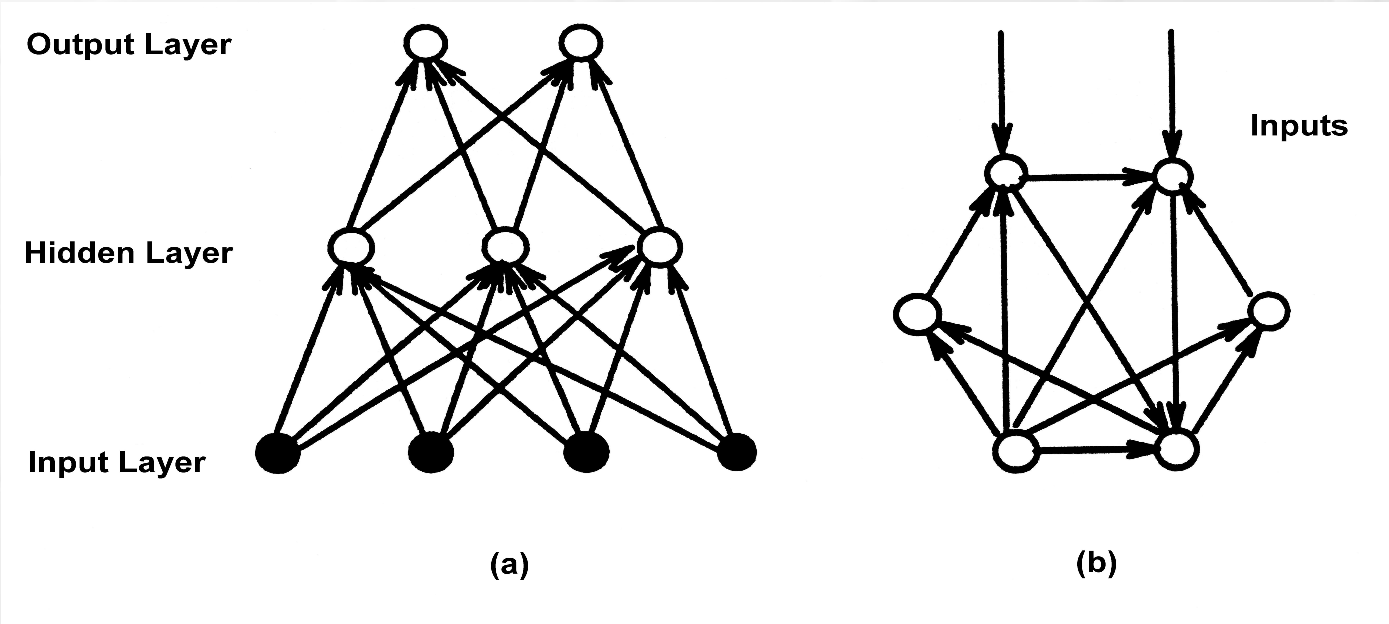
Network Topologies / Architectures

- Feedforward only vs. Feedback loop (Recurrent networks)
- Fully connected vs. sparsely connected
- Single layer vs. multilayer

Multilayer perceptrons, Hopfield network,
Boltzman machines, Kohonen network



Classification Problems

Given :

1) some "features" $(f_1, f_2, \ldots, f_n)$

2) some "classes" $(c_1, \ldots, c_m)$

Problem :

To classify an "object" according to its features

Example # 1

To classify an "object" as :

c_1 = " watermelon "

c_2 = " apple "

c_3 = " orange "

According to the following features :

f_1 = " weight "

f_2 = " color "

f_3 = " size "

Example :

weight = 80g

color = green

size = $10\ cm^3$

→ " **apple** "

Example # 2

Problem : Establish whether a patient got the flu

- Classes : { “ flu ” , “ non-flu ” }
- (Potential) Features :

f_1 : Body temperature

f_2 : Headache ? (yes / no)

f_3 : Throat is red ? (yes / no / medium)

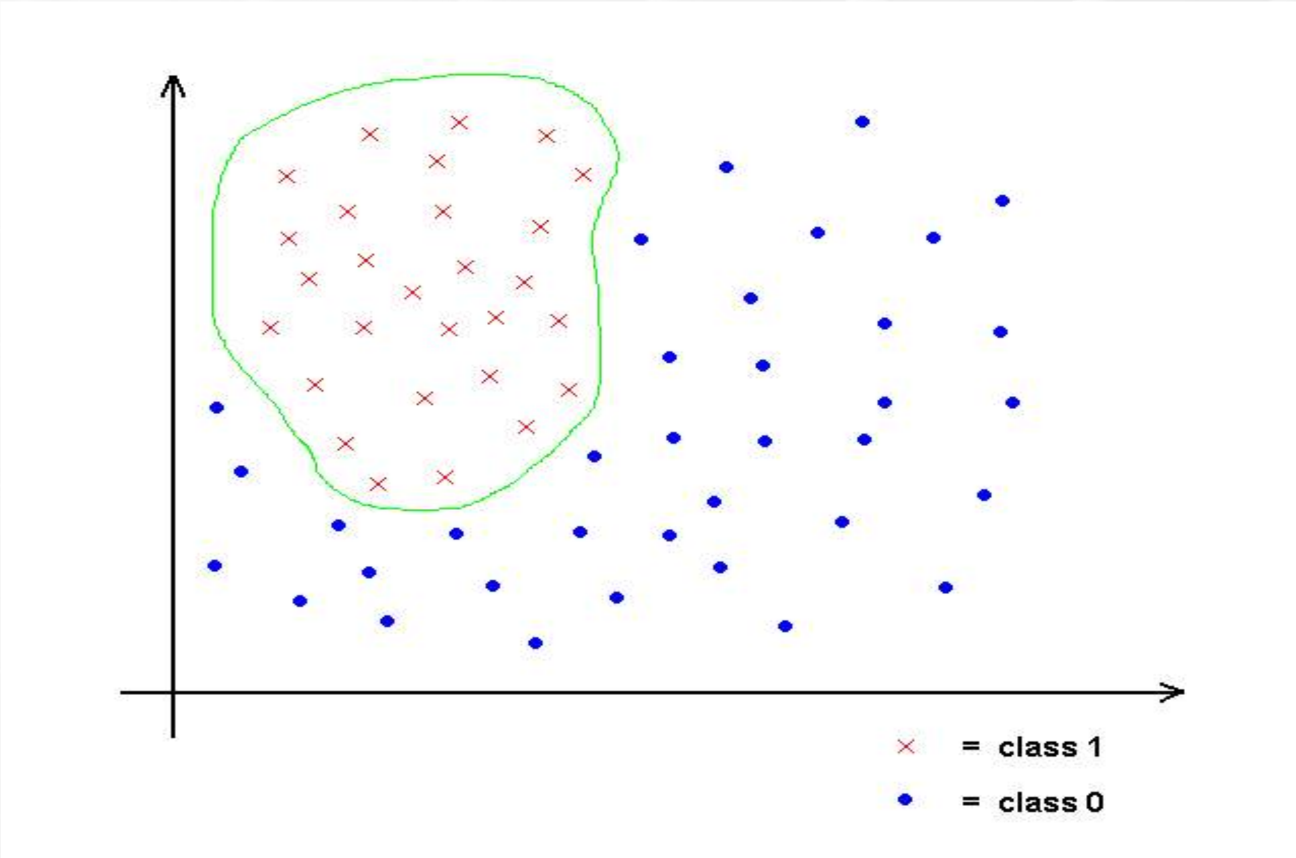
f_4 :

Example # 3

Classes = { 0 , 1 }

Features = x , y : both taking value in [0 , +∞ [

Idea : Geometric Representation



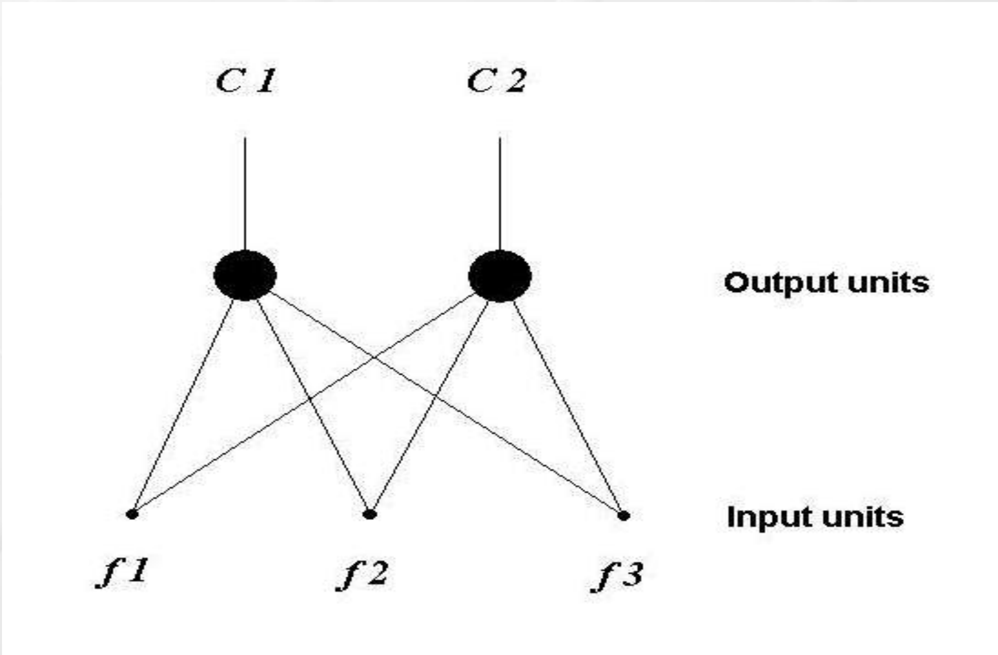
Neural Networks for Classification

A neural network can be used as a classification device .

Input ≡ features values

Output ≡ class labels

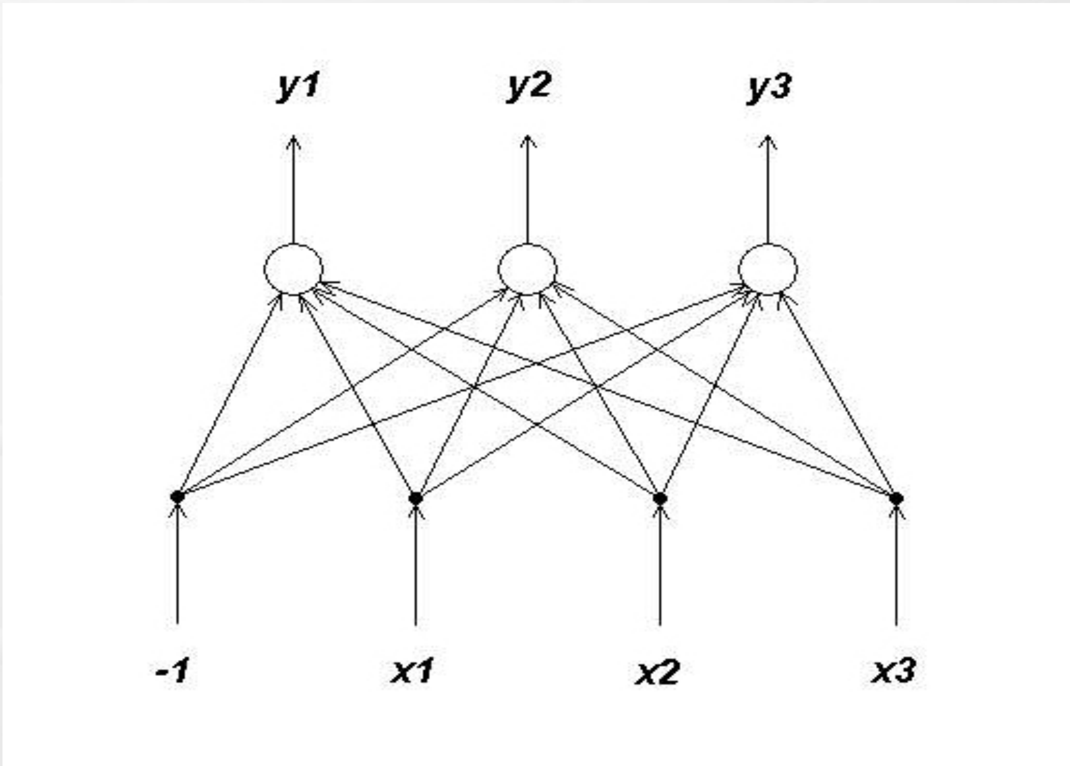
Example : 3 features , 2 classes



Thresholds

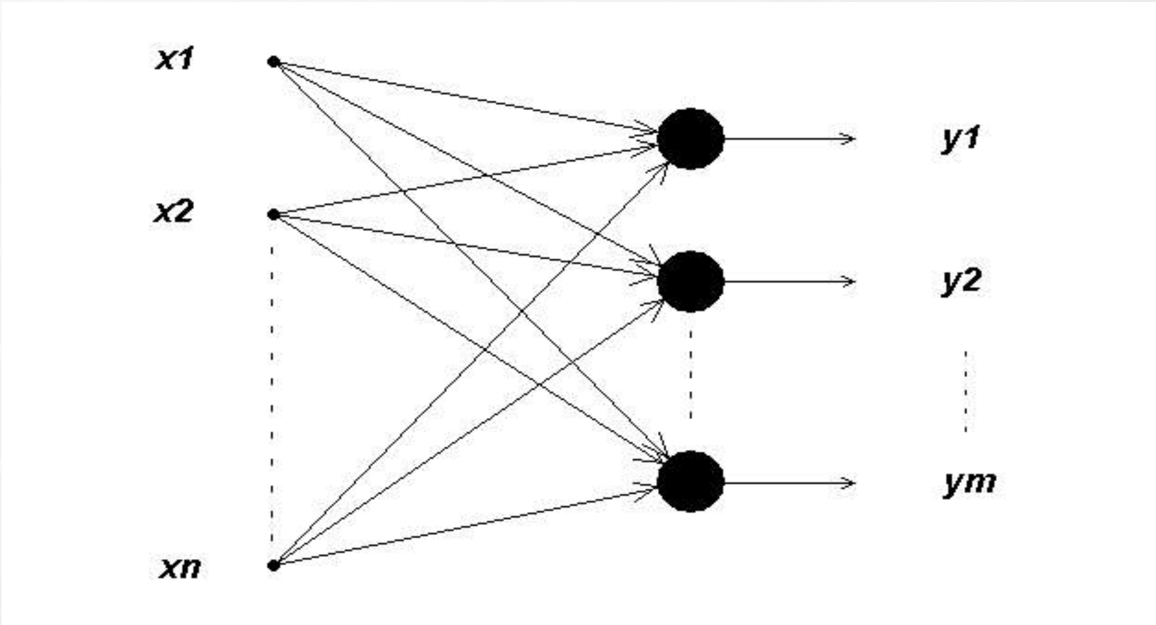
We can get rid of the thresholds associated to neurons by adding an extra unit permanently clamped at -1 .

In so doing, thresholds become weights and can be adaptively adjusted during learning.



Simple Perceptrons

A network consisting of one layer of M&P neurons connected in a feedforward way (i.e. no lateral or feedback connections).

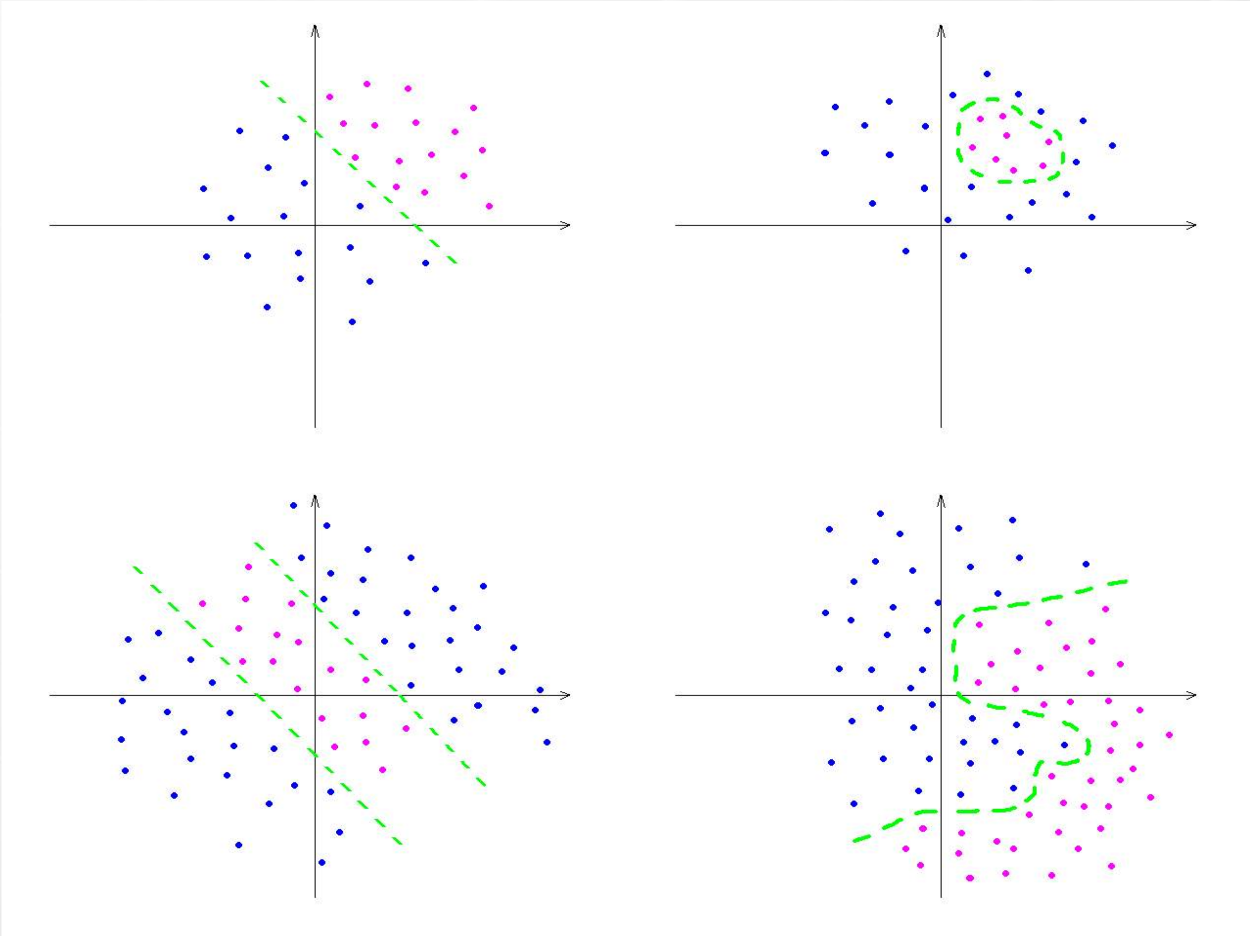


- Capable of "learning" from examples (Rosenblatt)
- They suffer from serious computational limitations (Minsky and Papert, 1969)

Decision Regions

It's an area wherein all examples of one class fall .

Examples :

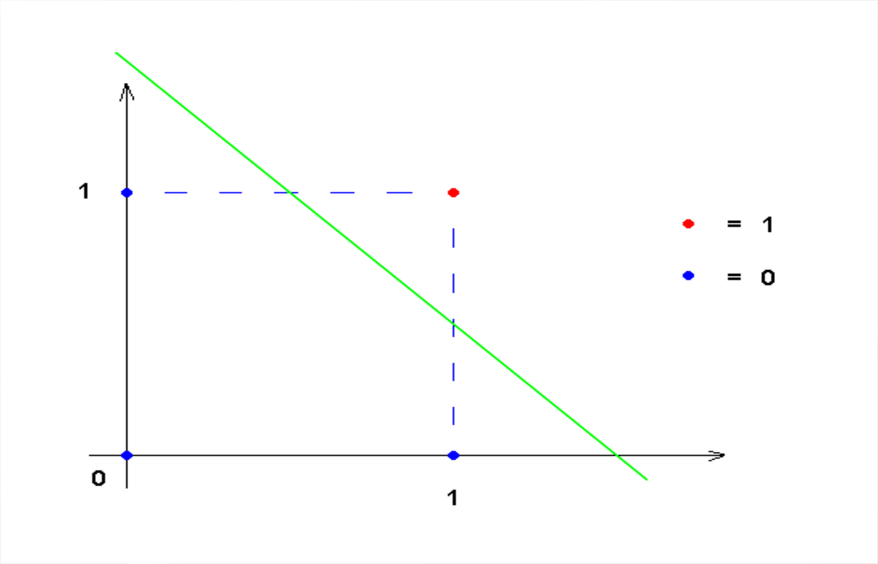


Linear Separability

A classification problem is said to be *linearly separable* if the decision regions can be separated by a hyperplane .

Example : AND

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

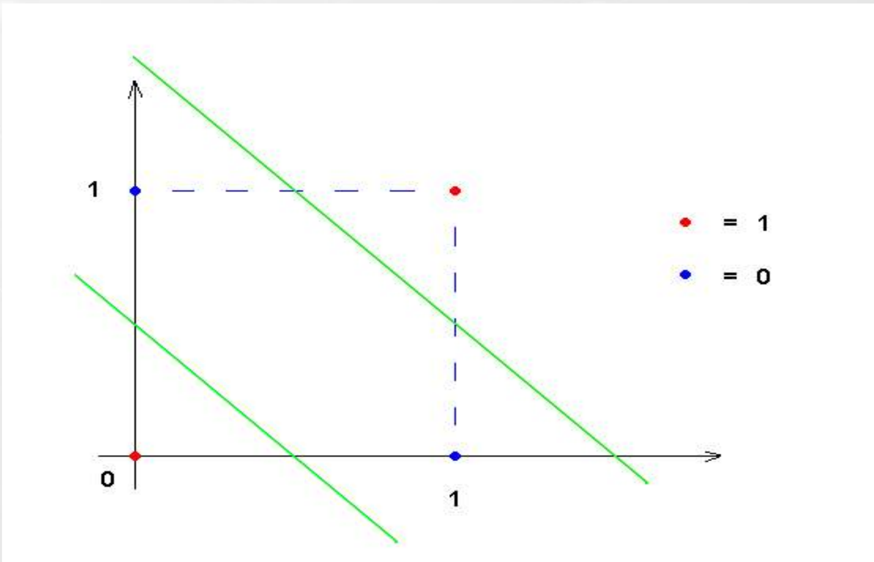


Limitations of Perceptrons

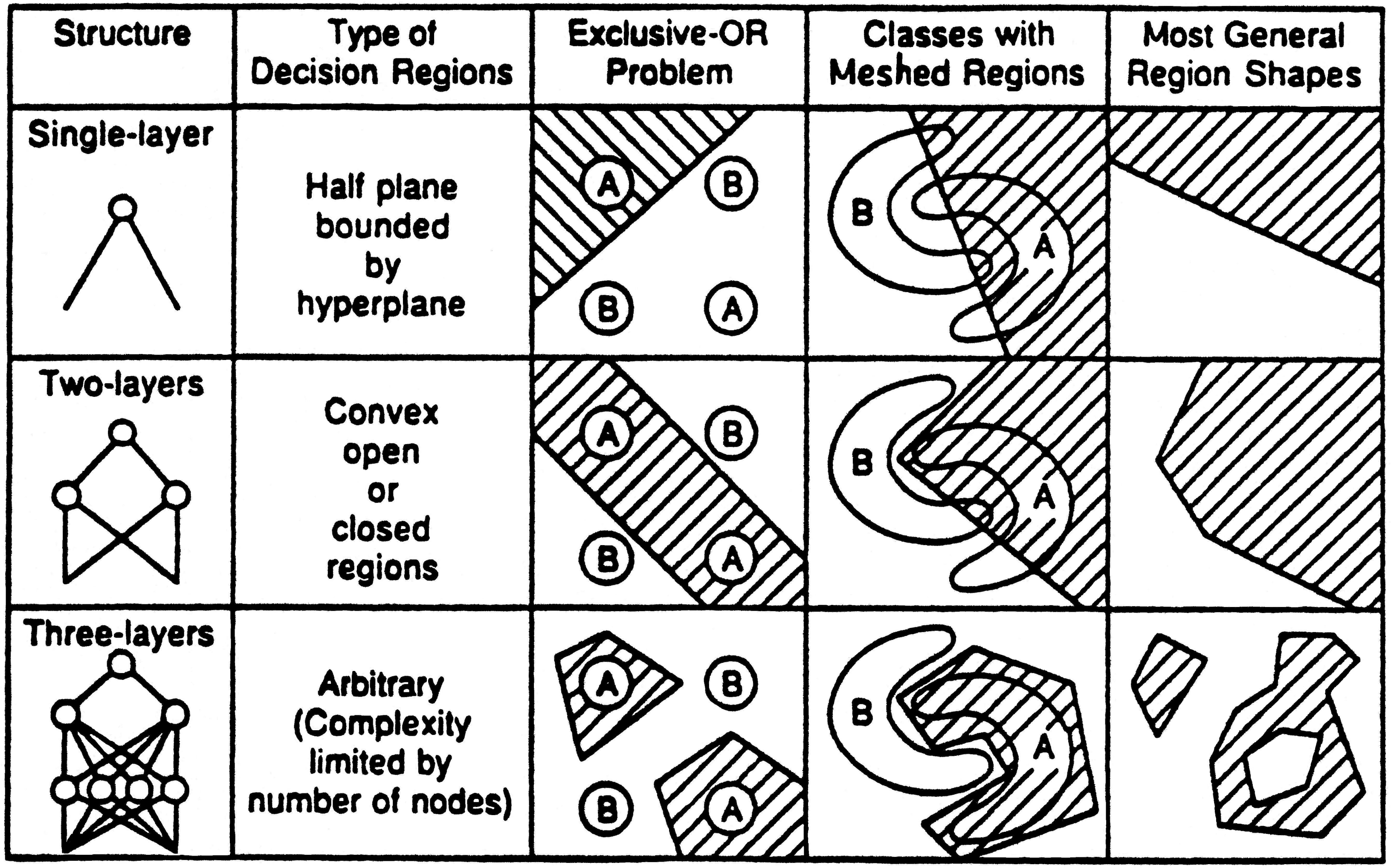
It has been shown that perceptrons can only solve linearly separable problems (Minsky and Papert , 1969) .

Example : XOR (exclusive OR)

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0



A View of the Role of Units



Convergence of Learning Algorithms

- If the problem is linearly separable, then the learning rule converges to an appropriate set of weights in a finite number of steps (Nilsson 1965)
- In practice, one does not know whether the problem is linearly separable or not. So decrease η with the number of iterations, letting $\eta \to 0$.
  The convergence so obtained is artificial and does not necessarily yield a valid weight vector that will classify all patterns correctly
- Some variations of the learning algorithm, e.g. Pocket algorithm, (Gallant, 1986)

Multi–Layer Feedforward Networks

- Limitation of simple perceptron: can implement only linearly separable functions
- Add “ hidden ” layers between the input and output layer. A network with just one hidden layer can represent any Boolean functions including XOR
- Power of multilayer networks was known long ago, but algorithms for training or learning, e.g. back-propagation method, became available only recently (invented several times, popularized in 1986)
- Universal approximation power: Two-layer network can approximate any smooth function (Cybenko, 1989; Funahashi, 1989; Hornik, et al.., 1989)
- Static (no feedback)