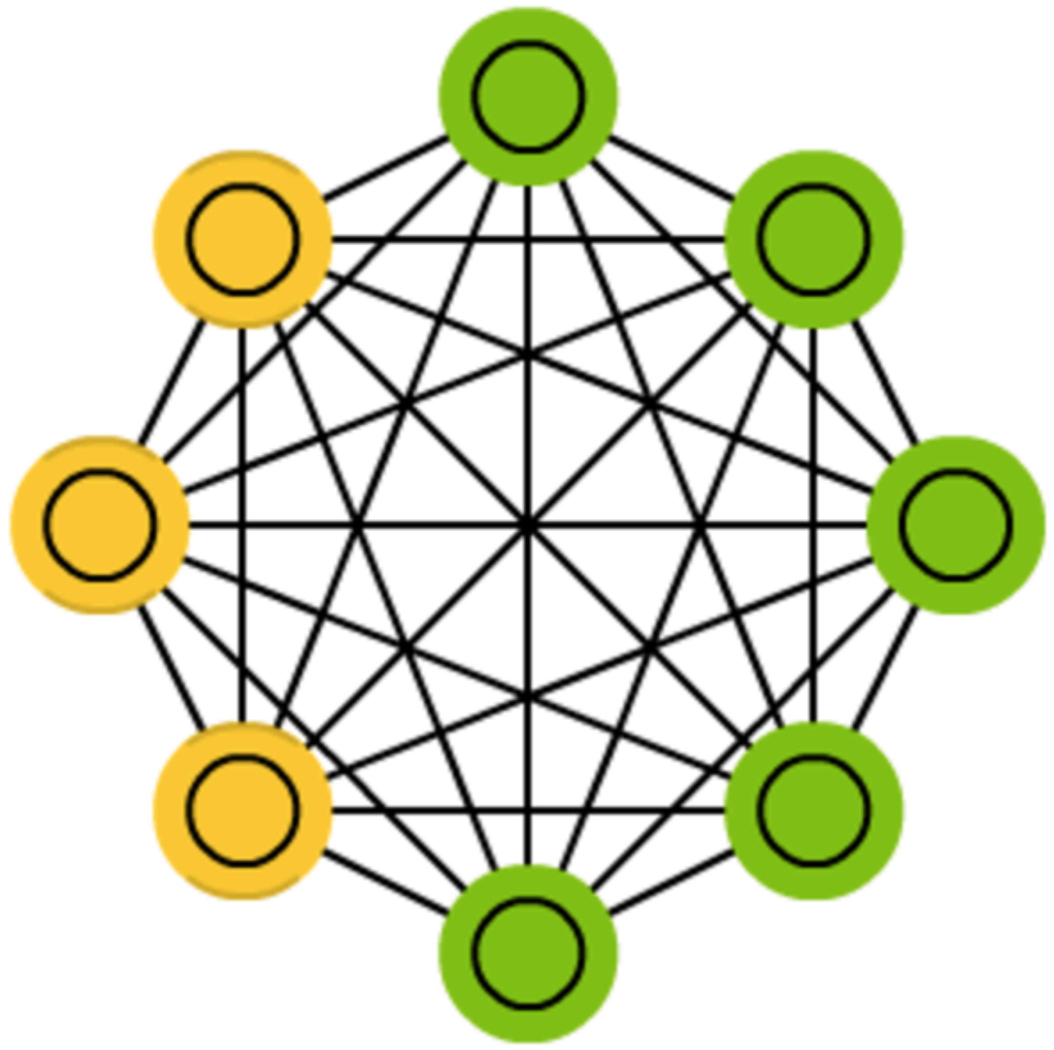
- A learning rate with decay.
- A neighboring mask with decay.
- A dissimilarity measurement.

SOM(an example)



Boltzmann machine

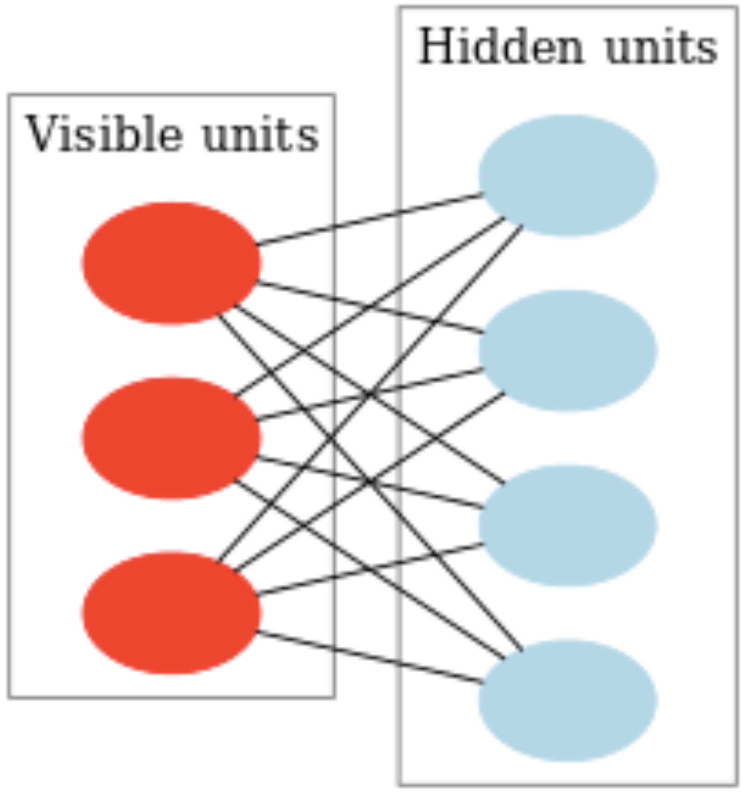
Boltzmann Machine (BM)



$$E = -\left(\sum_{i<j} w_{ij}\, s_i\, s_j + \sum_i \theta_i\, s_i \right)$$

Hopfield network,
fixed value

Restricted Boltzmann machine



$$P(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i=1}^{d} P(v_i \mid \boldsymbol{h}) \; ,$$

$$P(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j=1}^{q} P(h_j \mid \boldsymbol{v}) \; .$$

$$\Delta w = \eta \left(\boldsymbol{v}\boldsymbol{h}^{\top} - \boldsymbol{v}'\boldsymbol{h}'^{\top} \right)$$

Reading

- McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5, no. 4 (1943): 115-133.

- Minsky, Marvin, and Seymour Papert. "Perceptron: an introduction to computational geometry." The MIT Press, Cambridge, expanded edition 19, no. 88 (1969): 2.

- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2, no. 5 (1989): 359-366.

- Goodfellow, Ian J., David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. "Maxout networks." arXiv preprint arXiv:1302.4389 (2013).

- Fischer, Asja, and Christian Igel. "Training restricted Boltzmann machines: An introduction." Pattern Recognition 47, no. 1 (2014): 25-39.

Experiment 1

- Option 1: Code your own MLP and back-propagation.
- Option 2: Use Pytorch to train a classifier on cifar10.
- Option 3: Compare different activation function on MNIST for its performance.
- Option 4: Compare different 1st order optimizer on MNIST.

Experiment 1

- You are on your own.
- Code should be posted on Github.
- An digital experiment report by March 18th’s morning.
- The report’s file name should be like YOURNAME_ID.pdf.
- The report should be written in English.
- A link to the code should be presented in the report below the title and your name, above the abstract. (No cover page plz.)

End