Neural Networks and Deep Learning

www.cs.wisc.edu/~dpage/cs760/

Goals for the lecture

you should understand the following concepts

- perceptrons
- the perceptron training rule
- linear separability
- hidden units
- multilayer neural networks
- gradient descent
- stochastic (online) gradient descent
- sigmoid function
- gradient descent with a linear output unit
- gradient descent with a sigmoid output unit
- backpropagation

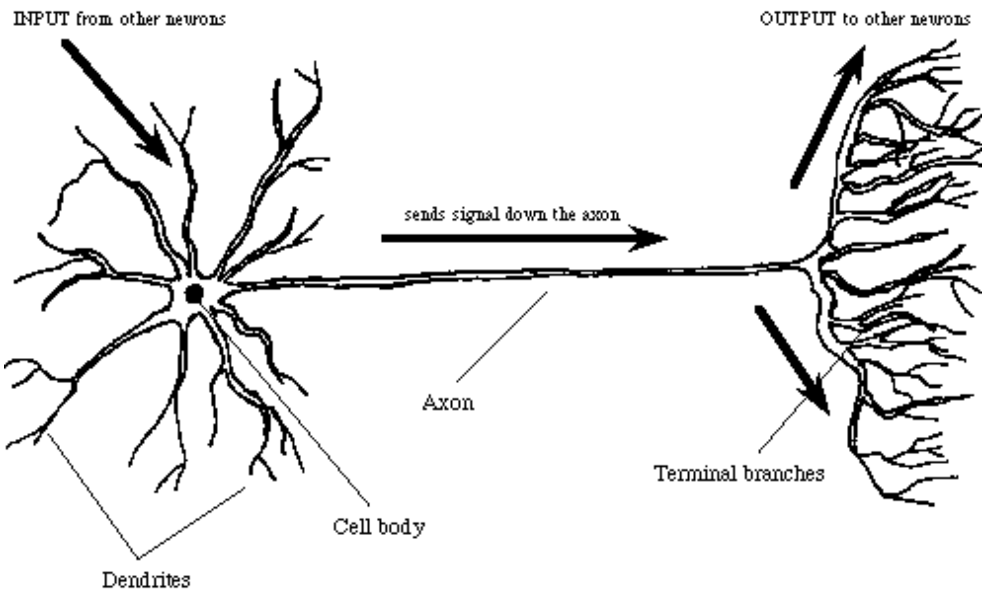
Goals for the lecture

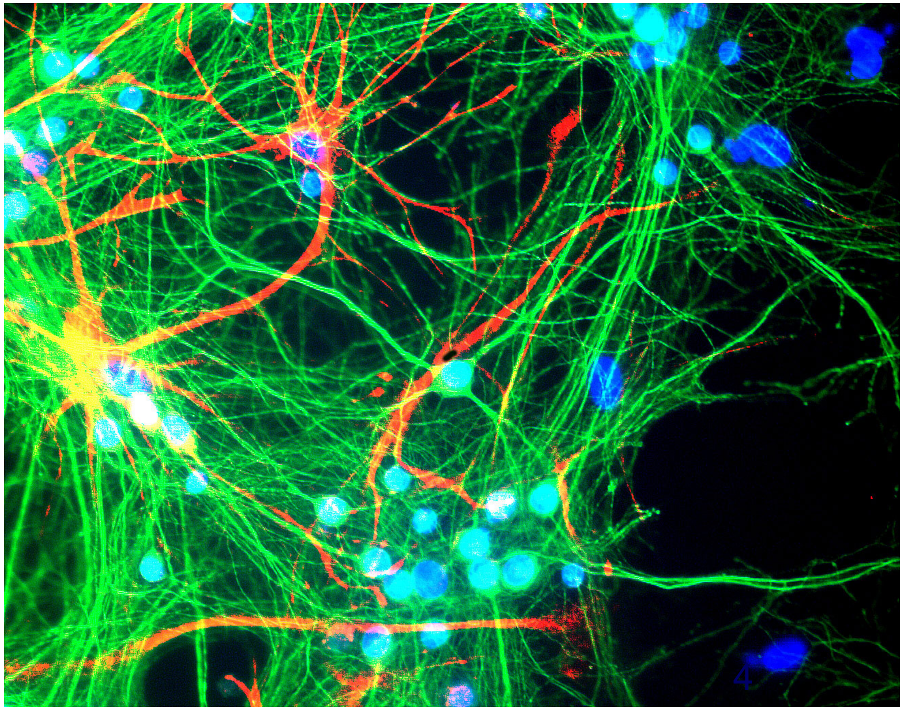
you should understand the following concepts

- weight initialization
- early stopping
- the role of hidden units
- input encodings for neural networks
- output encodings
- recurrent neural networks
- autoencoders
- stacked autoencoders

Neural networks

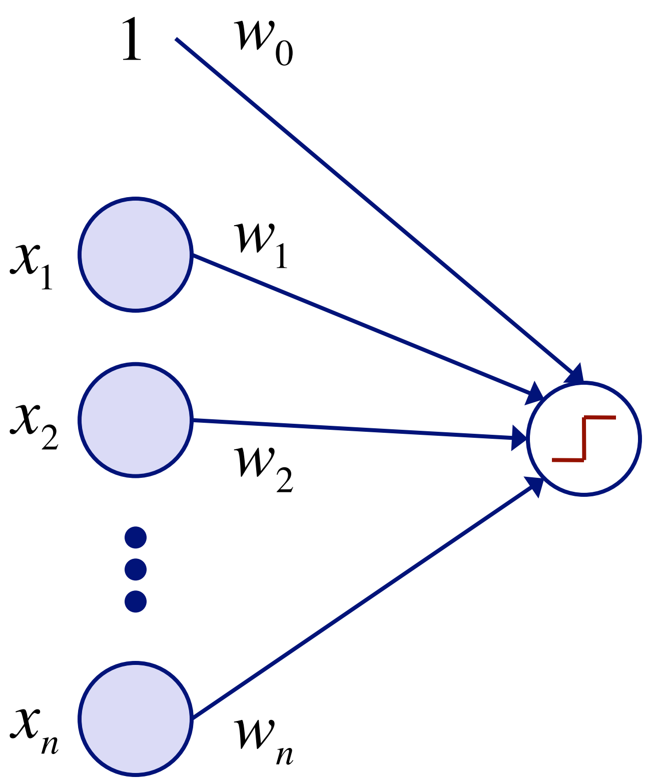
- a.k.a. *artificial neural networks*, *connectionist models*
- inspired by interconnected neurons in biological systems
    - simple processing units
    - each unit receives a number of real-valued inputs
    - each unit produces a single real-valued output





Perceptrons

[McCulloch & Pitts, 1943; Rosenblatt, 1959; Widrow & Hoff, 1960]



$$o = \begin{cases} 1 & \text{if } \; w_0 + \sum_{i=1}^{n} w_i x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

input units:
represent given $\boldsymbol{x}$

output unit:
represents binary classification

Learning a perceptron: the perceptron training rule

1. randomly initialize weights

2. iterate through training instances until convergence

   2a. calculate the output for the given instance

$$o = \begin{cases} 1 & \text{if } w_0 + \sum_{i=1}^{n} w_i x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

   2b. update each weight

$$\Delta w_i = \eta \left(y - o \right) x_i$$

$$w_i \leftarrow w_i + \Delta w_i$$

η is *learning rate*;
set to value << 1

Representational power of perceptrons

perceptrons can represent only *linearly separable* concepts

$$o = \begin{cases} 1 \;\; \text{if} \;\; w_0 + \sum_{i=1}^{n} w_i x_i > 0 \\ 0 \;\; \text{otherwise} \end{cases}$$

w

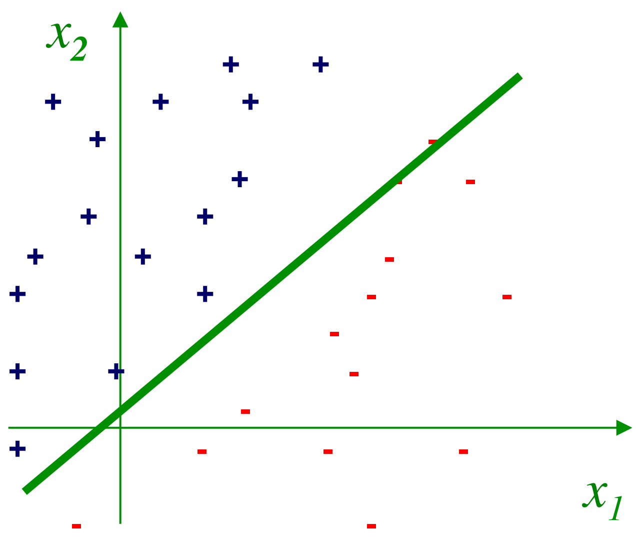
decision boundary given by:

$$1 \;\; \text{if} \;\; w_0 + w_1 x_1 + w_2 x_2 > 0$$

also write as: **wx** > 0

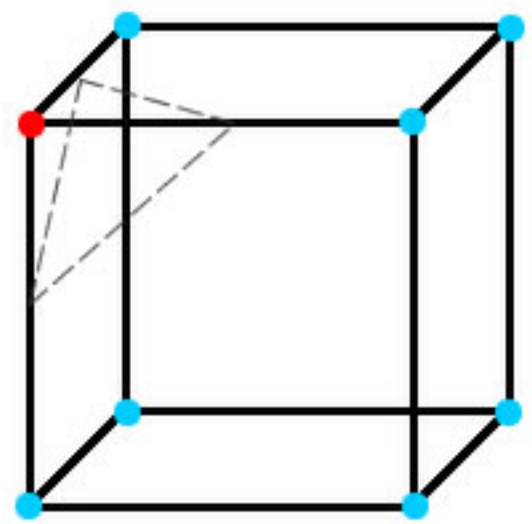
$$w_1 x_1 + w_2 x_2 = -w_0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$



Representational power of perceptrons

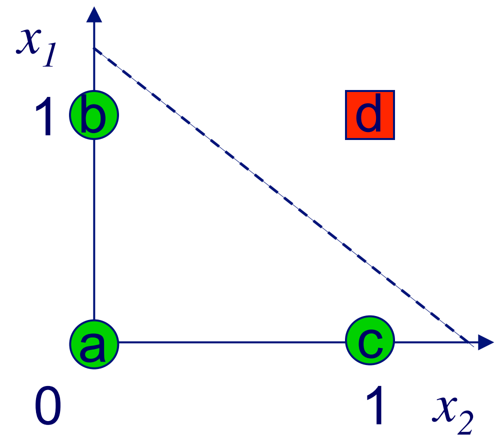
- in previous example, feature space was 2D so decision boundary was a line
- in higher dimensions, decision boundary is a hyperplane



Some linearly separable functions

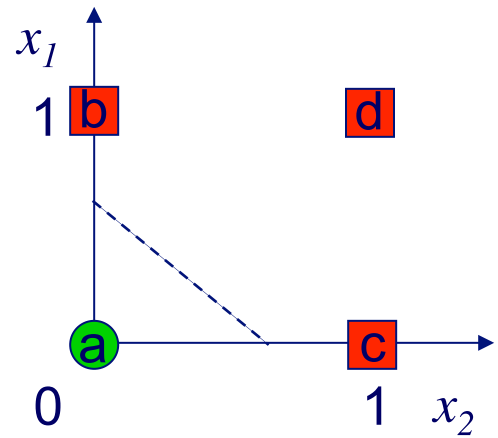
AND

	x_1	x_2	y
a	0	0	0
b	0	1	0
c	1	0	0
d	1	1	1



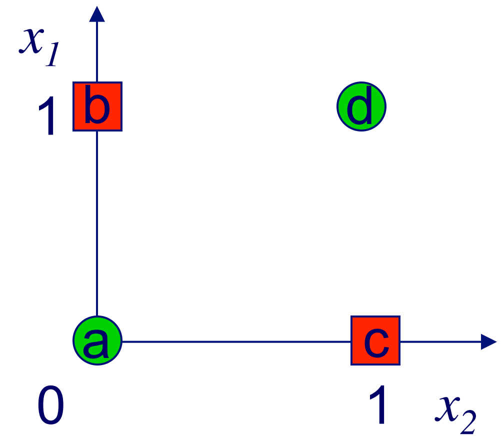
OR

	x_1	x_2	y
a	0	0	0
b	0	1	1
c	1	0	1
d	1	1	1

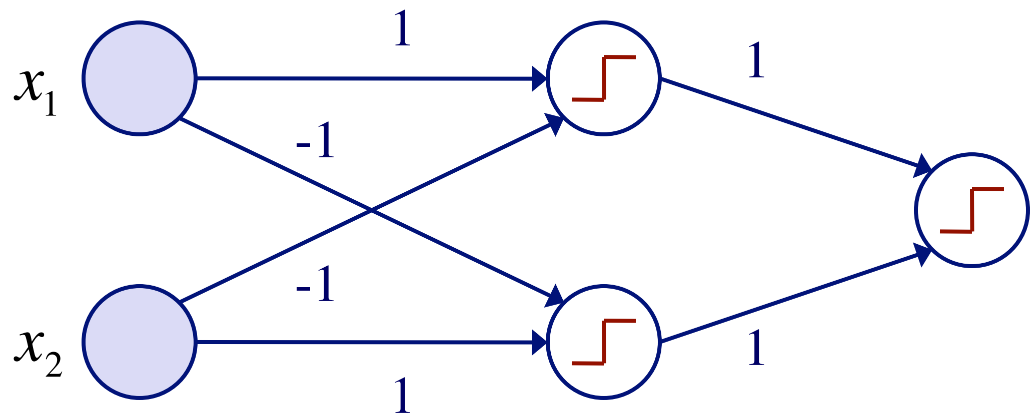


XOR is not linearly separable

	x_1	x_2	y
a	0	0	0
b	0	1	1
c	1	0	1
d	1	1	0



a multilayer perceptron can represent XOR



assume $w_0 = 0$ for all nodes