

Experimental Design for Machine Learning

Gerald Friedland (UC Berkeley)

Paper, Demo, etc:

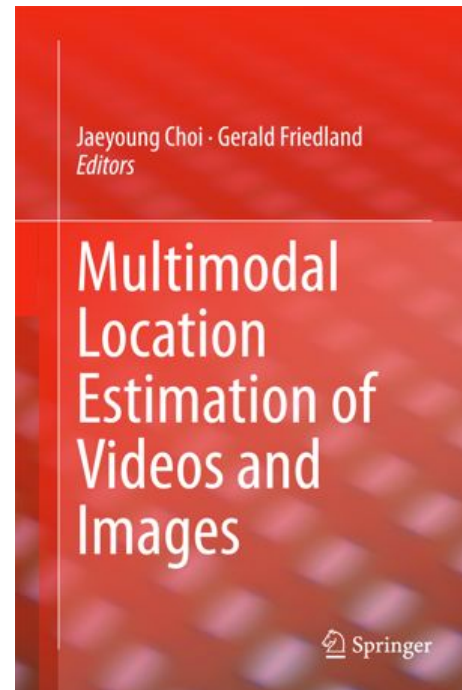
<https://tfmeter.icsi.berkeley.edu>

Commercial tool:

<http://brainome.ai>

About me....

- Adjunct Faculty, UC Berkeley
- Data Scientist at National Lab
- Started work in Machine Learning in 2001



Start of this work: Simple Question

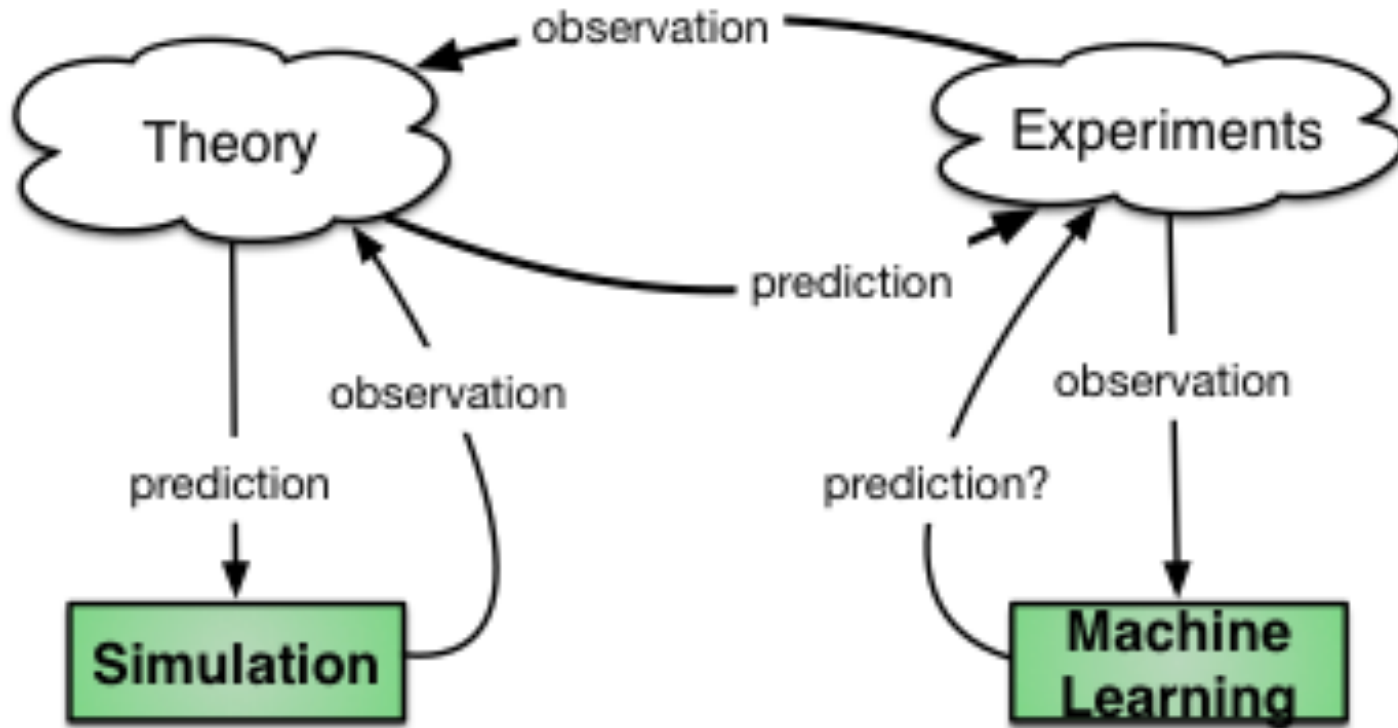
- **How much** money (cpu time, memory, IO) do I need to budget for my deep learning experiment?
- State of the Art: No answer.
For example, ImageNet models vary significantly:
 - AlexNet: 238MB model, 2.27Bn Ops
 - DarkNet: 28MB model, 0.96Bn Ops
 - VGG-16: 528 MB, 30.94Bn Ops

Source: <https://pjreddie.com/darknet/imagenet/>

A game...

- Continue the sequence:
 - 2, 4, 6, 8,
 - 6, 5, 1, 4,
- What is the next number?
 - 100000 (sequence 1)
 - 100000 (sequence 2)
- Why?

The Scientific Method



Data Science: The Science of Automating the Scientific Method

The Scientific Method: Practical (traditional)



Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data

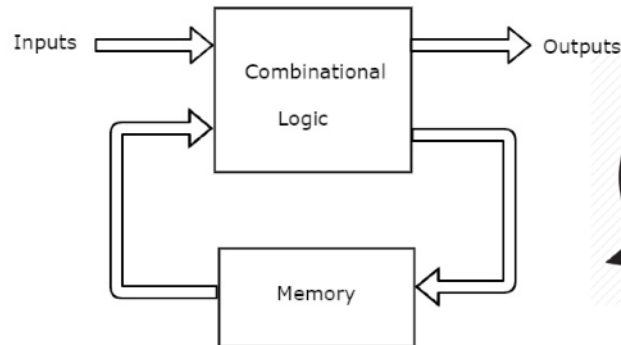


$$E = mc^2$$

The Scientific Method: Practical (new)



Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data



$$E = mc^2$$

Thought Framework: Machine Learning

- Intelligence: *The ability to adapt* (Binet and Simon, 1904)
- Machine learning *adapts a finite state machine M to an unknown function based on observations.*

- Input: n rows of observations (instances) in a table with header:

$$(x_1, x_2, \dots, x_m, f(\vec{x}))$$

where $f(\vec{x})$ is a column with labels we call target function.

- Output: State machine M that maps a point

$$(x_1, x_2, \dots, x_m) \implies f(\vec{x})$$

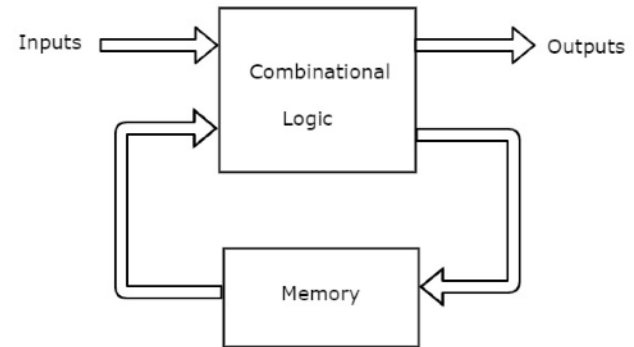
Thought Framework: Machine Learning

Assume

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

(binary classifier)

Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data



Question:

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

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.
Math: Assumptions needed! (infinity, distribution)
Engineering: Estimate using binning

Thought Framework: Machine Learning

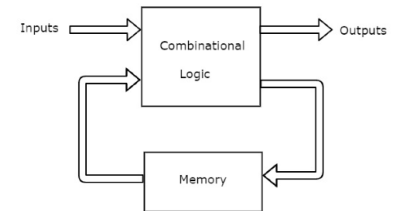
Assume

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

(binary classifier)

Question:

Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data



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

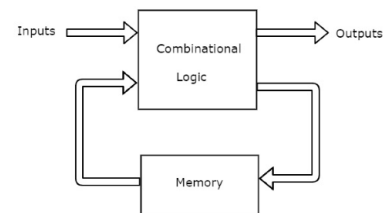
Maximally: #rows (lookup table)

Minimally: ? (Kolmogorov Complexity)

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

Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data



This Talk: Main trick

Memorization is worst-case generalization

- If we deduce nothing from data, the only thing we can do is memorize the observations verbatim.
- Using as many parameters as needed for memorization is therefore an indicator that the machine learner did not deduce anything (overfitting).
- Reducing parameters below memorization capacity will, in the best case, make the machine learner forget what's not relevant with regards to the target function: **generalization**.

Generalization in Machine Learning

Memorization is worst-case generalization.

For binary classifiers:

$$G = \frac{\text{\#correctly classified instances}}{\text{Memory Equivalent Capacity}} \left[\frac{\text{bits}}{\text{bit}} \right]$$

$G < 1 \Rightarrow M$ needs more training/data (not even memorizing)

$G = 1 \Rightarrow M$ is memorizing = overfitting

$1 < G < G_{MEM} \Rightarrow M$ could be implementing a lossless compression
(and still overfit)

$G > G_{MEM} \Rightarrow M$ is generalizing (no chance for overfitting)

Generalization in Machine Learning

$$G = \frac{\text{\#correctly classified instances}}{\text{Memory Equivalent Capacity}} \left[\frac{\text{bits}}{\text{bit}} \right]$$

Advantages of this definition:

- Keep current approach with training/validation/benchmark sets.
- No i.i.d. requirement for train/test set: Only requirement is input points are distinct!
- No distributional assumptions.

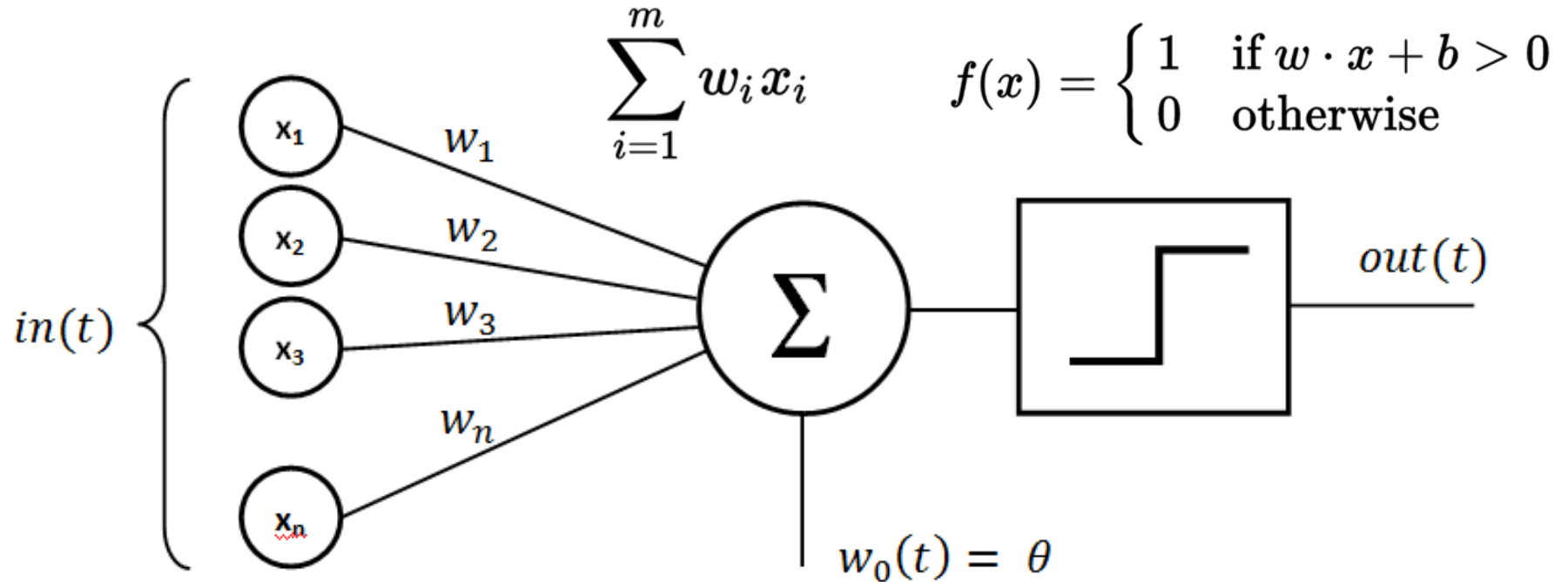
How do we calculate the Memory Equivalent Capacity?

- Binary Decision Tree: Depth of tree (if perfect).
- Neural Network (reminder of talk)
- Random Forrest: TBD
- SVN: TBD
- k-NN: TBD
- GMMs: TBD

Machine Learning as Engineering Discipline

- Supervised **Machine Learners have a Memory Equivalent 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.












Repeat: The Perceptron



Physical interpretation: Energy threshold

Source: Wikipedia

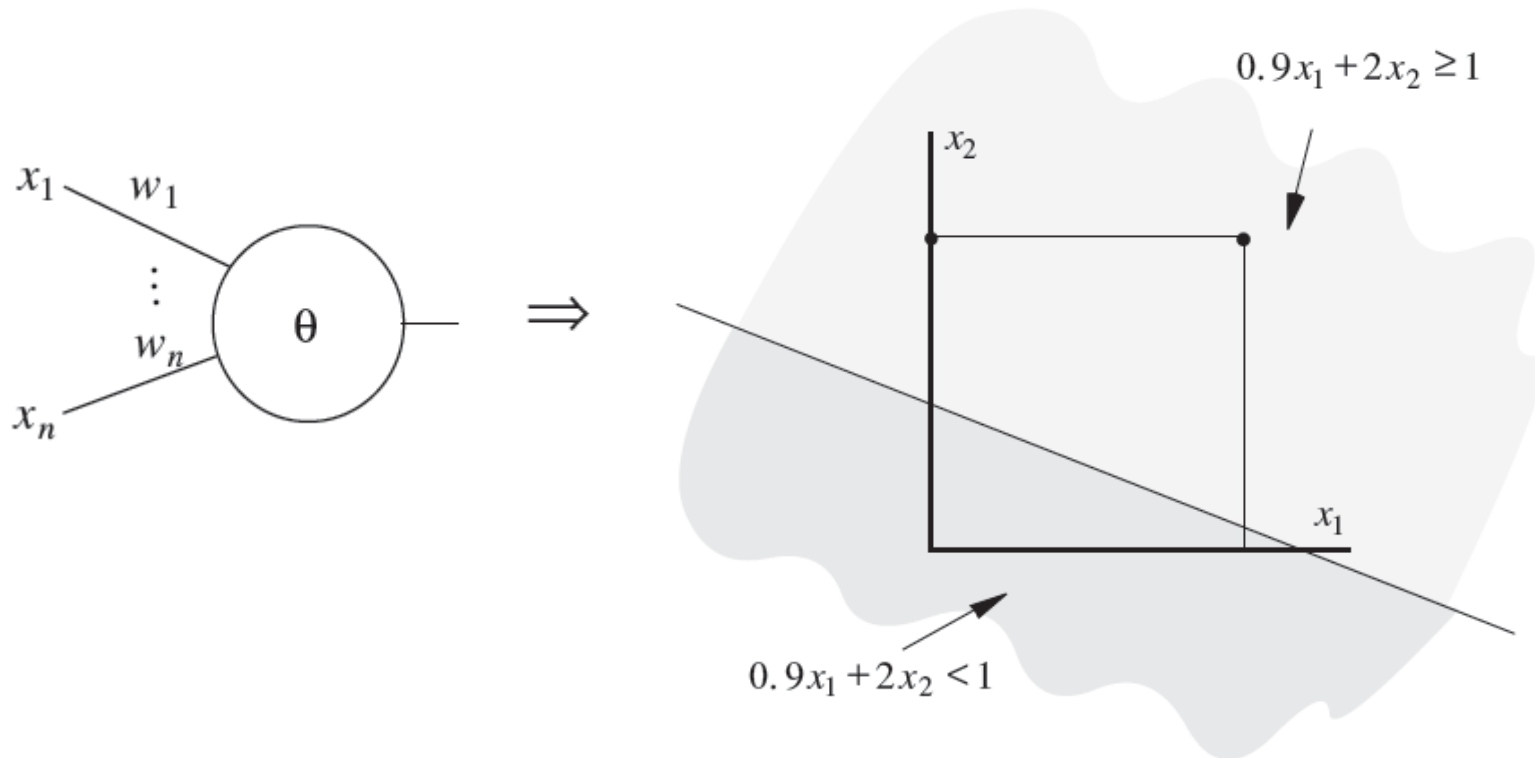
Repeat: Activation Functions (too many)

Identity		$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$	C^∞
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$	$\{0, 1\}$	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		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$	$(-\frac{\pi}{2}, \frac{\pi}{2})$	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]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$[0, \infty)$	C^0
Leaky rectified linear unit (Leaky ReLU)[10]		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$	C^0
Parametric rectified linear unit (PReLU)[11]		$f(\alpha, x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(\alpha, x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$	C^0
Randomized leaky rectified linear unit (RRReLU)[12]		$f(\alpha, x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ [1]	$f'(\alpha, x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$	C^0
Exponential linear unit (ELU)[13]		$f(\alpha, x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(\alpha, x) = \begin{cases} f(\alpha, x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\alpha, \infty)$	C^1 when $\alpha = 1$, otherwise C^0

Activation functions approximate the sharp decision boundary.

Source: Wikipedia

How many binary functions can on model using a single Perceptron?



Source: R. Rojas, Intro to Neural Networks

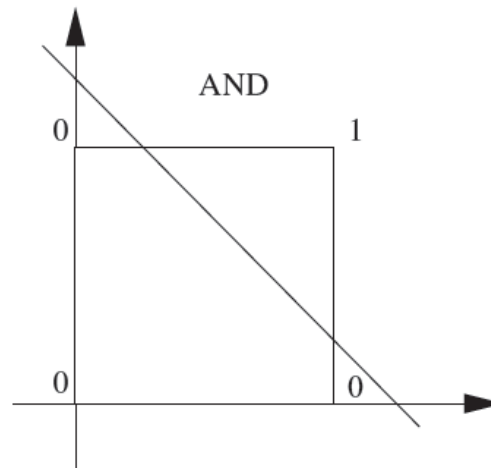
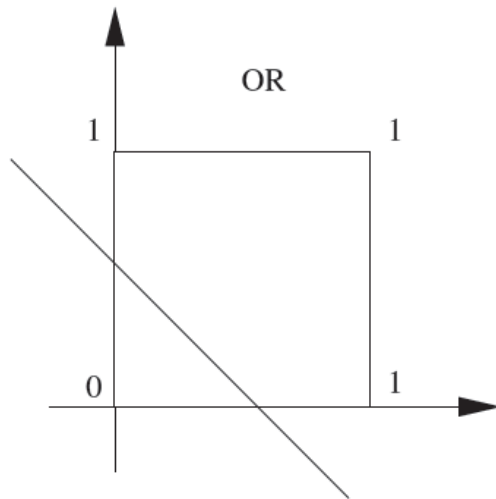
Example: Boolean Functions

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} possible labelings of v boolean variables

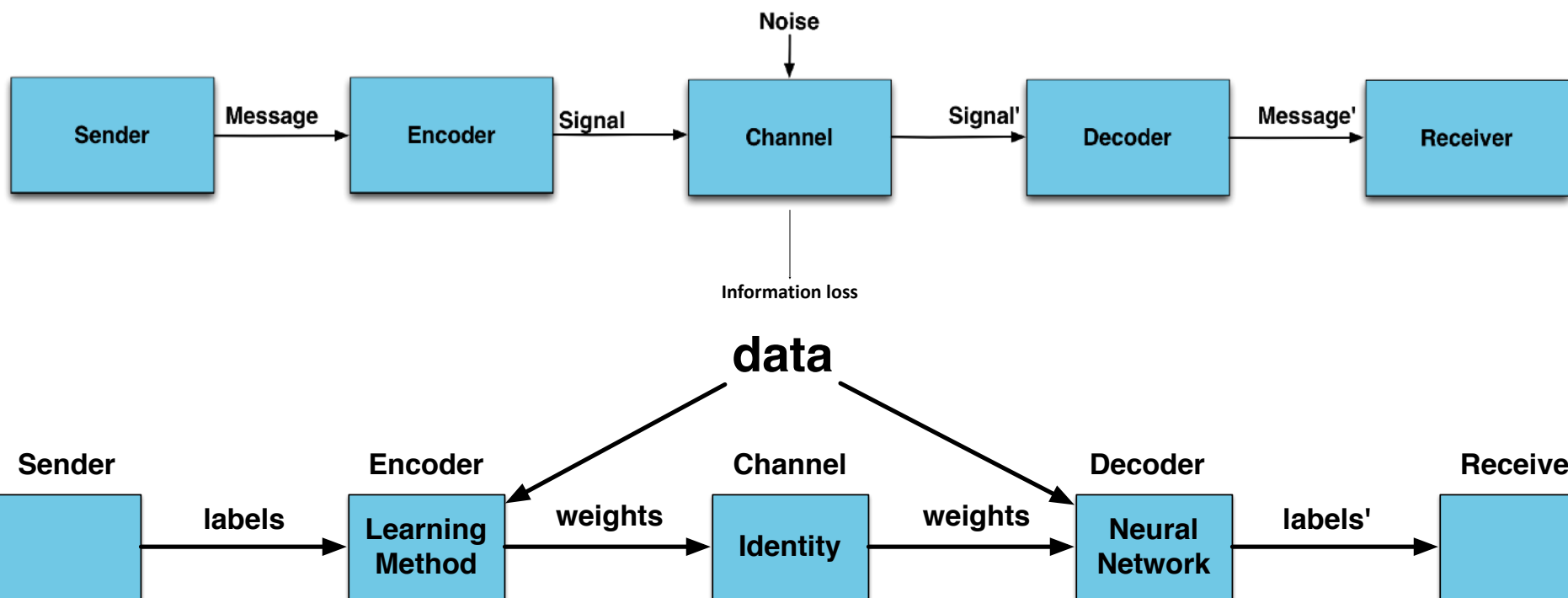
- 2^{2^v} labelings of 2^v points.

- For $v=2$, all but 2 functions work: XOR, NXOR



Source: R. Rojas, Intro to Neural Networks

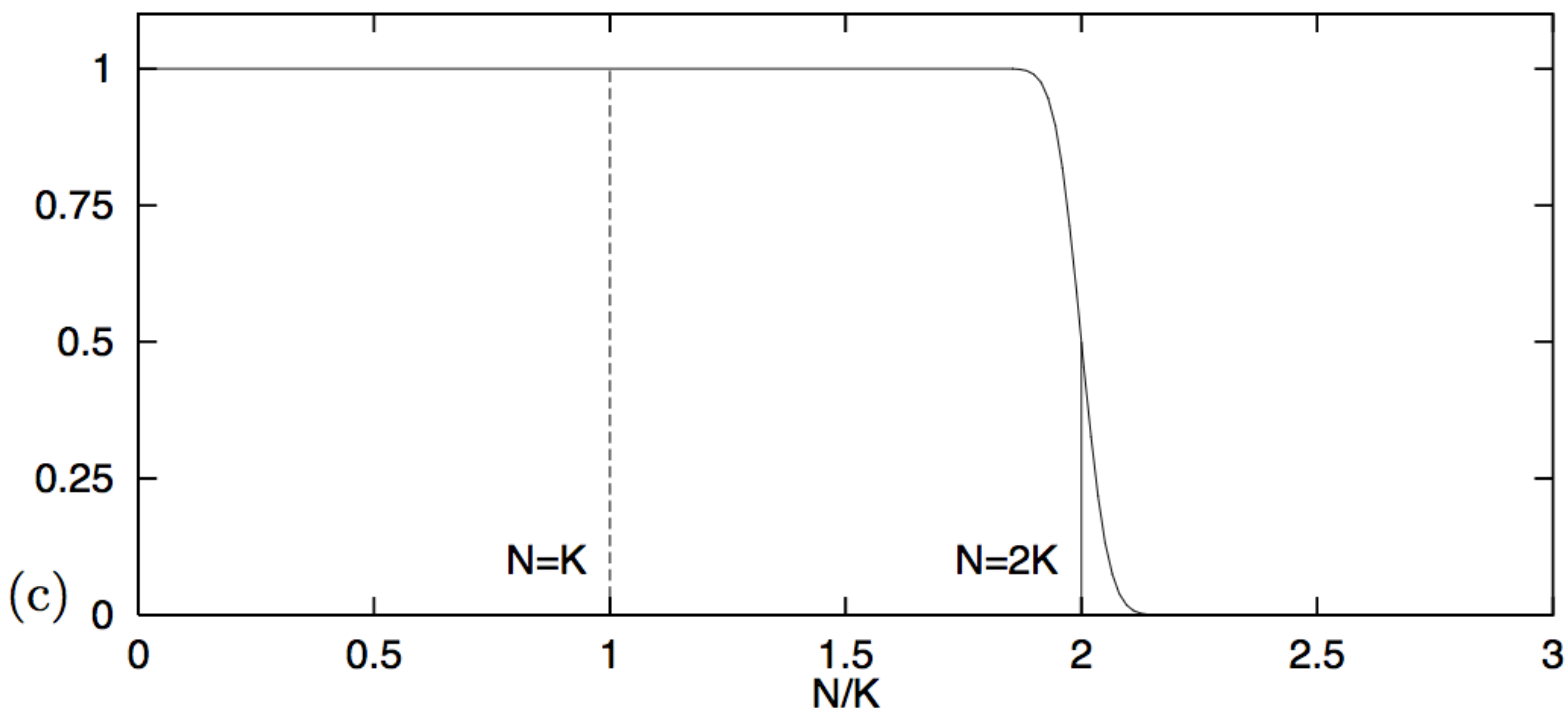
Machine Learning as an Encoder/Decoder



Main trick: Let the Machine Learner label random points!

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

Critical Points: Perceptron (Cover, MacKay)



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

$N=2K$: Cover/MacKay Information Capacity

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

Gerald Friedland, <http://www.gerald-friedland.org>³

From a Perceptron to Perceptron Networks

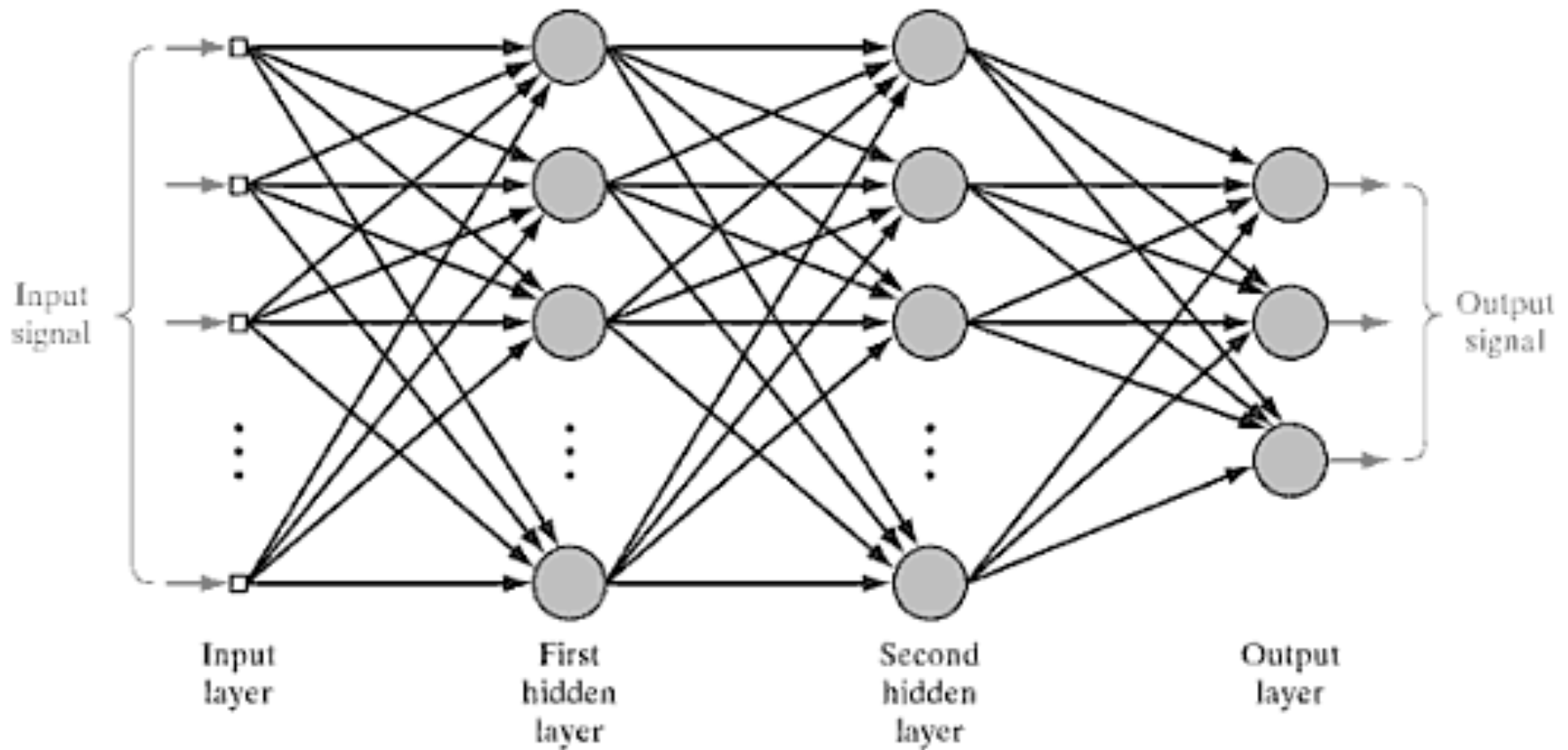
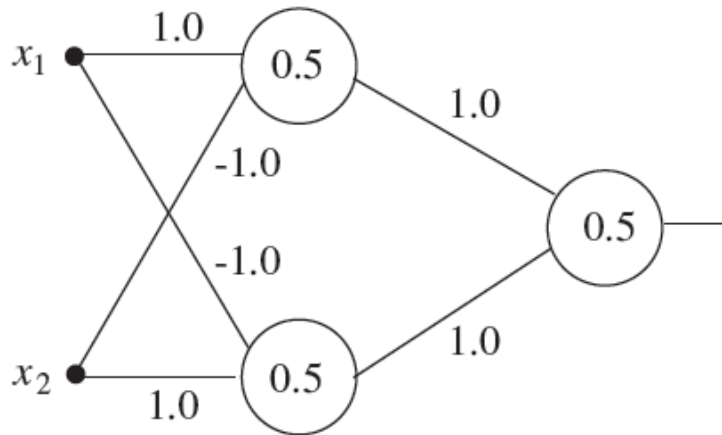


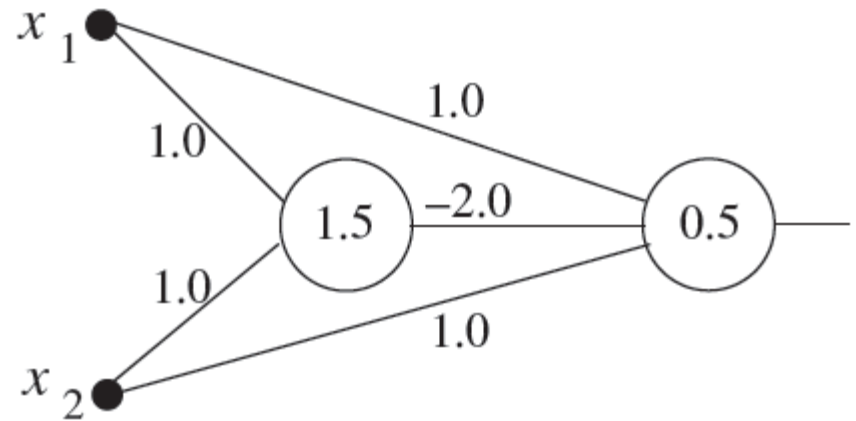
FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

Source: Wikipedia

Careful: Other Architectures



Typical MLP



Shortcut Network

Example Solutions to XOR

Source: R. Rojas, Intro to Neural Networks

Solution: Calculate in bits!

Assume: $y_i, x_i \in \{0, 1\}$, x_i uniformly distributed

n bits of memory: $f(x_1, \dots, x_n) = x_1, \dots, x_n$. (identity function).

Machine Learner:

binary classifier: $f(x_1, \dots, x_n) = y_1$

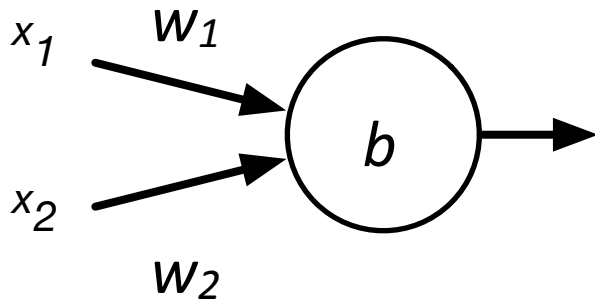
multi-class/regression: $f(x_1, \dots, x_n) = y_1, \dots, y_m$

Memory Equivalent Capacity: The number of configurations of uniformly distributed x_1, \dots, x_n that a machine learner can guarantee to label correctly.

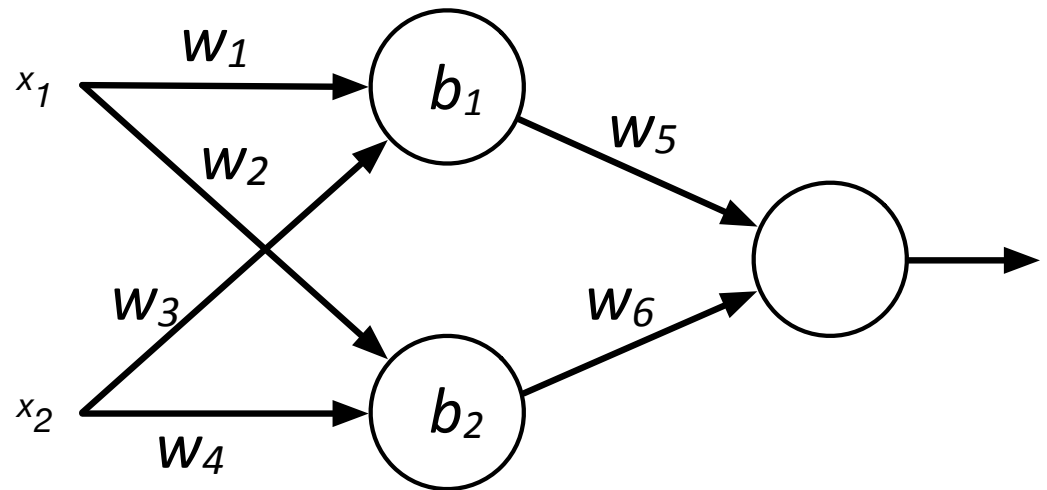
Memory Equivalent Capacity for Neural Networks

- 1) The output of a perceptron is maximally 1 bit.
- 2) The maximum memory capacity of a perceptron is the number of parameters (including bias) in bits.
(MacKay 2003)
- 3) The maximum memory capacity of perceptrons in parallel additive.
(MacKay 2003 speculative, Friedland and Krell 2017)
- 4) The maximum memory capacity of a layer of perceptrons depending on a previous layer of perceptrons is limited by the maximum output (in bits) of the previous layer.
(Data Processing Inequality, Tishby 2012)

Examples: How many bits of maximal capacity?

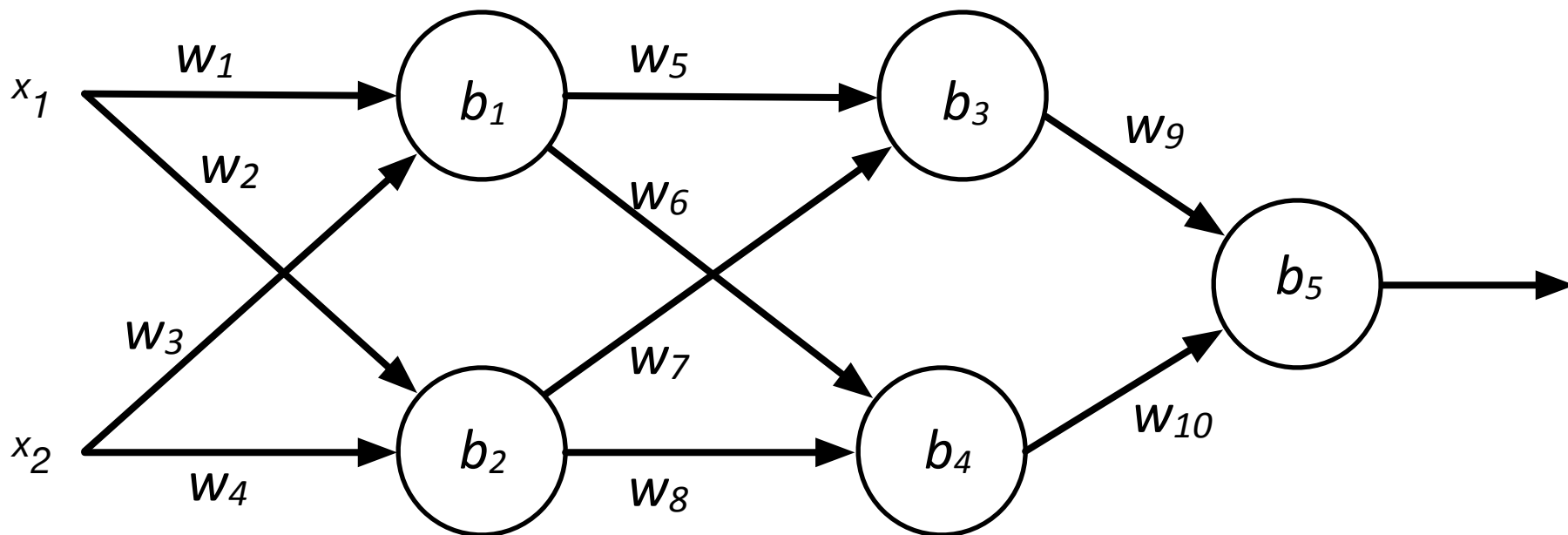


3 bits



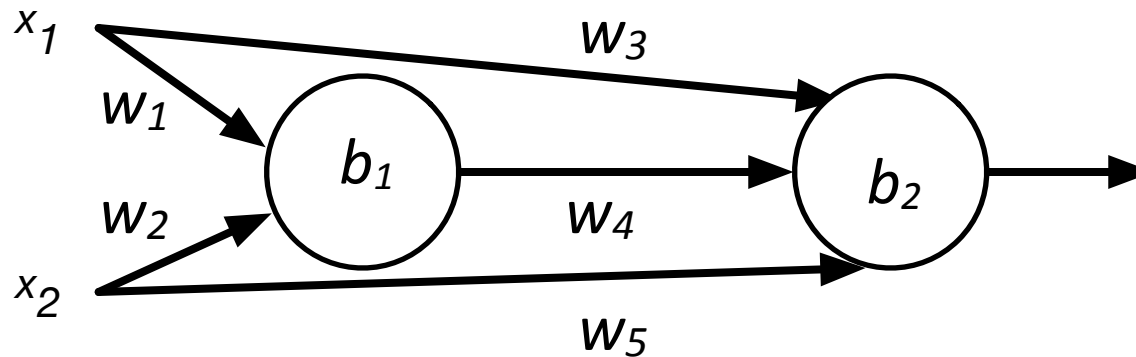
$2 * 3$ bits + $\min(2, 3)$ bits = 8 bits

Examples: How many bits of maximal capacity?



$$2 * 3 \text{ bits} + \min(2, 2 * 3) \text{ bits} + \min(2, 3) \text{ bits} = 10 \text{ bits}$$

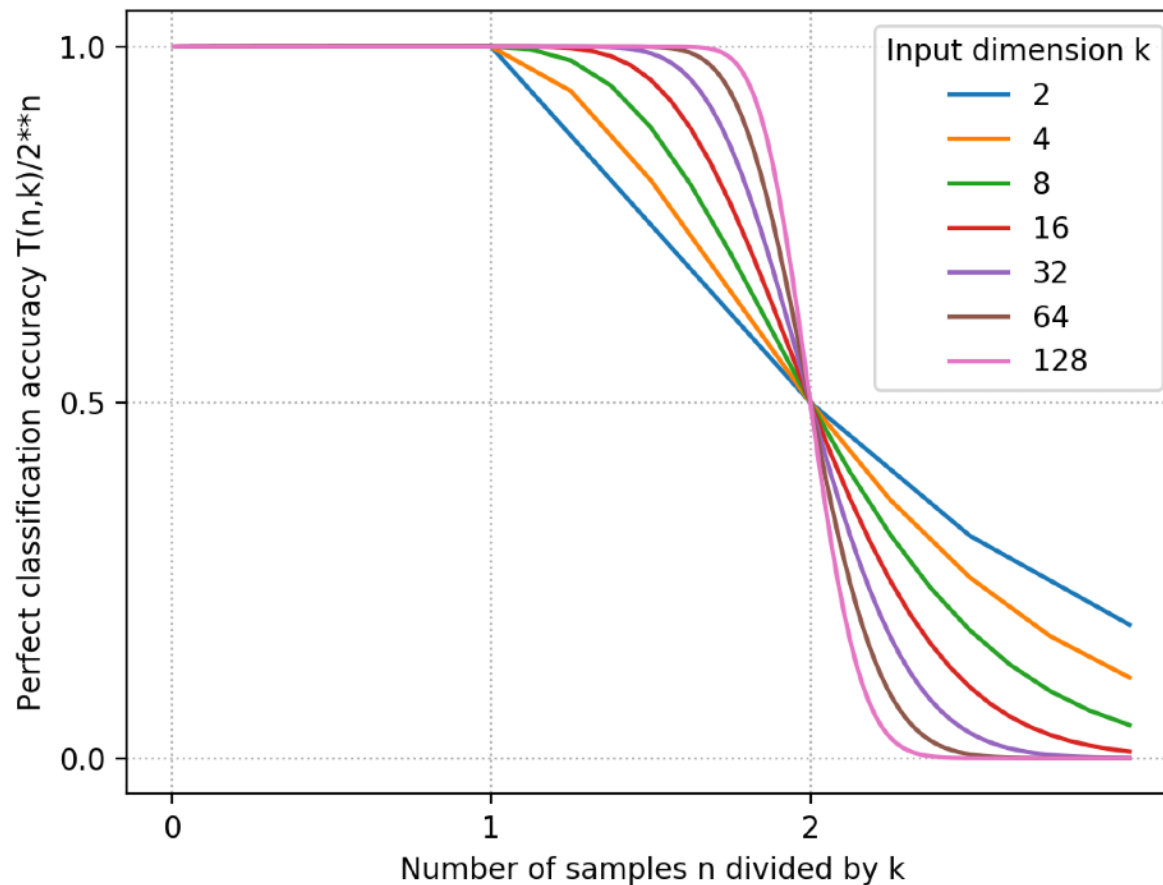
Examples: How many bits of maximal capacity?



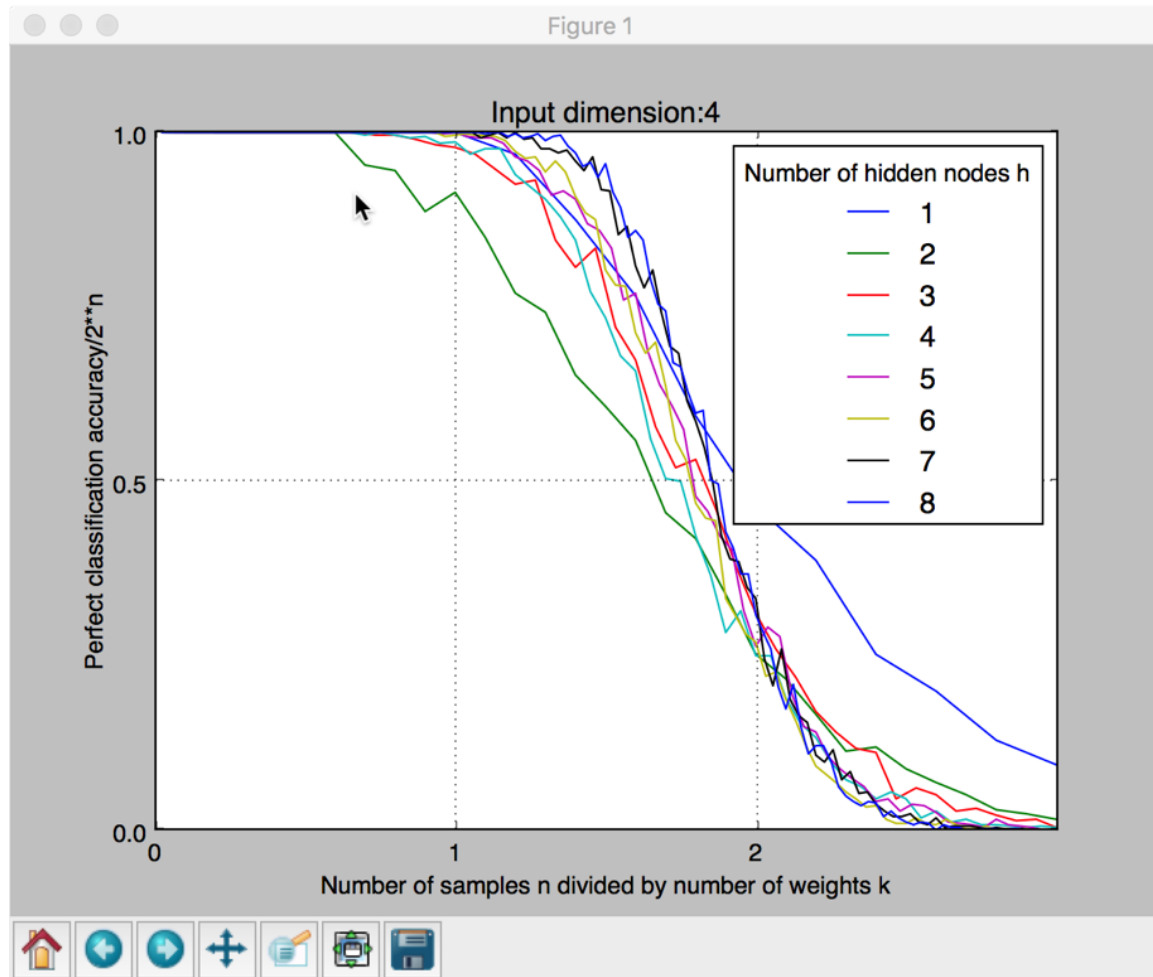
3 bits+4 bits=7 bits

Shortcut or ResNet

Characteristic Curve of a Theoretical 3-Layer MLP



Characteristic Curve of an Actual 3-Layer MLP



Python scikit-learn, 3-Layer MLP

Predicting Capacity Requirements

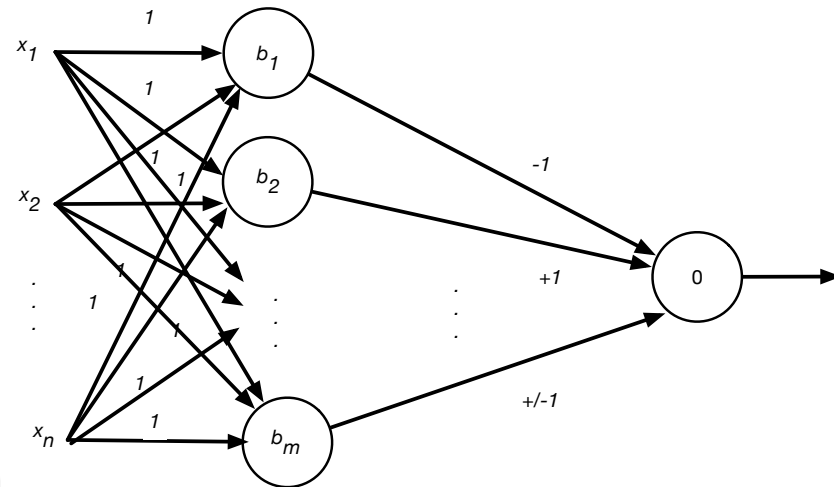
Given data and labels: How much actual capacity do I need to memorize the function?

Idea:

- 1) Worst case: Let's build a memorization network where only the biases are trained
- 2) Expected case: How much parameter reduction can (exponential) training buy us?

Predicting Maximum Memory Equivalent Capacity

```
data: array of length  $i$  containing vectors  $x$  with
dimensionality  $d$ 
labels: a column containing 0 or 1
MaxCapReq(data, labels)
thresholds  $\leftarrow 0$ 
loop over  $i$ :  $table[i] \leftarrow (\sum x[i][d], label[i])$ 
sortedtable  $\leftarrow sort(table, key = column\ 0)$ 
class  $\leftarrow 0$ 
loop over  $i$ : if not sortedtable[ $i$ ][1] == class then
|   class  $\leftarrow sortedtable[i][1]$ 
|   thresholds  $\leftarrow thresholds + 1$ 
end
maxcapreq  $\leftarrow thresholds * d + thresholds + 1$ 
expcapreq  $\leftarrow \log_2(thresholds + 1) * d$ 
print "Max: "+maxcapreq+" bits"
print "Exp: "+expcapreq+" bits"
```



“Dumb” Network

Runtime: $O(n \log n)$

Predicting Expected Minimum Memory Equivalent Capacity

Dumb Network:

- Highly inefficient.
- Potentially not 100% accurate (hash collisions).
- We can assume training weights (and biases) gets 100% accuracy while reducing parameters.

Expected Reduction: Exponential!

n thresholds should be able to be represented with $\log_2 n$ weights and biases (search tree!).

Empirical Results

Dataset	Max Capacity Requirement	Expected Capacity Requirement	Validation (% accuracy)
AND, 2 variables	4 bits	2 bits	2 bits (100%)
XOR, 2 variables	8 bits	4 bits	7 bits (100%)
Separated Gaussians (100 samples)	4 bits	2 bits	3 bits (100%)
2 Circles (100 samples)	224 bits	12 bits	12 bits (100%)
Checker pattern (100 samples)	144 bits	12 bits	12 bits (100%)
Spiral pattern (100 samples)	324 bits	14 bits	24 bits (98%)
ImageNet: 2000 images in 2 classes	906984 bits	10240 bits	10253 bits (98.2 %)

All results repeatable at: <https://github.com/fractor/nntailoring>

Training

- Everything we did so far assumes perfect training. This is, training that guarantees to reach the global minimum error.
- Perfect training requires exponential time.
- Imperfect training means Memory Equivalent Capacity is effectively reduced.
- How to measure that: ?

From Memorization to Generalization

Memorization is worst-case generalization.

Good news:

- Real-world data is not random.
- The information capacity of a perceptron is usually >1 bit per parameter (Cover, MacKay).

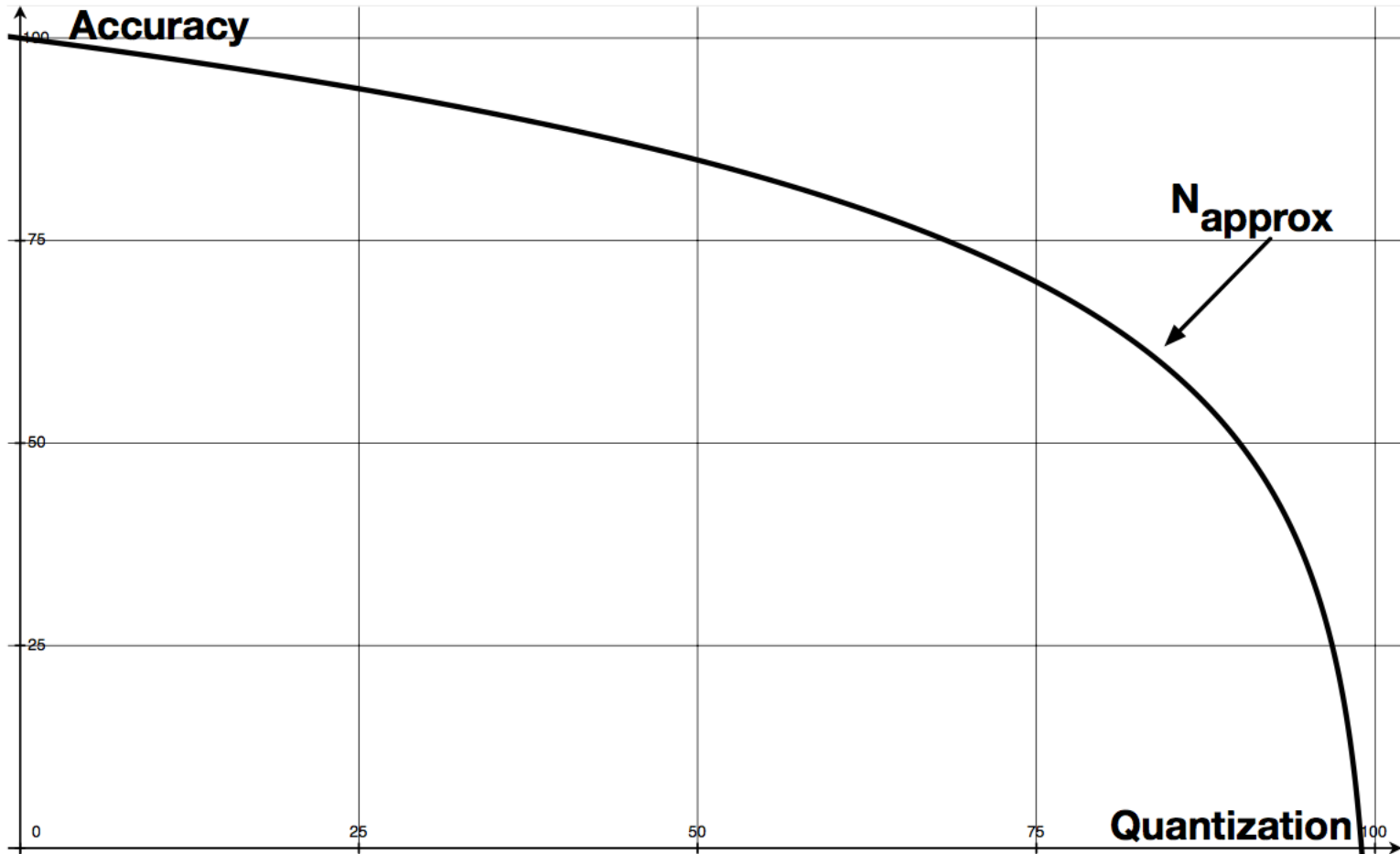
This means, we should be able to use less parameters than predicted by memory capacity calculations.

Suggested Engineering Process for Generalization

- Start at approximate expected capacity.
- Train to >98% accuracy. If impossible, increase parameters.
- Retrain iteratively with decreased capacity while testing against validation set.
Should see: decrease in training accuracy with increase in validation set accuracy
- Stop at minimum capacity for best held-out set accuracy.

Best case scenario: As parameters are reduced, neural network fails to memorize only the insignificant (noise) bits.

Generalization Process: Expected Curve



Overcapacity Machine Learning: Issues

- Waste of money, energy, and time. Bad for environment.
- The less parameters => the better the generalization rule => the higher adaptation per parameter => the higher the chance an unseen instance can be predicted correctly.
- Less parameters give a higher chance for explainability (Occam's Razor). See: G. Friedland, A. Metere: "*Machine Learning for Science*", UQ SciML Workshop, Los Angeles, June 2018.

Reminder: Occam's Razor

Among competing hypotheses, the one with the fewest assumptions should be selected.

For each accepted explanation of a phenomenon, there may be an extremely large, perhaps even incomprehensible, number of possible and more complex alternatives, because one can always burden failing explanations with ad hoc hypotheses to prevent them from being falsified; therefore, simpler theories are preferable to more complex ones because they are more testable.

(Wikipedia, Sep. 2017)

General Generalization

- Binary classifier (repeat):

$$G = \frac{\text{\#correctly classified instances}}{\text{Memory Equivalent Capacity}} \left[\frac{\text{bits}}{\text{bit}} \right]$$

- Multi-class/regression:

$$G = \frac{\text{\#correctly classified instances}}{\text{\#instances that can be memorized}}$$

Non-Statistical Definition (Literature)

$\forall x, x' \exists \delta$ such that $|x - x'| < \delta \implies f(x) = f(x')$

$x \in$ Training data

Informally: When do two different inputs lead to the same machine learner output.

$x' \in$ Test data

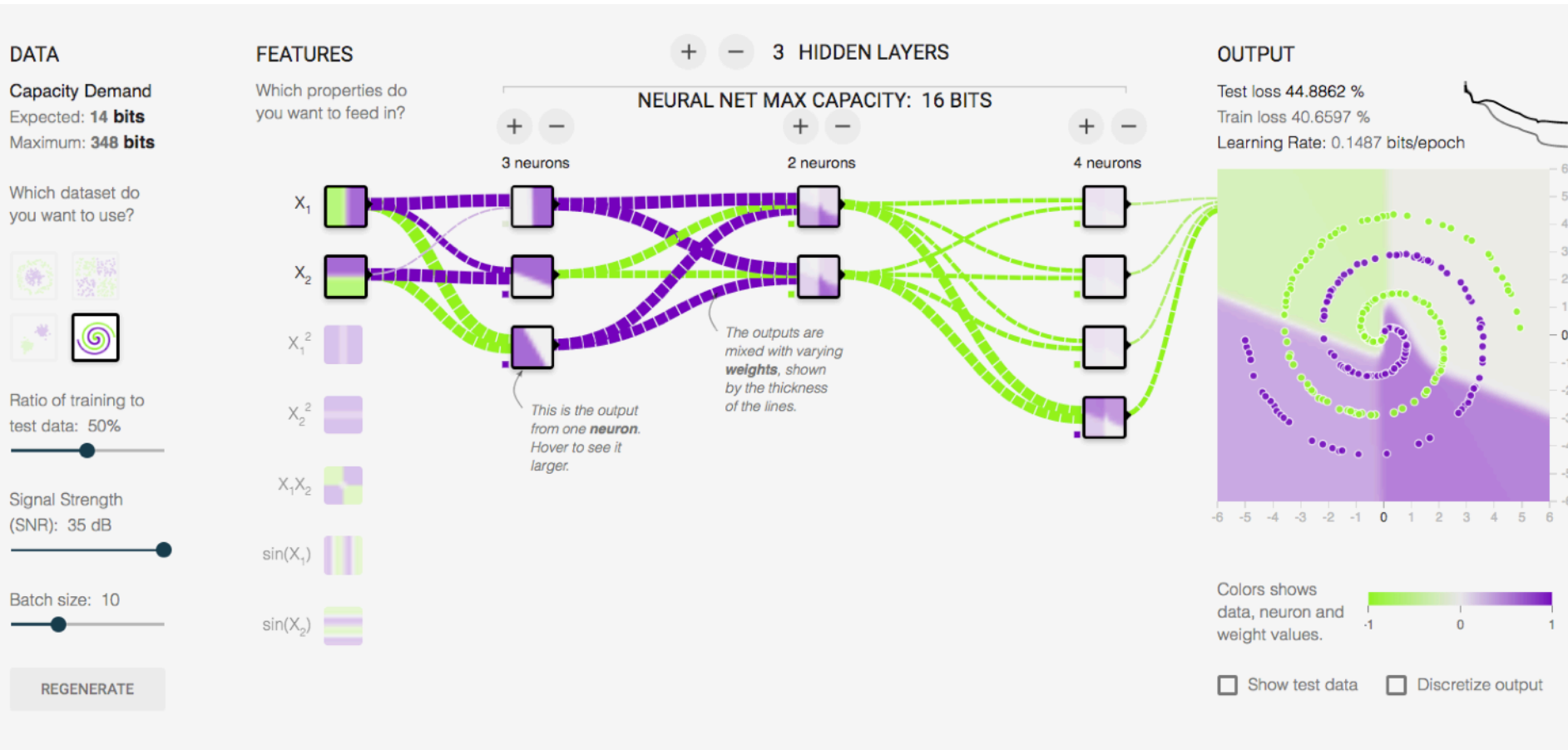
This is, which bits can be ignored in the comparison.

$|\circ|$ semi-metric

Statistical equivalent: How many bits per bit can be ignored on average (see G measure).

f machine learner

Demo: Experimental Design for TensorFlow



<http://tfmeter.icsi.berkeley.edu>