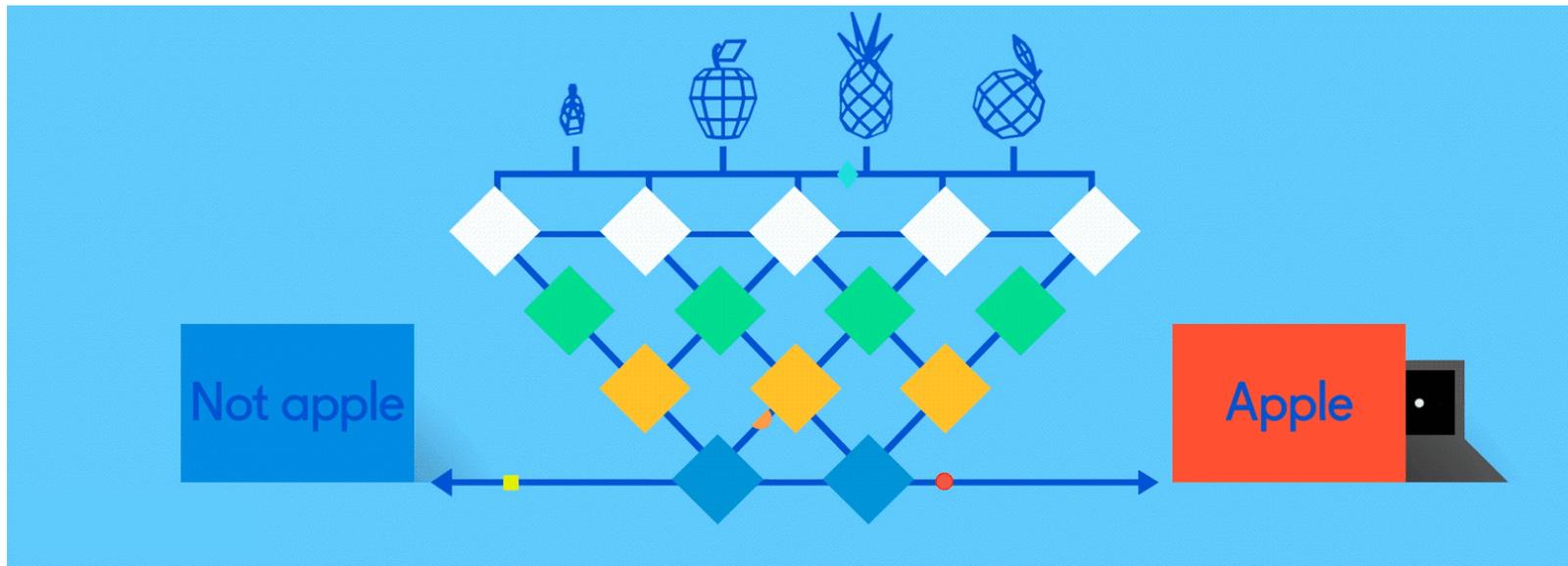


Neural Networks

History



Pitts & McCulloch (1943)

- First mathematical model of biological neurons
- All Boolean operations can be implemented by these neuron-like nodes
- Competitor to Von Neumann model for general purpose computing device
- Origin of automata theory

Hebb (1949)

- Hebbian rule of learning: increase connection strength between neurons i and j whenever both are activated
- Or increase connection strength between nodes i and j whenever both are simultaneously ON or OFF

Early booming (50s – early 60s)

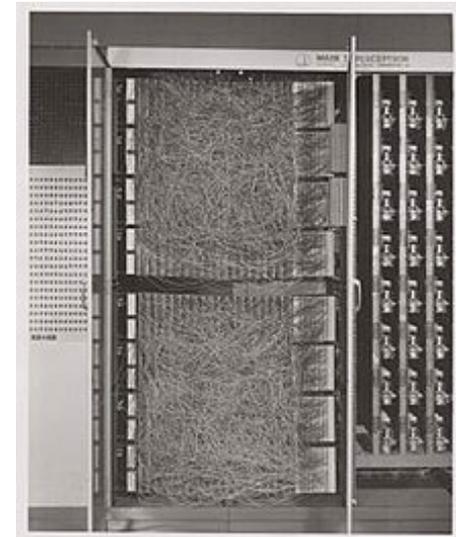
- **Rosenblatt (1958-1960)**

- Perceptron: network of threshold nodes for pattern classification; perceptron learning rule
- Mark 1 perceptron computer

- **Widrow and Hoff (1960, 1962)**

- Learning rule based on [gradient descent](#) (differentiable unit)

- Minsky tries to build general purpose machine with Pitts/McCulloch units



History: setback in mid 60s – late 70s)

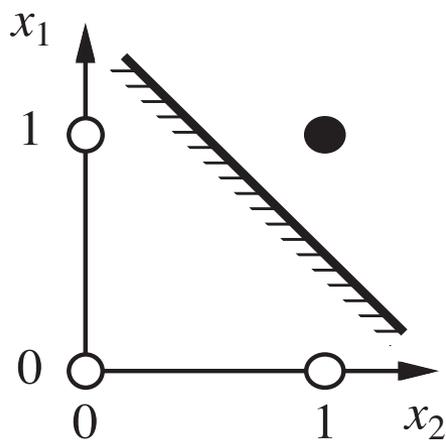
- Serious problems with perceptron model (Minsky's book 1969)
 - Single layer perceptrons cannot represent (learn) simple functions such as XOR
 - Multi-layer of non-linear units may have greater power but there is no learning rule for such nets
 - Scaling problem: connection weights may grow infinitely
 - First two problems overcame by latter effort in 80s, but scaling problem persists
- Death of Rosenblatt (1964)
- Striving of Von Neumann machine and AI

Renewed enthusiasm 80s

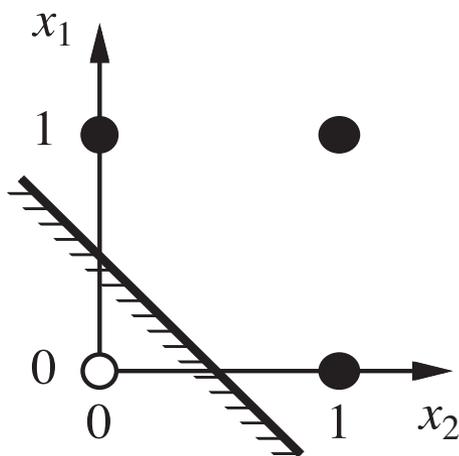
- New techniques
 - Backpropagation for multi-layer feed forward nets (with non-linear, differentiable node functions)
 - Thermodynamic models (Hopfield net, Boltzmann machine ...)
 - Unsupervised learning
- Applications: character recognition, speech recognition, text-to-speech, etc.
- Traditional approaches face difficult challenges
- Caution:
 - Don't underestimate difficulties and limitations
 - Poses more problems than solutions

Not with a perceptron ☹️

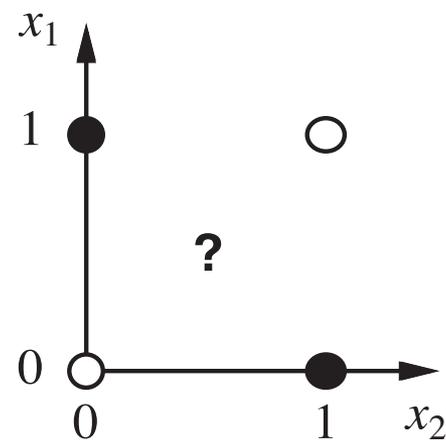
Training examples are not linearly separable for one case: $sum=1$ iff x_1 xor x_2



(a) x_1 and x_2



(b) x_1 or x_2

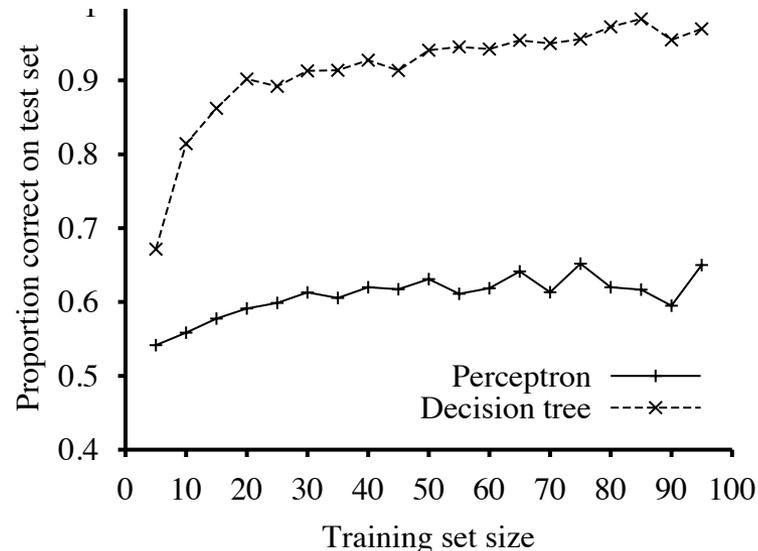
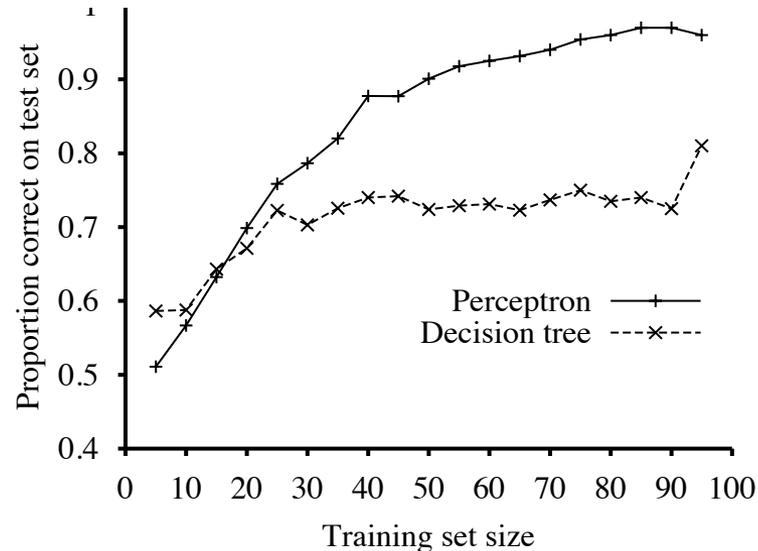


(c) x_1 xor x_2

Works well on some problems

Learning curves

Are majority of inputs 1?



Restaurant example: WillWait?