http://xkcd.com/894/

Neural Networks

David Kauchak
CS30
Spring 2016

Machine Learning is…

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.

WIKIPEDIA
The Free Encyclopedia

Machine Learning is…

Machine learning is programming computers to optimize a performance criterion using example data or past experience.
-- Ethem Alpaydin

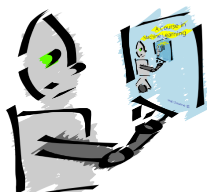
The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest.
-- Kevin P. Murphy

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions.
-- Christopher M. Bishop

Machine Learning is…

Machine learning is about predicting the future based on the past.

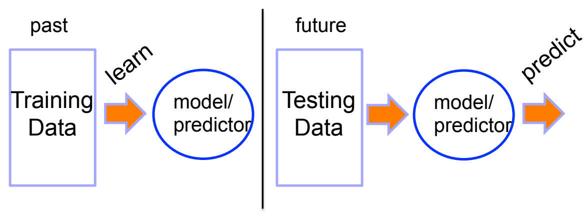
-- Hal Daume III



Machine Learning is…

Machine learning is about predicting the future based on the past.

-- Hal Daume III



Machine Learning, aka

data mining: machine learning applied to "databases", i.e. collections of data

inference and/or *estimation* in statistics

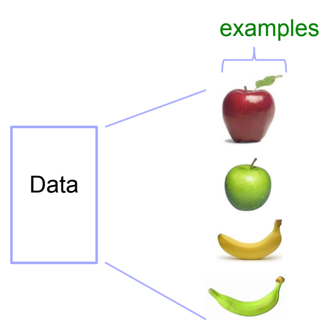
pattern recognition in engineering

signal processing in electrical engineering

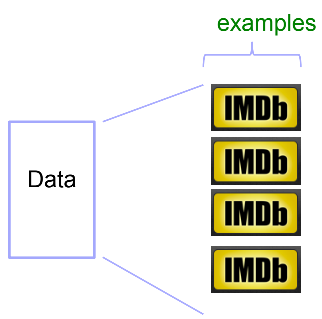
induction

optimization

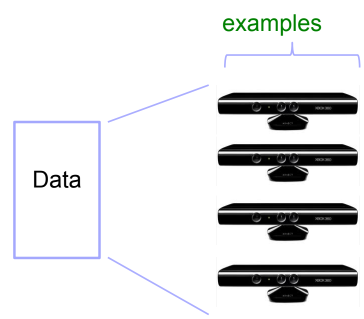
Data



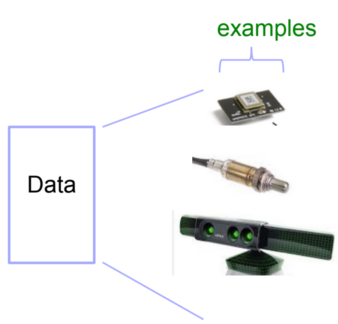
Data



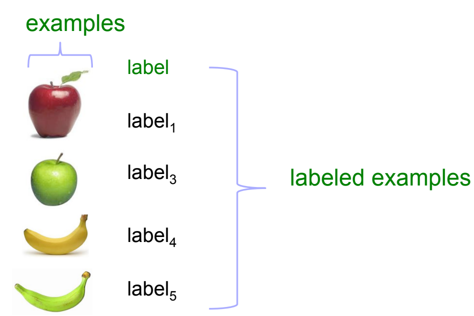
Data



Data

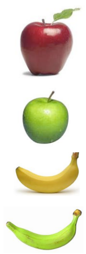


Supervised learning



Supervised learning: given labeled examples

Supervised learning



label

label$_1$

label$_3$

label$_4$

label$_5$

model/ predictor

Supervised learning: given labeled examples

Supervised learning

model/ predictor

predicted label

Supervised learning: learn to predict new example

Supervised learning: classification

label

apple

apple

banana

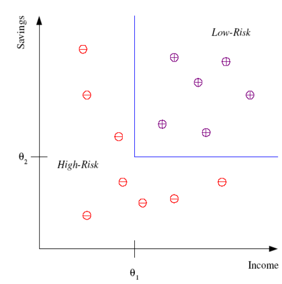
banana

Classification:
a finite set of labels

Supervised learning: given labeled examples

Classification Example

Differentiate between low-risk and high-risk customers from their *income* and *savings*



Classification Applications

Face recognition

Character recognition

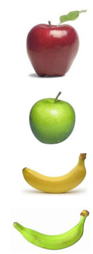
Spam detection

Medical diagnosis: From symptoms to illnesses

Biometrics: Recognition/authentication using physical and/or behavioral characteristics: face, iris, signature, etc

...

Supervised learning: regression



	label
	-4.5
	10.1
	3.2
	4.3

Regression:
label is real-valued

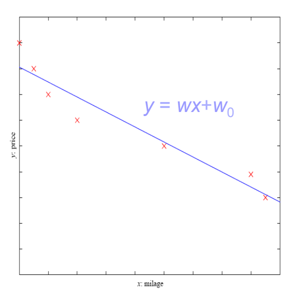
Supervised learning: given labeled examples

Regression Example

Price of a used car

x : car attributes
(e.g. mileage)

y : price



19

Regression Applications

Economics/Finance: predict the value of a stock

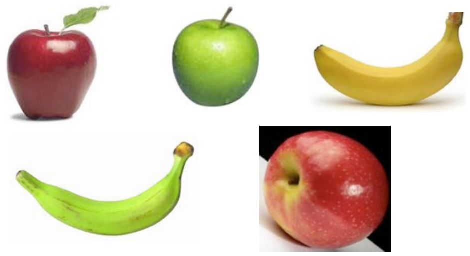
Epidemiology

Car/plane navigation: angle of the steering wheel, acceleration, …

Temporal trends: weather over time

…

Unsupervised learning



Unsupervised learning: given data, i.e. examples, but no labels

Unsupervised learning applications

learn clusters/groups without any label

customer segmentation (i.e. grouping)

image compression

bioinformatics: learn motifs

…

Reinforcement learning

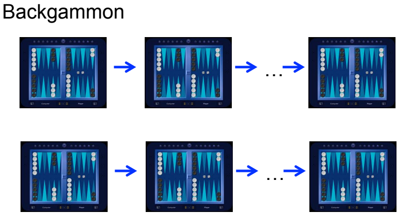
left, right, straight, left, left, left, straight	**GOOD**
left, straight, straight, left, right, straight, straight	**BAD**

left, right, straight, left, left, left, straight	**18.5**
left, straight, straight, left, right, straight, straight	**-3**

Given a *sequence* of examples/states and a *reward* after completing that sequence, learn to predict the action to take in for an individual example/state

Reinforcement learning example

Backgammon



WIN!

LOSE!

Given sequences of moves and whether or not the player won at the end, learn to make good moves

Reinforcement learning example

http://www.youtube.com/watch?v=VCdxqn0fcnE

Other learning variations

What data is available:
- Supervised, unsupervised, reinforcement learning
- semi-supervised, active learning, …

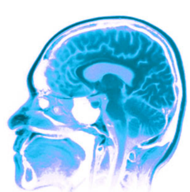
How are we getting the data:
- online vs. offline learning

Type of model:
- generative vs. discriminative
- parametric vs. non-parametric

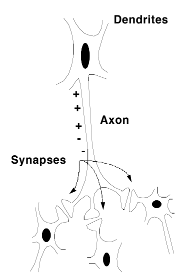
Neural Networks

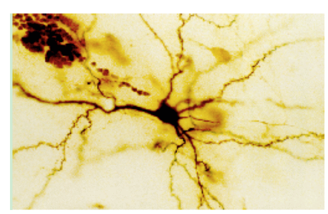
Neural Networks try to mimic the structure and function of our nervous system

People like biologically motivated approaches



Our Nervous System

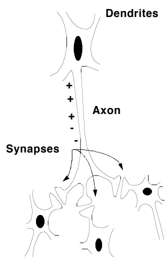




Neuron

What do you know?

Our nervous system: the computer science view

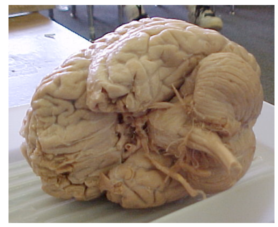


the human brain is a large collection of interconnected neurons

a NEURON is a brain cell

- they collect, process, and disseminate electrical signals
- they are connected via synapses
- they FIRE depending on the conditions of the neighboring neurons

Our nervous system

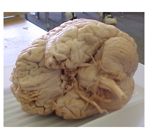


The human brain

- contains ~10^{11} (100 billion) neurons
- each neuron is connected to ~10^{4} (10,000) other neurons
- Neurons can fire as fast as 10^{-3} seconds

How does this compare to a computer?

Man vs. Machine

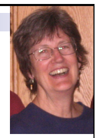


10^{11} neurons	10^{10} transistors
10^{11} neurons	10^{11} bits of ram/memory
10^{14} synapses	10^{13} bits on disk
10^{-3} "cycle" time	10^{-9} cycle time

Brains are still pretty fast

Who is this?

Brains are still pretty fast



If you were me, you'd be able to identify this person in 10^{-1} (1/10) s!

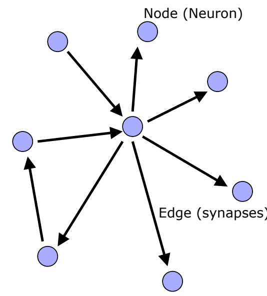
Given a neuron firing time of 10^{-3} s, how many neurons in sequence could fire in this time?

- A few hundred

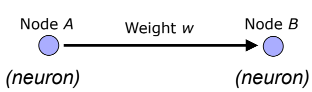
What are possible explanations?

- either neurons are performing some very complicated computations
- brain is taking advantage of the **massive** parallelization (remember, neurons are connected ~10,000 other neurons)

Artificial Neural Networks



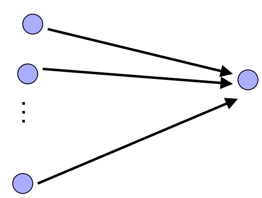
our approximation



W is the strength of signal sent between A and B.

If *A* fires and *w* is **positive**, then *A* **stimulates** *B*.

If *A fires* and *w* is **negative**, then *A* **inhibits** *B*.

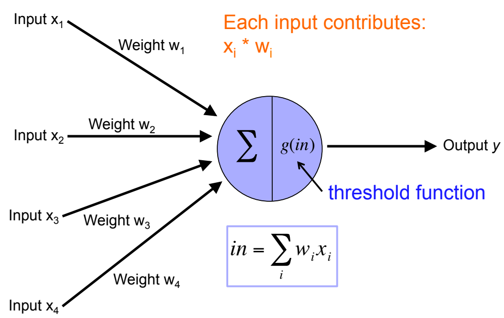


A given neuron has many, many connecting, input neurons

If a neuron is stimulated enough, then it also fires

How much stimulation is required is determined by its **threshold**

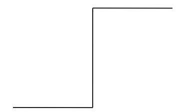
A Single Neuron/Perceptron



Possible threshold functions

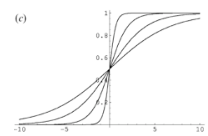
hard threshold

$$g(x) = \begin{cases} 1 & \text{if } x \geq threshold \\ 0 & otherwise \end{cases}$$

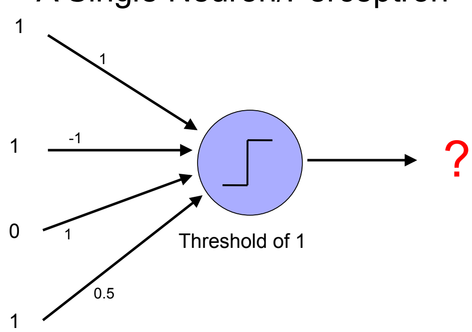


sigmoid

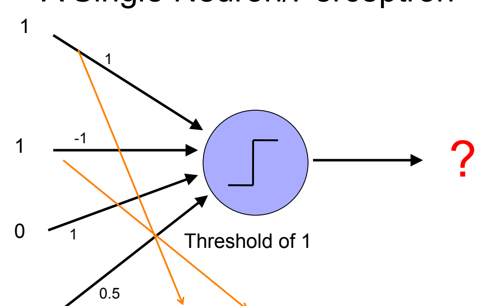
$$g(x) = \frac{1}{1 + e^{-ax}}$$



A Single Neuron/Perceptron

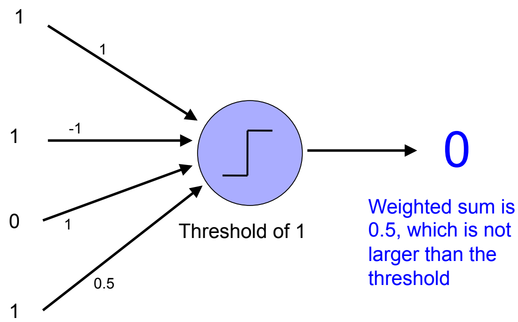


A Single Neuron/Perceptron

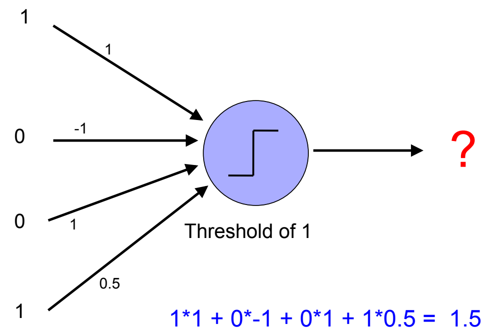


1*1 + 1*-1 + 0*1 + 1*0.5 = 0.5

A Single Neuron/Perceptron

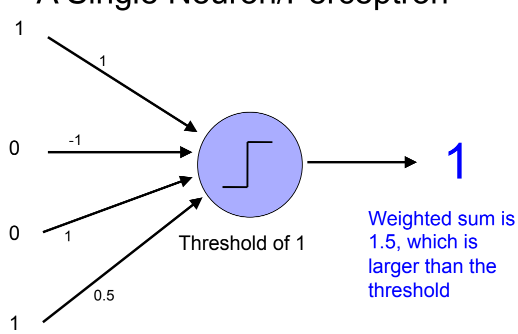


A Single Neuron/Perceptron

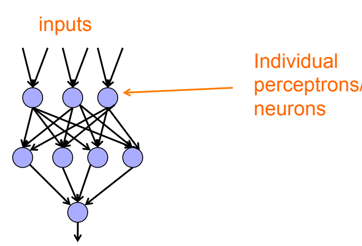


$1*1 + 0*-1 + 0*1 + 1*0.5 = 1.5$

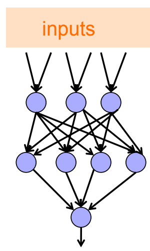
A Single Neuron/Perceptron



Neural network

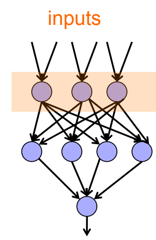


Neural network



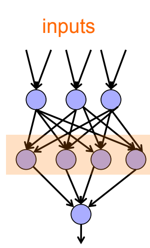
some inputs are provided/entered

Neural network



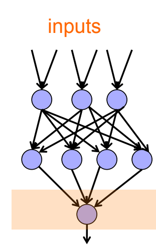
each perceptron computes and calculates an answer

Neural network



those answers become inputs for the next level

Neural network



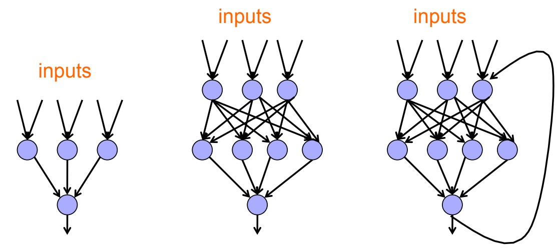
finally get the answer after all levels compute

Activation spread

http://www.youtube.com/watch?v=Yq7d4ROvZ6I

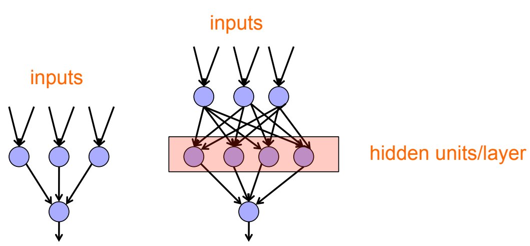
Neural networks

Different kinds/characteristics of networks



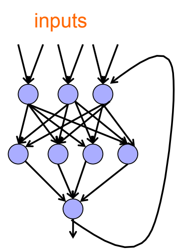
How are these different?

Neural networks



Feed forward networks

Neural networks



Recurrent network

Output is fed back to input

Can support memory!

How?

History of Neural Networks

McCulloch and Pitts (1943) – introduced model of artificial neurons and suggested they could learn

Hebb (1949) – Simple updating rule for learning

Rosenblatt (1962) - the *perceptron* model

Minsky and Papert (1969) – wrote *Perceptrons*

Bryson and Ho (1969, but largely ignored until 1980s--Rosenblatt) – invented back-propagation learning for multilayer networks

Training the perceptron

First wave in neural networks in the 1960's

Single neuron

Trainable: its threshold and input weights can be modified

If the neuron doesn't give the desired output, then it has made a mistake

Input weights and threshold can be changed according to a learning algorithm

Examples - Logical operators

AND – if all inputs are 1, return 1, otherwise return 0

OR – if at least one input is 1, return 1, otherwise return 0

NOT – return the opposite of the input

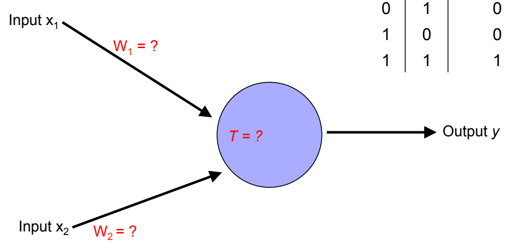
XOR – if exactly one input is 1, then return 1, otherwise return 0

AND

x_1	x_2	x_1 **and** x_2
0	0	0
0	1	0
1	0	0
1	1	1

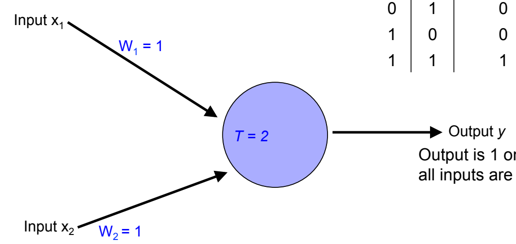
AND

x_1	x_2	x_1 **and** x_2
0	0	0
0	1	0
1	0	0
1	1	1



AND

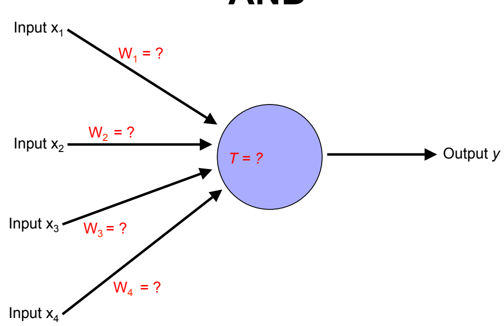
x_1	x_2	x_1 **and** x_2
0	0	0
0	1	0
1	0	0
1	1	1



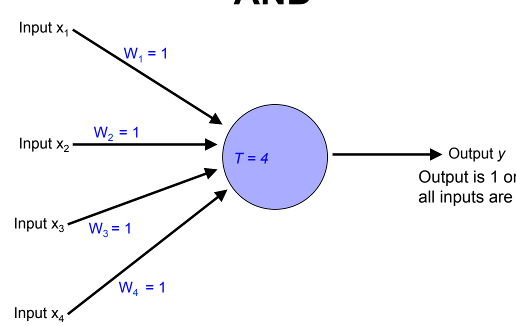
Output is 1 only if all inputs are 1

Inputs are either 0 or 1

AND



AND



Output is 1 only if all inputs are 1

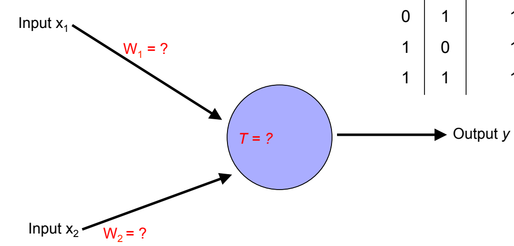
Inputs are either 0 or 1

OR

x_1	x_2	x_1 **or** x_2
0	0	0
0	1	1
1	0	1
1	1	1

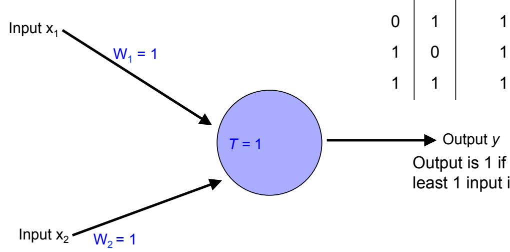
OR

x_1	x_2	x_1 **or** x_2
0	0	0
0	1	1
1	0	1
1	1	1



OR

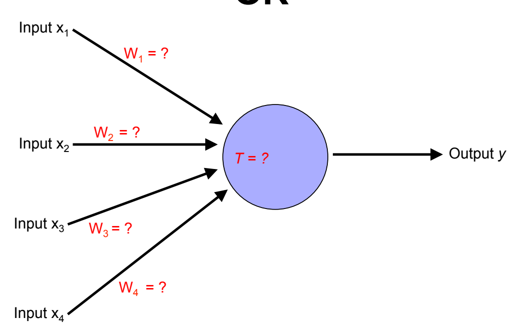
x_1	x_2	x_1 **or** x_2
0	0	0
0	1	1
1	0	1
1	1	1



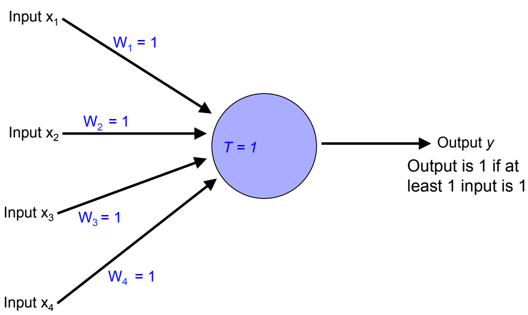
Output is 1 if at least 1 input is 1

Inputs are either 0 or 1

OR



OR



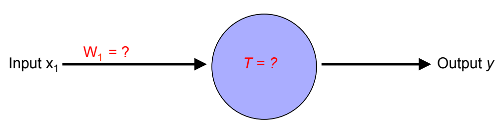
Inputs are either 0 or 1

NOT

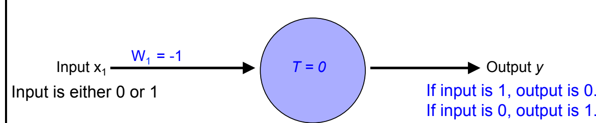
x_1	**not** x_1
0	1
1	0

NOT

x_1	**not** x_1
0	1
1	0

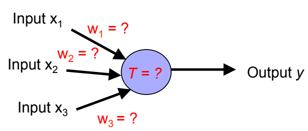


NOT

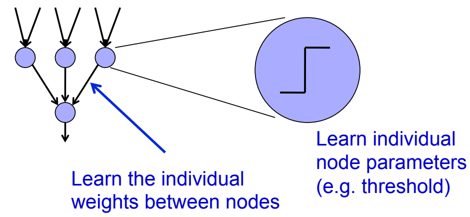


How about…

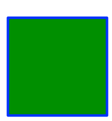
x_1	x_2	x_3	x_1 **and** x_2
0	0	0	1
0	1	0	0
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	1
1	0	1	1
1	1	1	0



Training neural networks

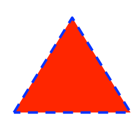


Positive or negative?



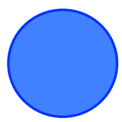
NEGATIVE

Positive or negative?



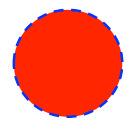
NEGATIVE

Positive or negative?



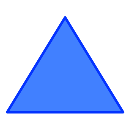
POSITIVE

Positive or negative?



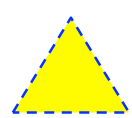
NEGATIVE

Positive or negative?



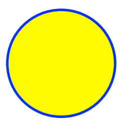
POSITIVE

Positive or negative?



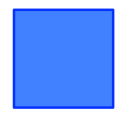
POSITIVE

Positive or negative?



NEGATIVE

Positive or negative?



POSITIVE

A method to the madness

blue = positive

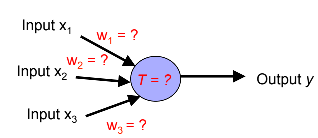
yellow triangles = positive

all others negative

How did you figure this out (or some of it)?

Training neural networks

x_1	x_2	x_3	x_1 **and** x_2
0	0	0	1
0	1	0	0
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	1
1	0	1	1
1	1	1	0



1. start with some initial weights and thresholds
2. show examples repeatedly to NN
3. update weights/thresholds by comparing NN output to actual output

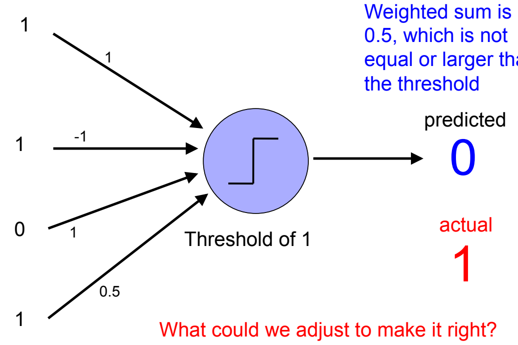
Perceptron learning algorithm

repeat until you get all examples right:

- for each “training” example:
  - calculate current prediction on example
  - if *wrong*:
    - update weights and threshold towards getting this example correct

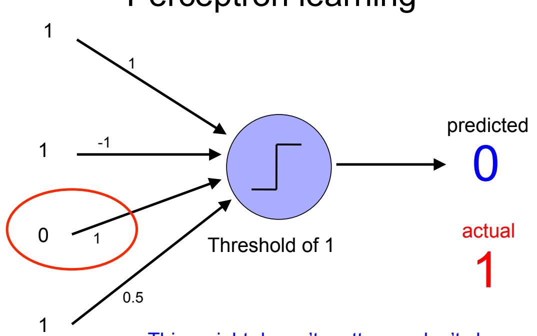
Perceptron learning

Weighted sum is 0.5, which is not equal or larger than the threshold



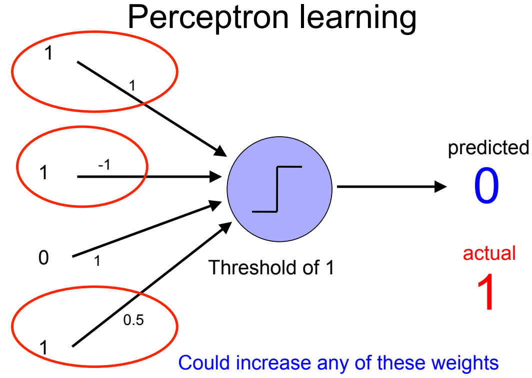
What could we adjust to make it right?

Perceptron learning



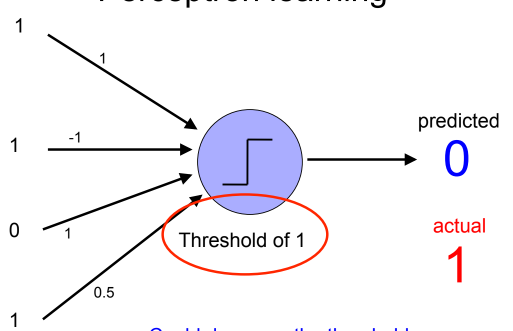
This weight doesn’t matter, so don’t change

Perceptron learning



Could increase any of these weights

Perceptron learning



Could decrease the threshold

Perceptron learning

A few missing details, but not much more than this

Keeps adjusting weights as long as it makes mistakes

If the training data is linearly separable the perceptron learning algorithm is guaranteed to converge to the "correct" solution (where it gets all examples right)

Linearly Separable

x_1	x_2	x_1 and x_2
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	x_1 or x_2
0	0	0
0	1	1
1	0	1
1	1	1

x_1	x_2	x_1 xor x_2
0	0	0
0	1	1
1	0	1
1	1	0

A data set is linearly separable if you can separate one example type from the other

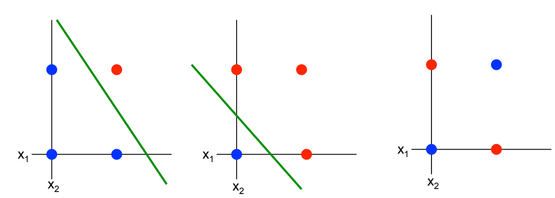
Which of these are linearly separable?

Which of these are linearly separable?

x_1	x_2	x_1 and x_2
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	x_1 or x_2
0	0	0
0	1	1
1	0	1
1	1	1

x_1	x_2	x_1 xor x_2
0	0	0
0	1	1
1	0	1
1	1	0



Perceptrons

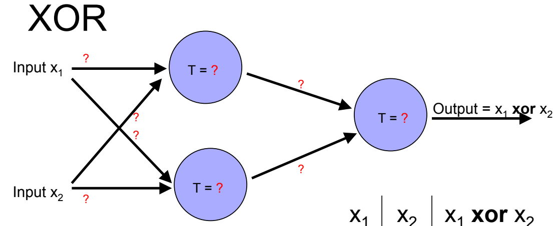
1969 book by Marvin Minsky and Seymour Papert

The problem is that they can only work for classification problems that are linearly separable

Insufficiently expressive

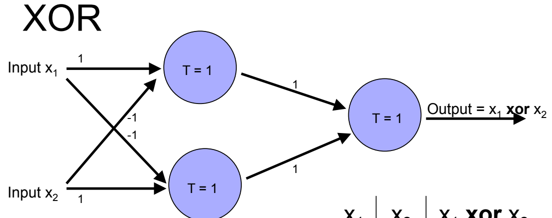
"Important research problem" to investigate multilayer networks although they were pessimistic about their value

XOR



x_1	x_2	x_1 **xor** x_2
0	0	0
0	1	1
1	0	1
1	1	0

XOR



x_1	x_2	x_1 **xor** x_2
0	0	0
0	1	1
1	0	1
1	1	0