Experimental Design for Machine Learning on Multimedia Data
Lecture 3

Dr. Gerald Friedland,
fractor@eecs.berkeley.edu

Website: http://www.icsi.berkeley.edu/~fractor/fall2020/

Project Specs and HW coming this week

- Homework 1: Will be online (delayed due to Labor Day)
- Project specs will be online on website
- Please start forming teams as of next week.
- Check Piazza posts.

Reminder:

Office hours: Monday 1-2pm, this Zoom

Refresh: Memory Arithmetic

- *Information is reduction of uncertainty*:
  $H=-\log_2 P= -\log_2 \frac{1}{\#states} = \log_2 \#states$
  measured in bits.

- Information: $\log_2 \#states$ (positive bits)
  Uncertainty: $\log_2 P=\log_2 \frac{1}{\#states}$ (negative bits)

- If states are not equiprobable, *Shannon Entropy* provides tighter bound.

  Important for homework!

Refresh: Thought Framework: Machine Learning

Assume

$$x_i \in \mathbb{R}, f(\vec{x}) \in \{0,1\}$$

(binary classifier)

Question:

How many state transitions does *M* need to model the training data?

Maximally: #rows (lookup table)
Minimally: ?

Refresh: Thought Framework: Machine Learning

- ***Intellectual Capacity:*** *The number of unique target functions a machine learner is able to represent (as a function of the number of model parameters).*

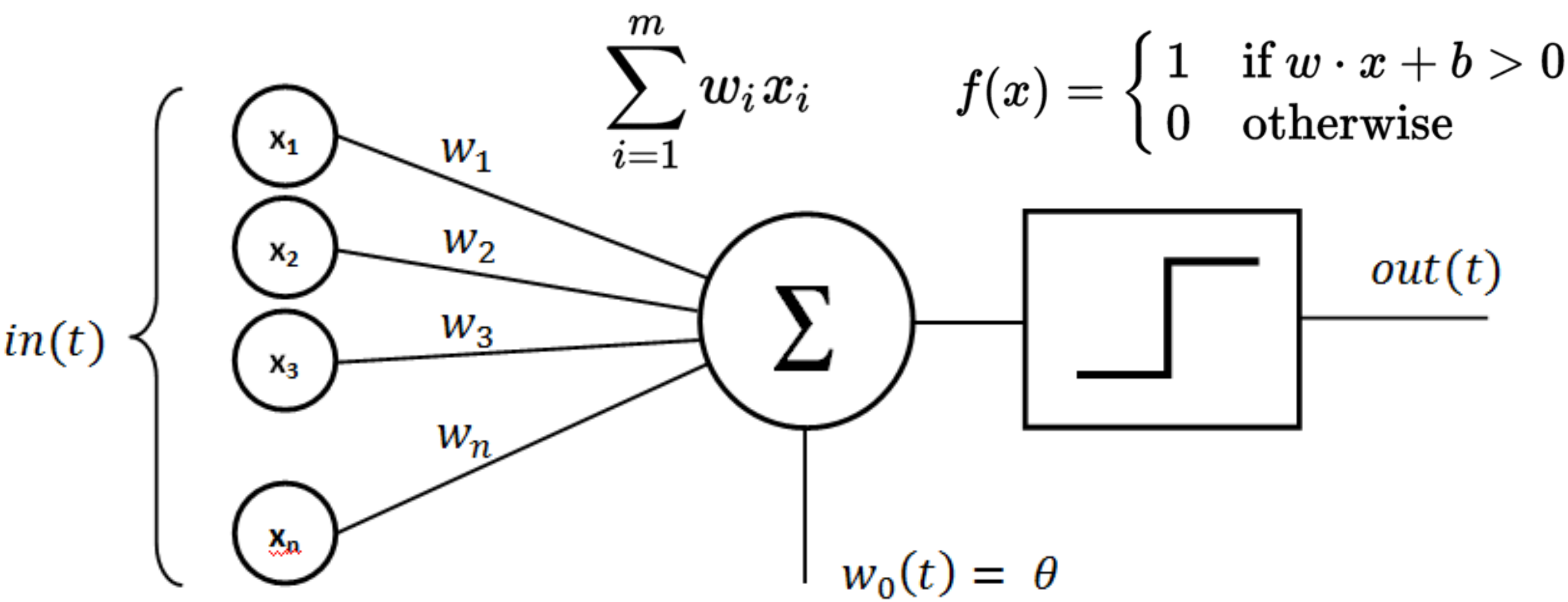
- ***Memory Equivalent Capacity (MEC):*** *A machine learner's intellectual capacity is memory-equivalent to N bits when the machine learner is able to represent all 2^N binary labeling functions of N inputs.*

- At MEC or higher, *M* is able to **memorize** all possible state transitions from the input to the output.

Recap: Machine Learning as Engineering Discipline

- Supervised **Machine Learners have a memory capacity in bits** that is **computable** and **measurable**.
  - Artificial Neural Networks with gating functions (Sigmoid, ReLU, etc.) have
    - a capacity upper limit that can be determined analytically using 4 principles
    - an effective capacity that can be measured on actual implementations.
- Predicting and measuring capacity allows for task-independent optimization of a concrete network architecture, learning algorithm, convergence tricks, etc…
- Capacity requirement can be approximately predicted given the input data and ground truth.
- Generalization can fail as a result of input redundancies. Occam's Razor helps to minimize the risk.

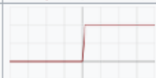
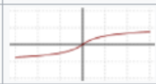
The Perceptron (Rosenblatt, Widrow)



Physical interpretation: Energy threshold

Source: Wikipedia

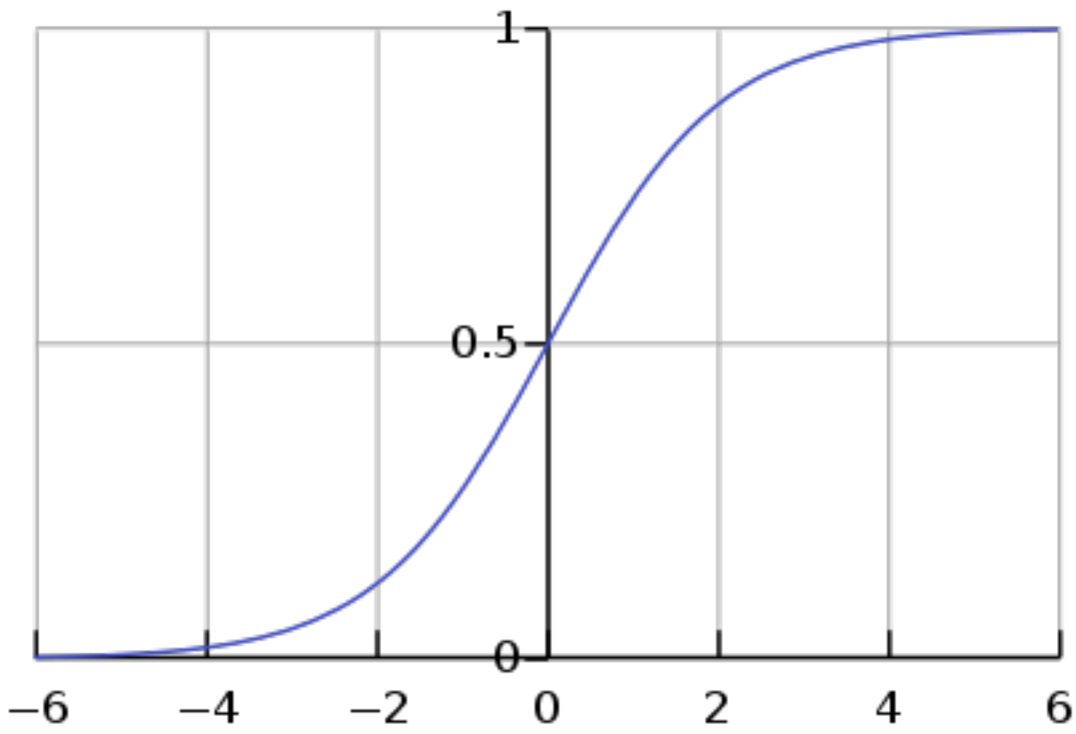
Activation Functions (too many)

Identity		$f(x)=x$	$f'(x)=1$	$(-\infty,\infty)$	C^∞
Binary step		$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x\ne 0\\ ? & \text{for } x=0\end{cases}$	$\{0,1\}$	C^{-1}
Logistic (a.k.a. Soft step)		$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$	$(0,1)$	C^∞
TanH		$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$	$(-1,1)$	C^∞
ArcTan		$f(x)=\tan^{-1}(x)$	$f'(x)=\frac{1}{x^2+1}$	$\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$	C^∞
Softsign [7][8]		$f(x)=\frac{x}{1+\|x\|}$	$f'(x)=\frac{1}{(1+\|x\|)^2}$	$(-1,1)$	C^1
Rectified linear unit (ReLU)[9]		$f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$[0,\infty)$	C^0
Leaky rectified linear unit (Leaky ReLU)[10]		$f(x)=\begin{cases}0.01x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(x)=\begin{cases}0.01 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$(-\infty,\infty)$	C^0
Parameteric rectified linear unit (PReLU)[11]		$f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(\alpha,x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$(-\infty,\infty)$	C^0
Randomized leaky rectified linear unit (RReLU)[12]		$f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ [1]	$f'(\alpha,x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$(-\infty,\infty)$	C^0
Exponential linear unit (ELU)[13]		$f(\alpha,x)=\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$	$f'(\alpha,x)=\begin{cases}f(\alpha,x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$	$(-\alpha,\infty)$	C^1 when $\alpha=1$, otherwise C^0

Activation functions approximate the sharp decision boundary.

Source: Wikipedia

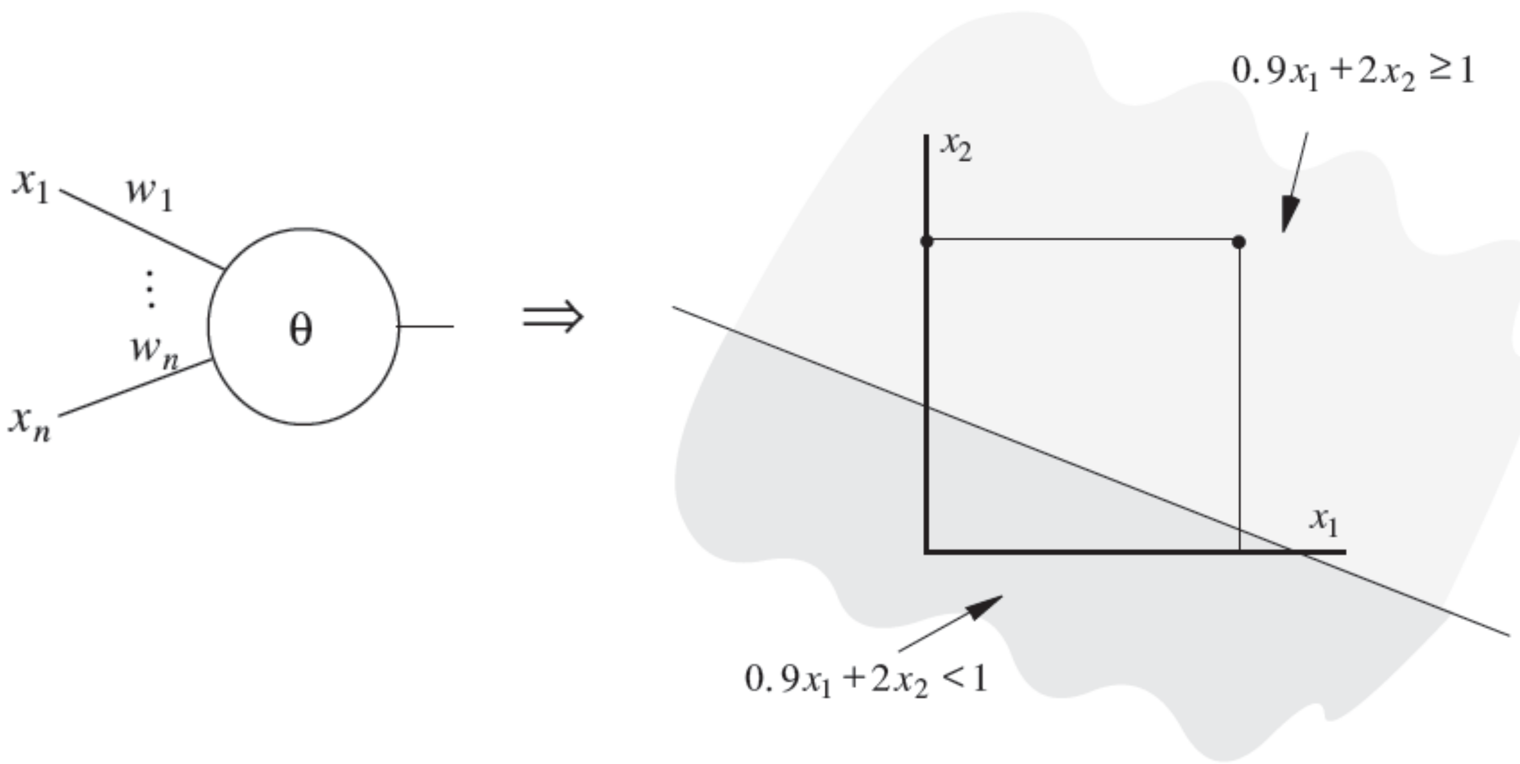
Activation Functions Make no Difference for Information Content!



What is the probability that the decision is 0?
What is the probability that the decision is 1?

Source: Wikipedia

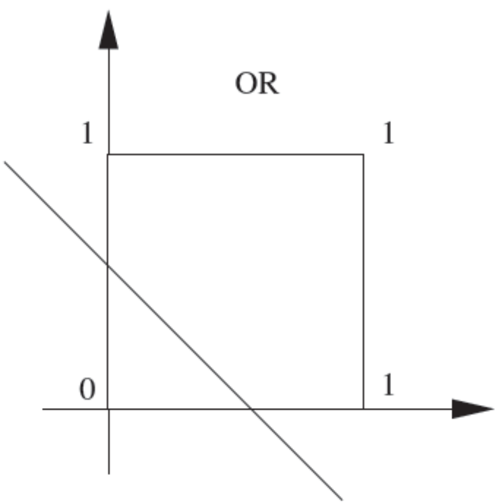
How many functions can be modeled using a Perceptron?

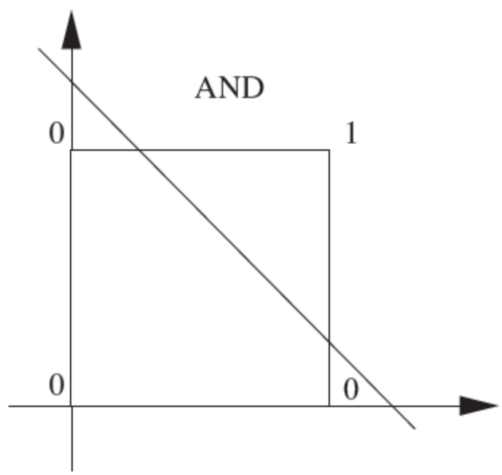


Source: R. Rojas, Intro to Neural Networks

Minsky's Problem

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1





- 2^{2^v} functions of *v* boolean variables
- 2^v labelings of 2^v points.
- For *v=2*, all but *2* functions work: XOR, NXOR

Source: R. Rojas, Intro to Neural Networks

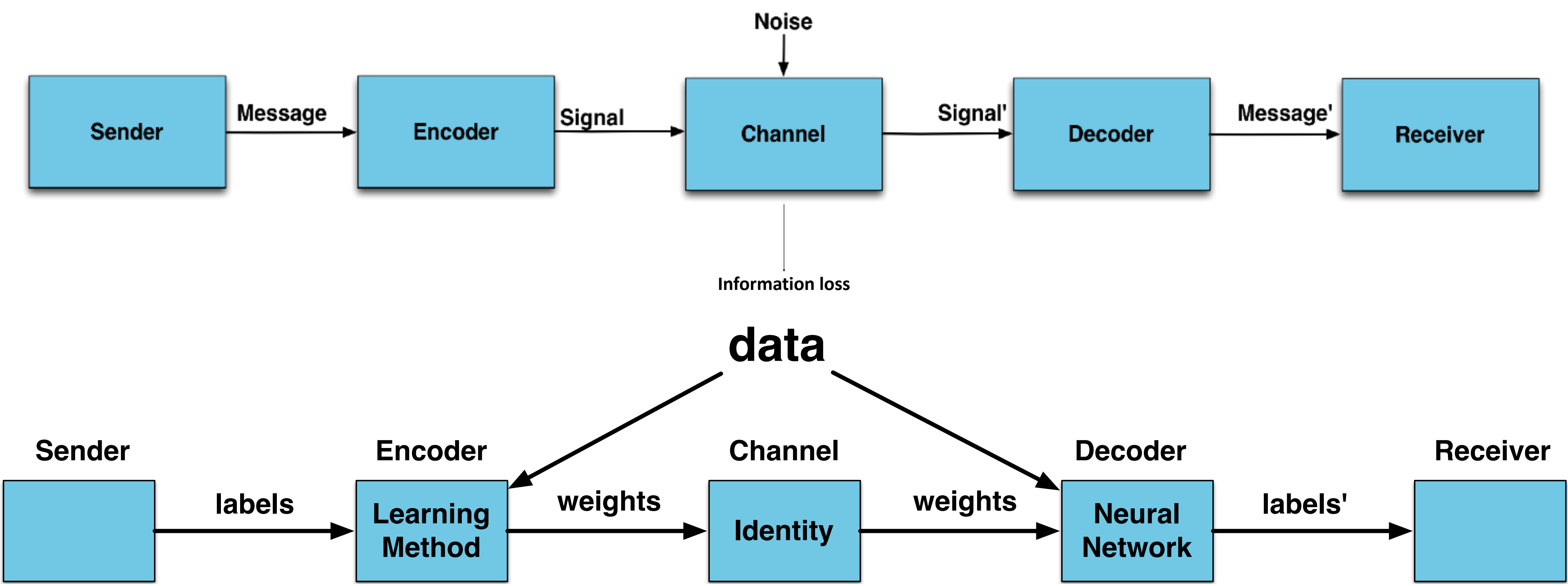
Orthogonal Concept: Vapnik-Chervonenkis Dimension

Definition 3.1 (VC Dimension [47]). The VC dimension D_{VC} of a hypothesis space f is the maximum integer $D = D_{VC}$ such that *some dataset* of cardinality D can be shattered by f. Shattered by f means that any arbitrary labeling can be represented by a hypothesis in f. If there is no maximum, it holds $D_{VC} = \infty$.

Some $\neq$ Any!

Memory Equivalent Capacity = Lossless Memory Dimension D_{LM}
= *any* dataset of Cardinality D.

MacKay's Solution: Machine Learning as an Encoder/Decoder



Main trick: Let the Machine Learner 'learn' uniform random points!

Source: D. MacKay: Information Theory, Inference and Learning

How many points can a Perceptron label in general?

Formula by Schlaefli (1852):

$$T(n,k) = T(n-1,k) + T(n-1,k-1), \qquad (3)$$

where $T(n,1) = T(1,k) = 2$ or iteratively:

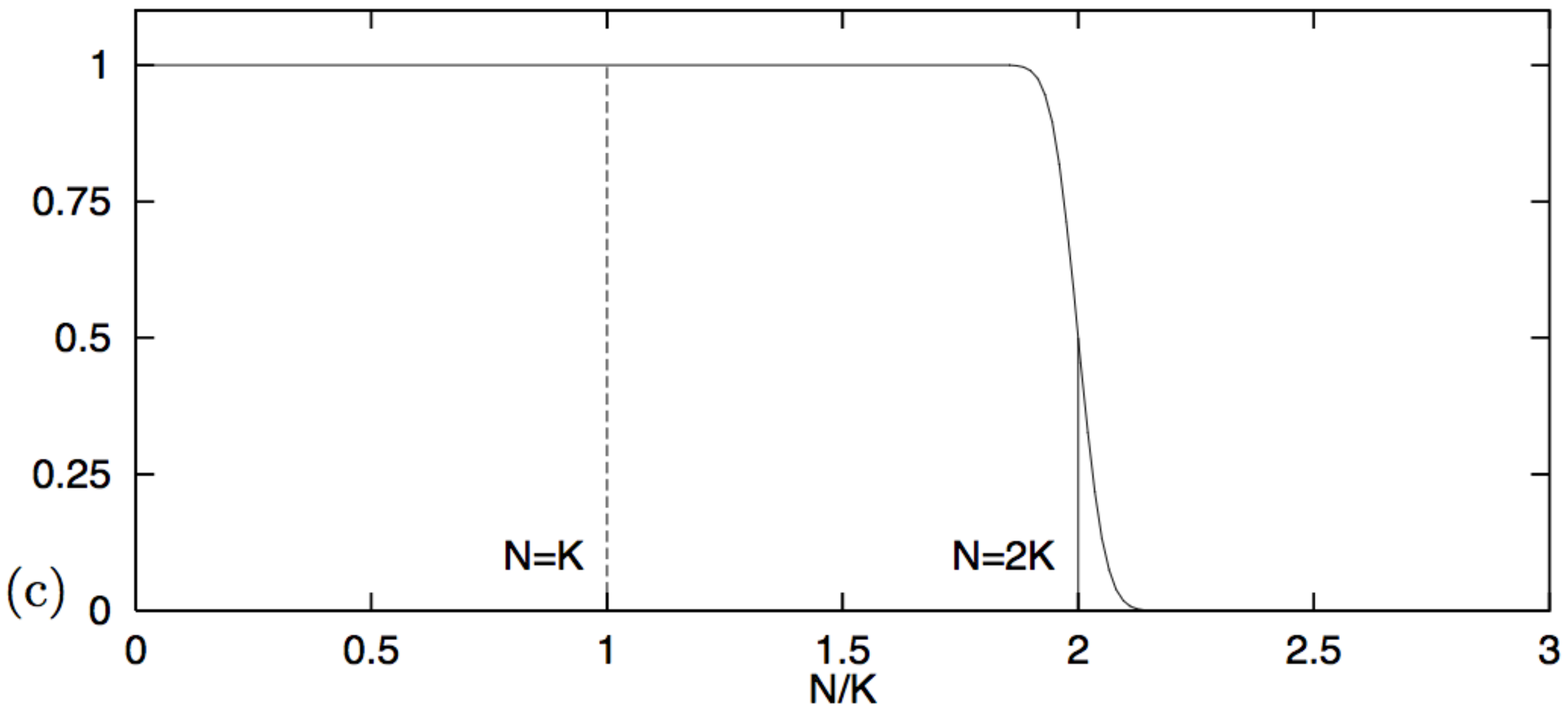
$$T(n,k) = 2\sum_{l=0}^{k-1} \binom{n-1}{l} \qquad (4)$$

$n \backslash k$	1	2	3	4	5	6	7	8
1	**2**	2	2	2	2	2	2	2
2	2	**4**	4	4	4	4	4	4
3	2	6	**8**	8	8	8	8	8
4	2	8	14	**16**	16	16	16	16
5	2	10	22	30	**32**	32	32	32
6	2	12	32	52	62	**64**	64	64
7	2	14	44	84	114	126	**128**	128
8	2	16	58	128	198	240	254	**256**

Table 1: Some values of the $T(n,k)$ function indicating the number of distinct threshold functions on n points in general position in k dimensions as defined by [22].

$$T(n,k) = 2^n \text{ for } k >= n.$$

Critical Points: Perceptron (Cover, MacKay)



N=K: VC Dimension (for points in random position)

N=2K: Cover/MacKay Capacity

Source: D. MacKay: Information Theory, Inference and Learning

Next Lecture: Generalizing from Perceptron to Neural Networks

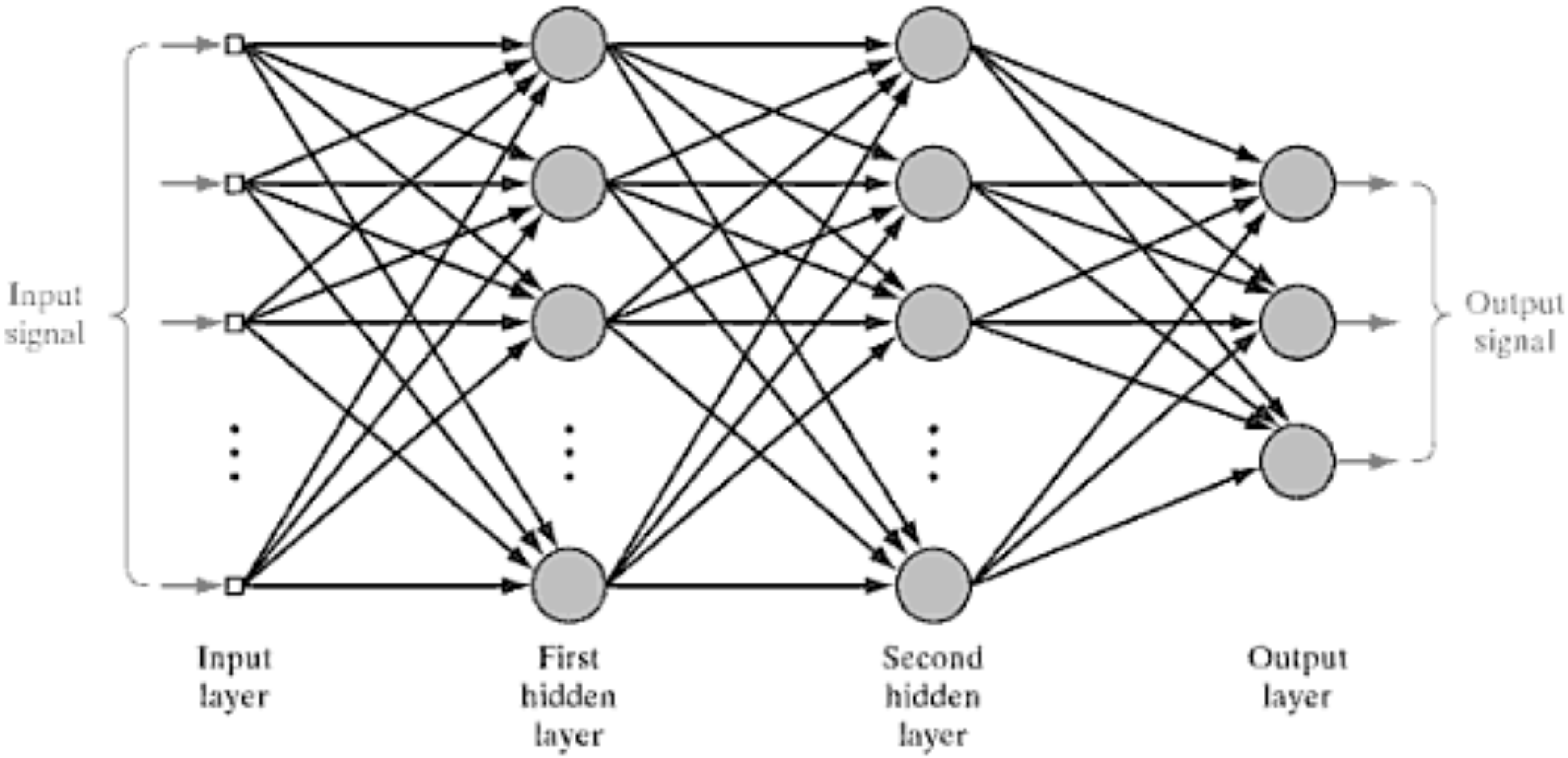
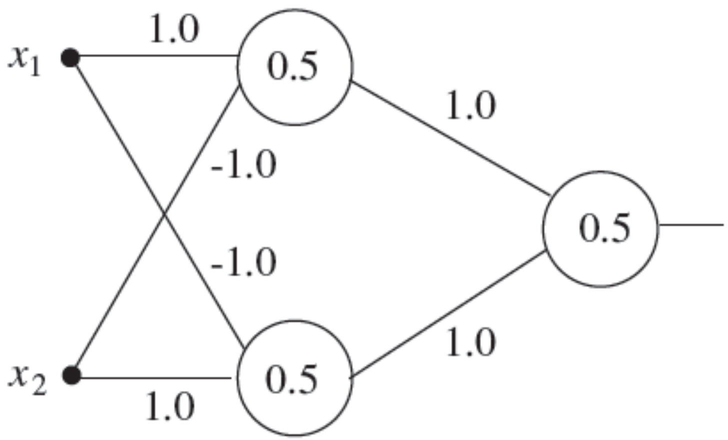


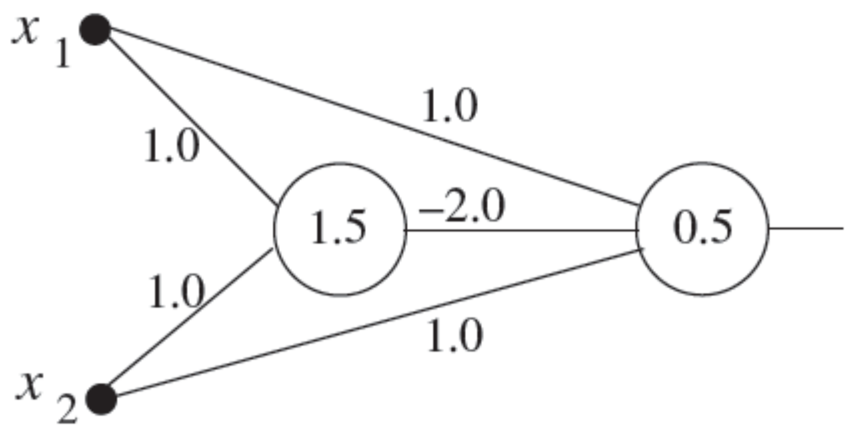
FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

Source: Wikipedia

Careful: Other Architectures



Typical MLP



Shortcut Network

Example Solutions to XOR

Source: R. Rojas, Intro to Neural Networks

Next Lecture

- Capacity for Neural Networks explained: See also cheat sheet.
- Practical applications
- Demo