ACM

ACMMULTIMEDIA 2019

21 - 25 October 2019, Nice, France

Reproducibility and Experimental Design for Machine Learning

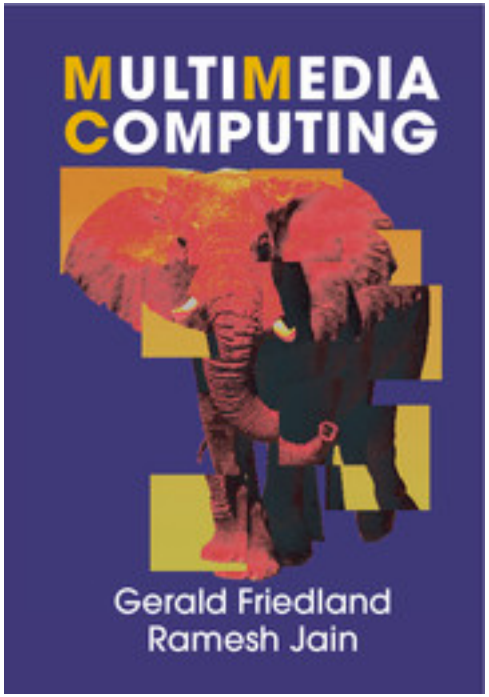
Gerald Friedland (UC Berkeley)

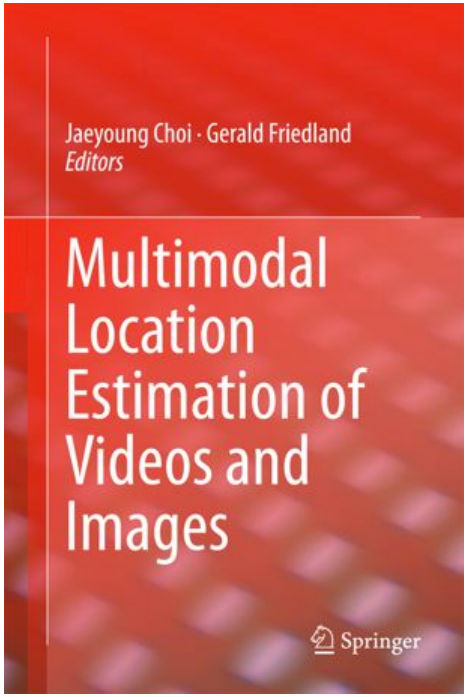
Results Reproduced
acm

Paper, Demo, etc:
https://tfmeter.icsi.berkeley.edu

About me….

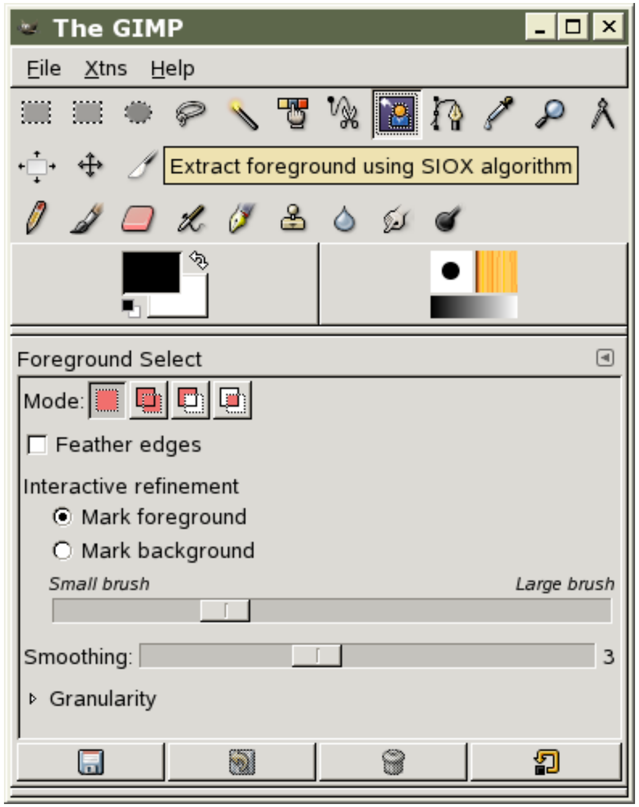
- Adjunct Faculty, UC Berkeley
- Data Scientist at National Lab
- Started work in Machine Learning in 2001




Jaeyoung Choi · Gerald Friedland
Editors
Multimodal Location Estimation of Videos and Images
Springer

TECHNOLOGY IN ACTION™
Beginning Programming Using Retro Computing
Learn BASIC with a Commodore Emulator
Gerald Friedland
Apress®

Some Previous Work



Foreground Select

Mode:

□ Feather edges

Interactive refinement

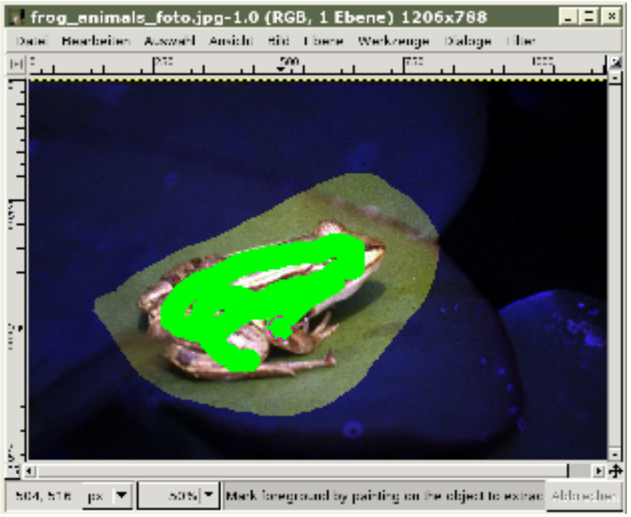
● Mark foreground

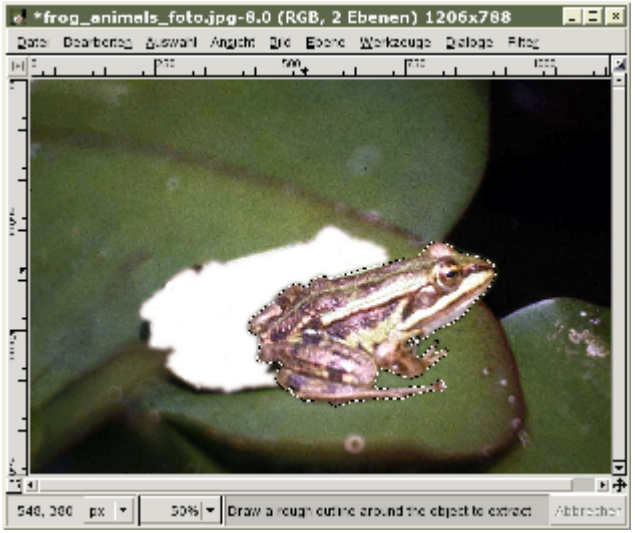
○ Mark background

Small brush — Large brush

Smoothing: 3

Granularity


frog_animals_foto.jpg-1.0 (RGB, 1 Ebene) 1206x788
Datei Bearbeiten Auswahl Ansicht Bild Ebene Werkzeuge Dialoge Filter
50%
Mark foreground by painting on the object to extract
Abbrechen


*frog_animals_foto.jpg-8.0 (RGB, 2 Ebenen) 1206x788
Datei Bearbeiten Auswahl Ansicht Bild Ebene Werkzeuge Dialoge Filter
548, 380 px
50%
Draw a rough outline around the object to extract
Abbrechen

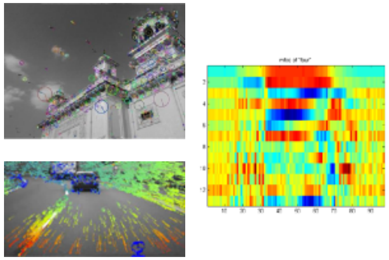
http://www.siox.org

G. Friedland, K. Jantz, T. Lenz, F. Wiesel, R. Rojas: *A Practical Approach to Boundary-Accurate Multi-Object Extraction from Still Images and Videos*, to appear in Proceedings of the IEEE International Symposium on Multimedia (ISM2006), San Diego (California), December, 2006

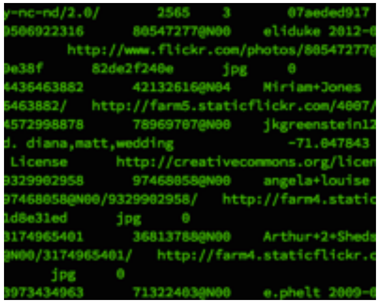
In GIMP, Krita, Inkscape, and other open source tools since 2005

The Multimedia Commons (YFCC100M)

100.2M Photos
800K Videos

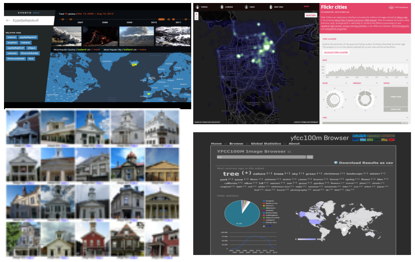


Features for Machine Learning
(Visual, Audio, Motion, etc.)



```
y-nc-nd/2.0/        2565    3        07aeded917
0506922316       80547277@N00    eliduke 2012-0
        http://www.flickr.com/photos/80547277@
0e38f     82de2f240e      jpg    0
4436463882       42132616@N04    Miriam+Jones
5463882/   http://farm5.staticflickr.com/4007/
4572998878       78969707@N00    jkgreenstein12
d. diana,matt,wedding                -71.047843
 License     http://creativecommons.org/licen
9329902958       97468058@N00    angela+louise
97468058@N00/9329902958/   http://farm4.static
1d8e31ed      jpg    0
3174965401       36813788@N00    Arthur+2+Sheds
@N00/3174965401/   http://farm4.staticflickr.c
     jpg    0
3973434963       71322403@N00    e.phelt 2009-0
```

User-Supplied Metadata
and New Annotations



Tools for Searching,
Processing, and Visualizing

100M videos and images, and a growing pool of **tools** for research with **easy access** through **Cloud Computing**

Collaboration Between Academia and Industry:

ICSI INTERNATIONAL COMPUTER SCIENCE INSTITUTE

LAWRENCE LIVERMORE NATIONAL LABORATORY

YAHOO! LABS

amazon web services™

Berkeley UNIVERSITY OF CALIFORNIA

Benchmarks & Grand Challenges:

DIGITAL VIDEO RETRIEVAL at NIST

ACM mm

2016 ICMR

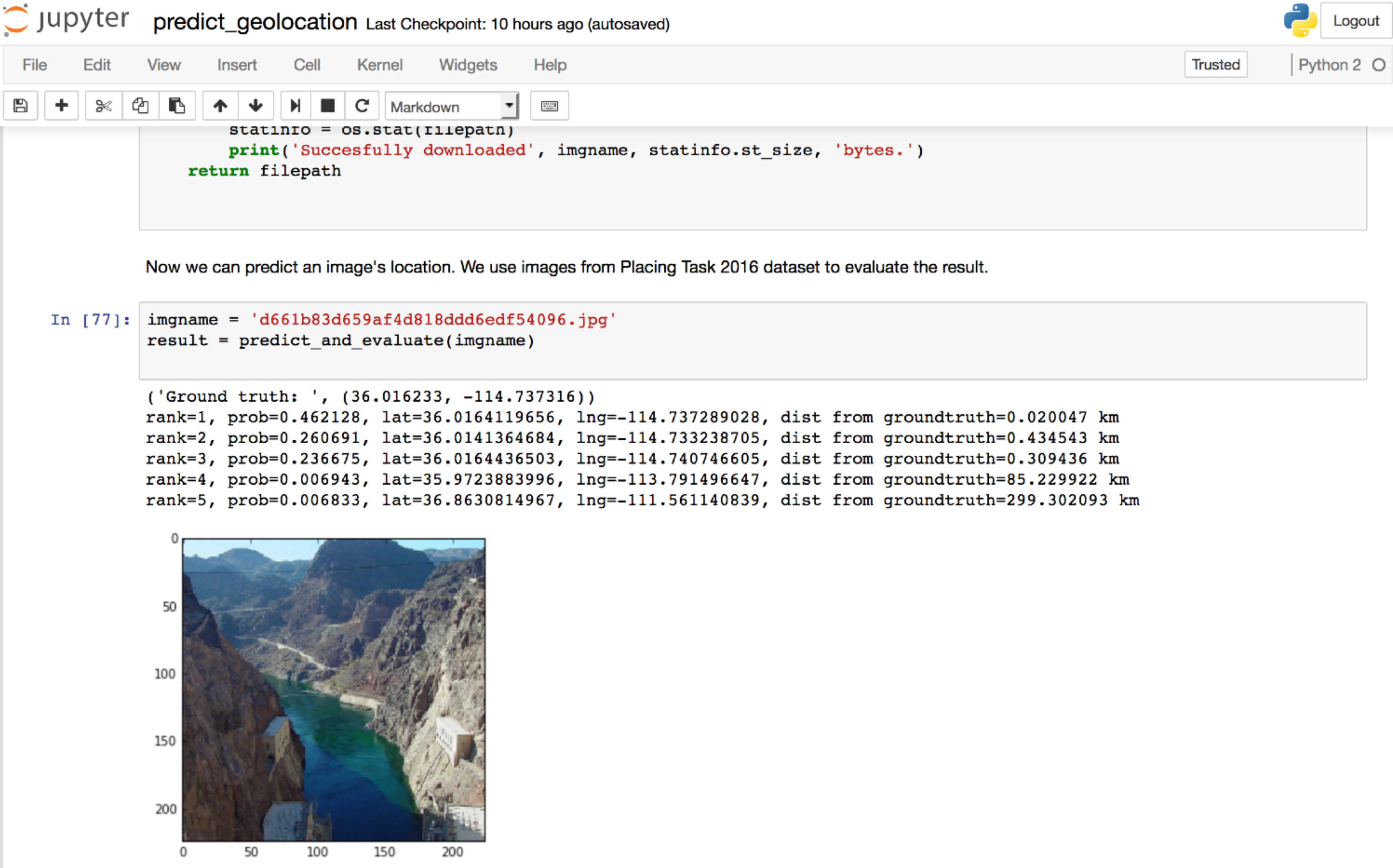
MediaEval Benchmark

Creative Commons or Public Domain

Supported in part by NSF Grant 1251276
“BIGDATA: Small: DCM: DA: Collaborative Research:
SMASH: Scalable Multimedia content AnalysiS in a High-level language”

Jupyter Integration

YFCC100M+MMCommons+Amazon's MXNet



```
        statinfo = os.stat(filepath)
        print('Succesfully downloaded', imgname, statinfo.st_size, 'bytes.')
    return filepath
```

Now we can predict an image's location. We use images from Placing Task 2016 dataset to evaluate the result.

```
In [77]: imgname = 'd661b83d659af4d818ddd6edf54096.jpg'
         result = predict_and_evaluate(imgname)
```

```
('Ground truth: ', (36.016233, -114.737316))
rank=1, prob=0.462128, lat=36.0164119656, lng=-114.737289028, dist from groundtruth=0.020047 km
rank=2, prob=0.260691, lat=36.0141364684, lng=-114.733238705, dist from groundtruth=0.434543 km
rank=3, prob=0.236675, lat=36.0164436503, lng=-114.740746605, dist from groundtruth=0.309436 km
rank=4, prob=0.006943, lat=35.9723883996, lng=-113.791496647, dist from groundtruth=85.229922 km
rank=5, prob=0.006833, lat=36.8630814967, lng=-111.561140839, dist from groundtruth=299.302093 km
```

Start of this work: Simple Question

- **How much** money (cpu time, memory, IO) do I need to budget for my deep learning experiment?

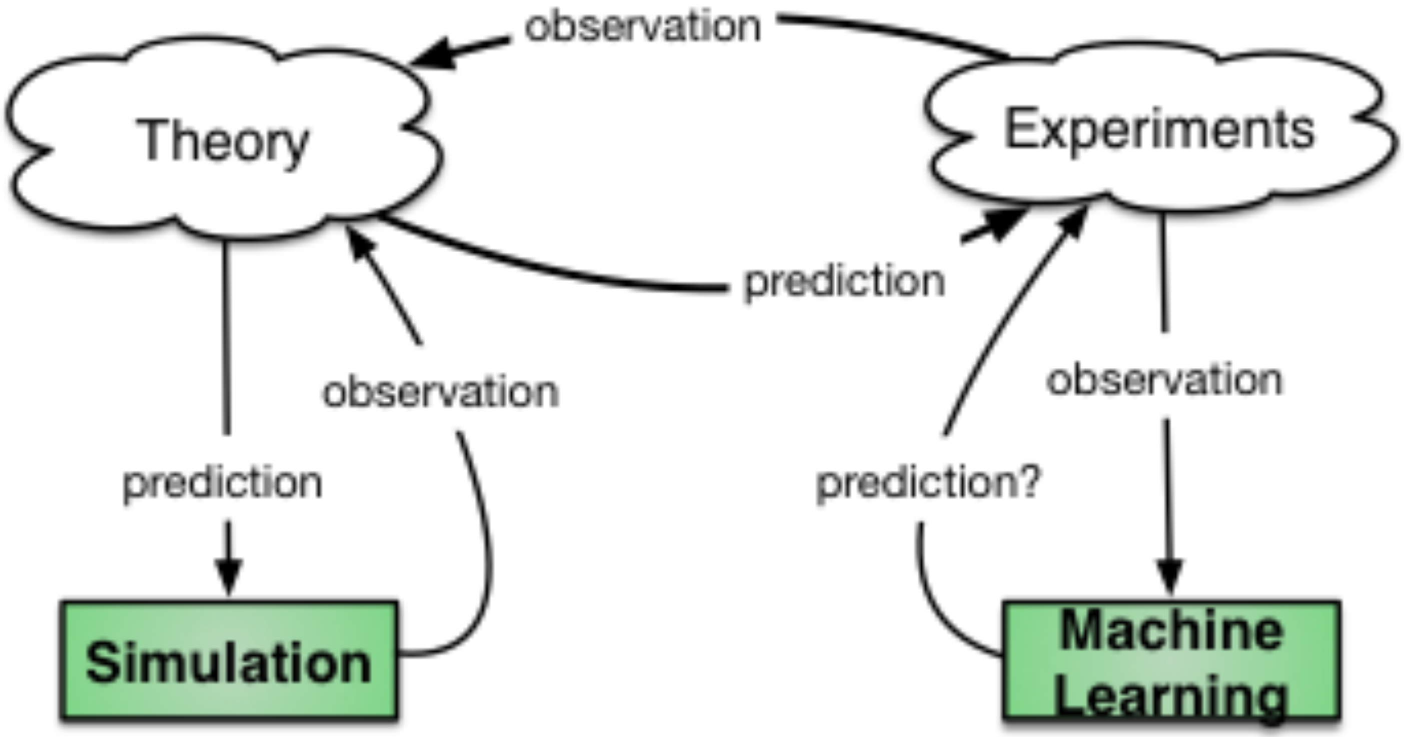
- State of the Art: No answer.
  For example, ImageNet models vary significantly:
  - AlexNet: 238MB model, 2.27Bn Ops
  - DarkNet: 28MB model, 0.96Bn Ops
  - VGG-16: 528 MB, 30.94Bn Ops

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

A game…

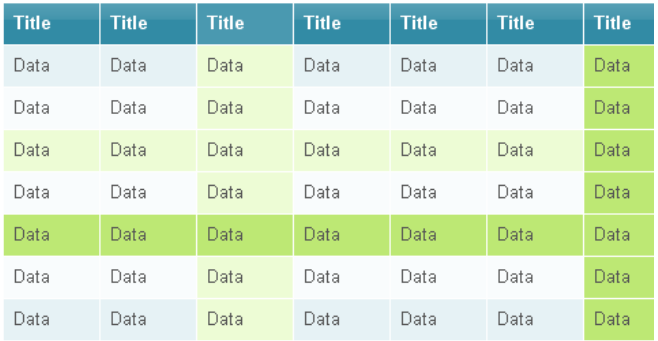
- Continue the sequence:
  - 2, 4, 6, 8, ….
  - 6, 5, 1, 4, …..
- What is the next number?
  - 100000 (sequence 1)
  - 100000 (sequence 2)
- Why?

The Scientific Method

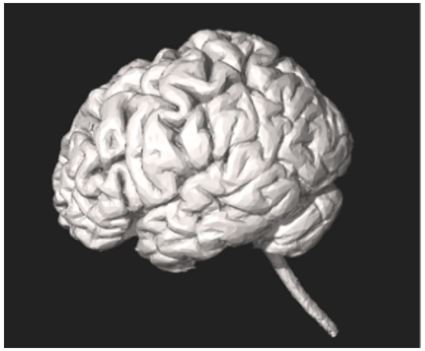


Data Science: The Science of Automating the Scientific Method

The Scientific Method: Practical (traditional)



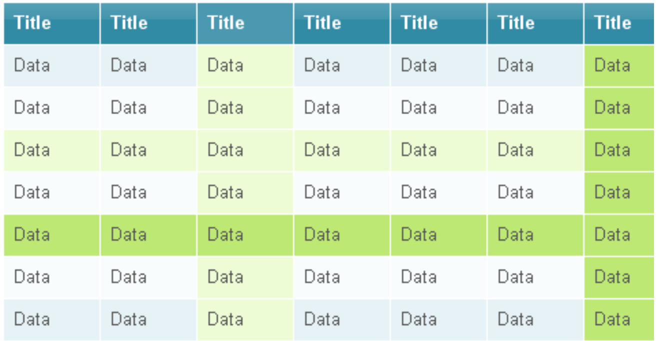
Title	Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	Data



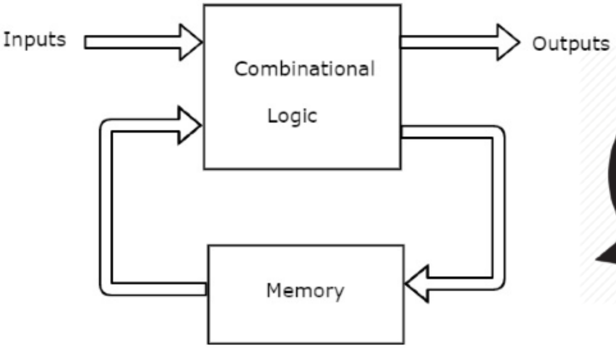
$$E = mc^2$$

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: Reproducibility

An experimental result is not fully established unless it can be independently reproduced.
https://project.inria.fr/acmmmreproducibility/

- Repetition of experiment: Predict outcome X, observe outcome X.
- Reproduction of an experiment: Independent party recreates the experimental setup of an experiment and repeats it.
- Reproduction requires a complete description of all factors that make the experiment repeatable.
- Understanding: What are the minimum amount of factors that make the experiment repeatable and how do they influence each other?

Repetition, Reproduction, Understanding?

- Downloading the docker instance an re-running on your campus.
- Downloading the trained model and re-running it on the same data.
- Using the exact software versions and exact configuration of hyper parameters and retraining the same model to obtain similar results.
- Taking the description from a paper, rebuilding the setup as described and obtaining the same accuracy and adversarial examples.
- Taking the description from a paper, rebuilding the setup as described, obtaining the same accuracy and adversarial examples and explaining why a different setup yields other limits.

What factors make Machine Learning Repeatable, Reproducible, Understandable?

- Repeatable:
  Set of all hyper parameters, seed, architecture, exact software versions of all libraries, exact order of training sample presentation, etc (?)

- Reproducible:
  Capacity, perfect training (i.e. training that guarantees to reach a global minimum error).

- Understandable:
  Minimum capacity, perfect training.

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:

$$(x_1, x_2, \ldots, x_m, f(\overrightarrow{x}))$$

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

- Output: State machine *M* that maps a point

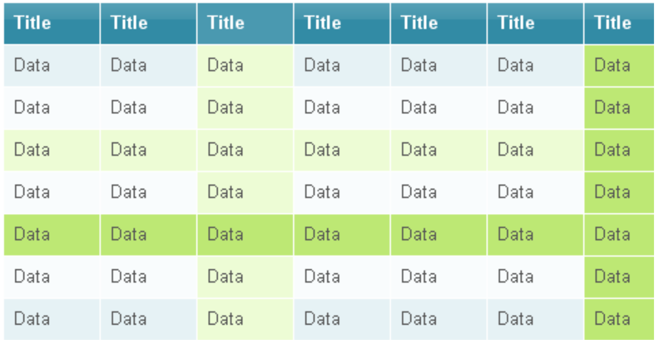
$$(x_1, x_2, \ldots, x_m) \implies f(\overrightarrow{x})$$

Thought Framework: Machine Learning

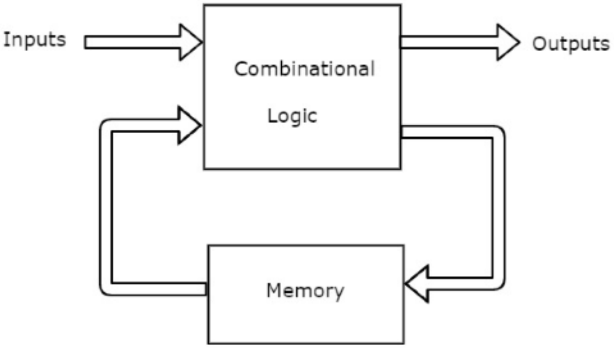
- Assume

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

(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

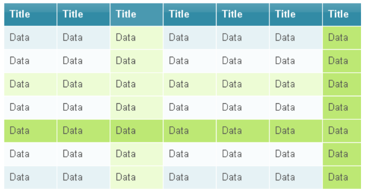
Thought Framework: Machine Learning

Assume

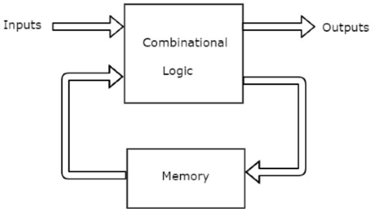
$$x_i \in \mathbb{R}, f(\overrightarrow{x}) \in \{0,1\}$$

(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?

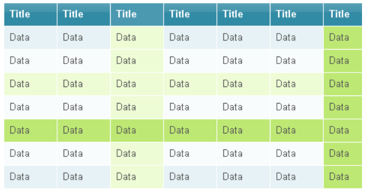
Maximally: #rows (lookup table)
Minimally: ? (Kolmogorov Complexity)

Thought Framework: Machine Learning

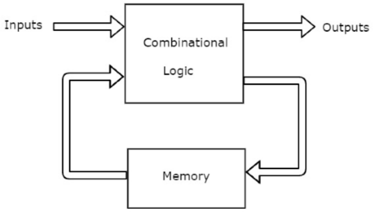
- ***Intellectual Capacity:*** *The number of unique target functions a machine learner is able to represent (as a function of the number of model parameters).*

- ***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



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:

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

G<1 => *M* needs more training/data (not even memorizing)
G=1 => *M* is memorizing = overfitting
1<G<G_{MEM} => M could be implementing a lossless compression
(and still overfit)
G> G_{MEM}=> *M* is generalizing (no chance for overfitting)

Maximum Lossless Compression

$$G_{MEM} = \frac{\sum_{i=1}^{|\Sigma|} p_i \log_2 p_i}{\log_2 |\Sigma|} \quad [\frac{bits}{bit}]$$

$\Sigma =$ Set of observations (rows in table)

$p_i =$ Probability to observe an element of Σ

Generalization in Machine Learning

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

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.

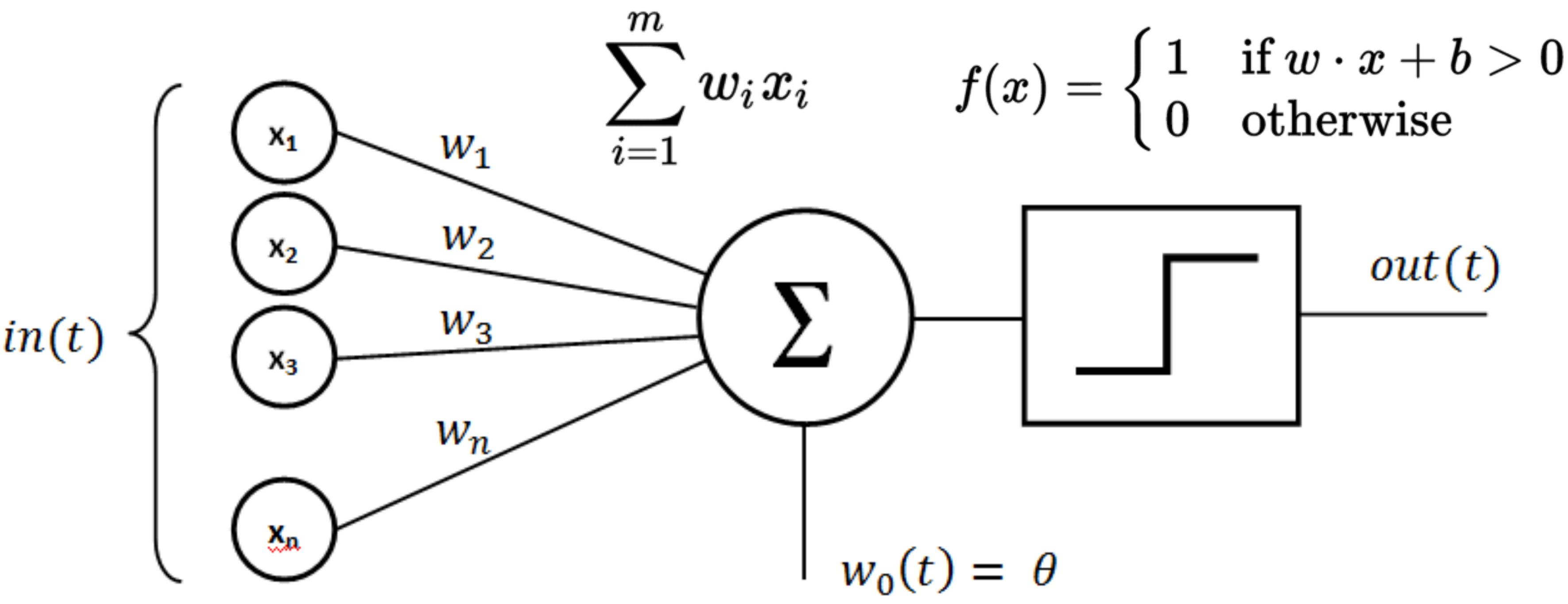
How do we calculate the Memory Equivalent Capacity?

- 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.

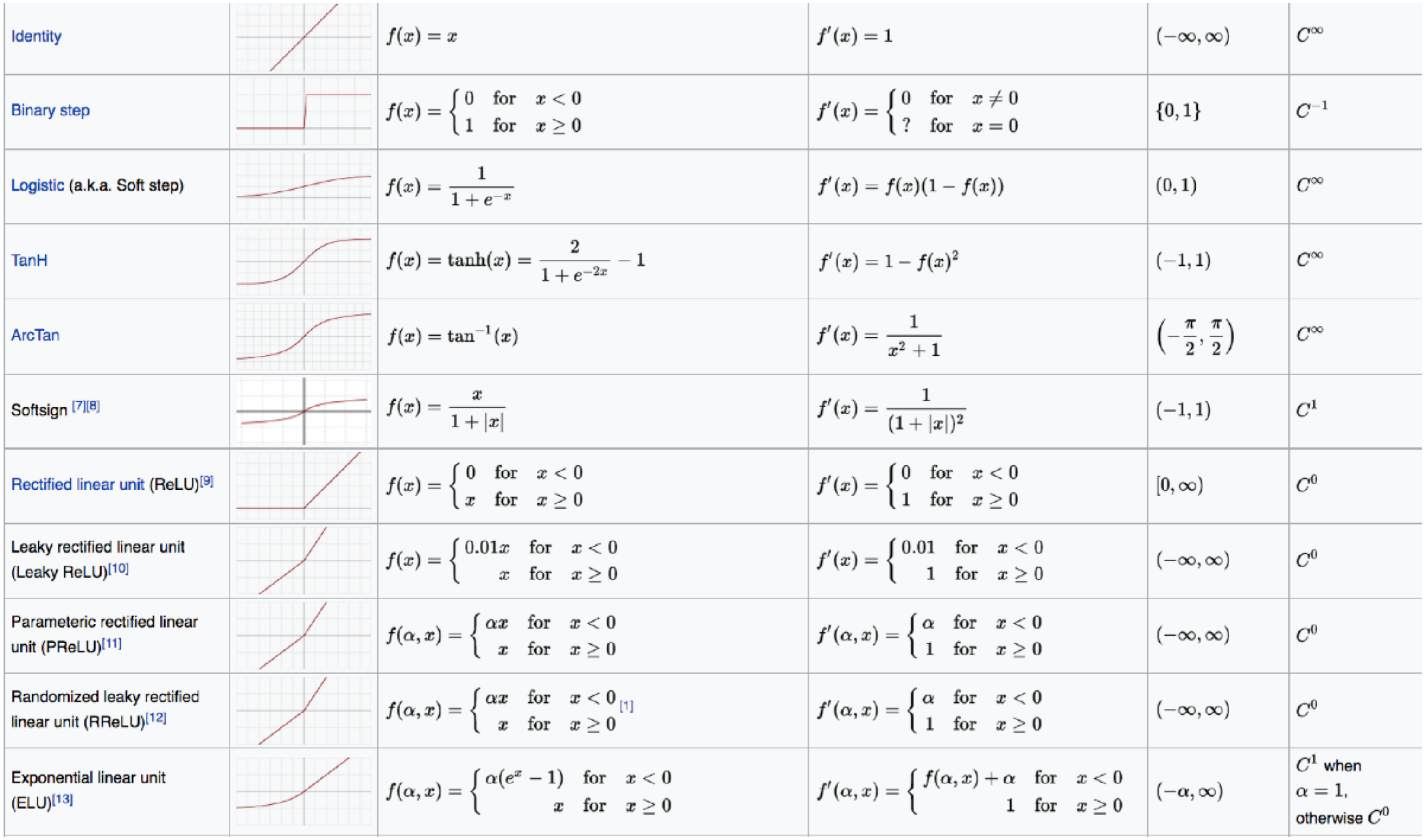
Repeat: The Perceptron



Physical interpretation: Energy threshold

Source: Wikipedia

Repeat: Activation Functions (too many)

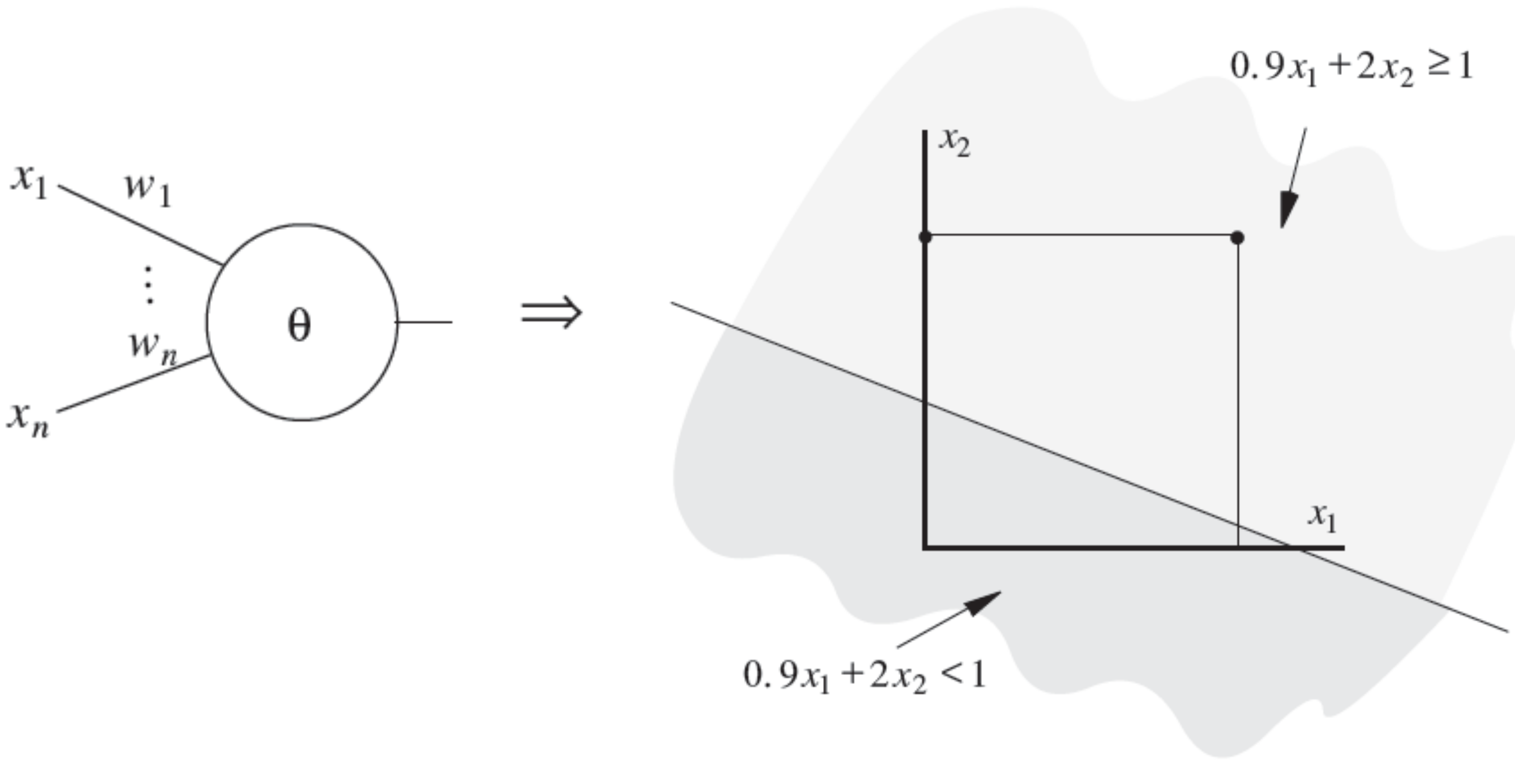


Identity		$f(x)=x$	$f'(x)=1$	$(-\infty,\infty)$	C^∞
Binary step		$f(x)=\begin{cases}0 & \text{for} \quad x<0\\ 1 & \text{for} \quad x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for} \quad x\neq 0\\ ? & \text{for} \quad x=0\end{cases}$	$\{0,1\}$	C^{-1}
Logistic (a.k.a. Soft step)		$f(x)=\frac{1}{1+e^{-x}}$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

Activation functions approximate the sharp decision boundary.

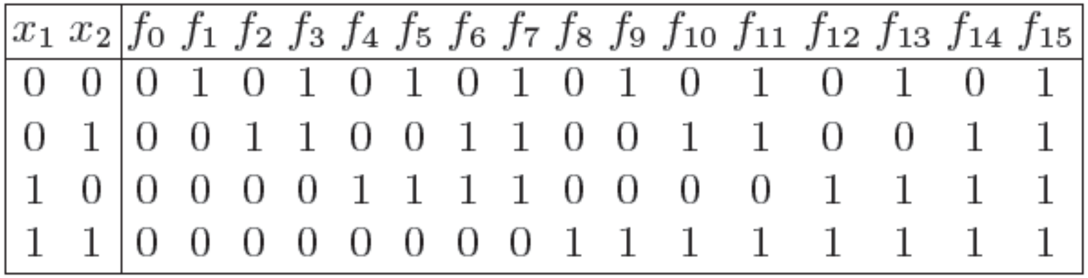
Source: Wikipedia

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



Source: R. Rojas, Intro to Neural Networks

Example: Boolean Functions

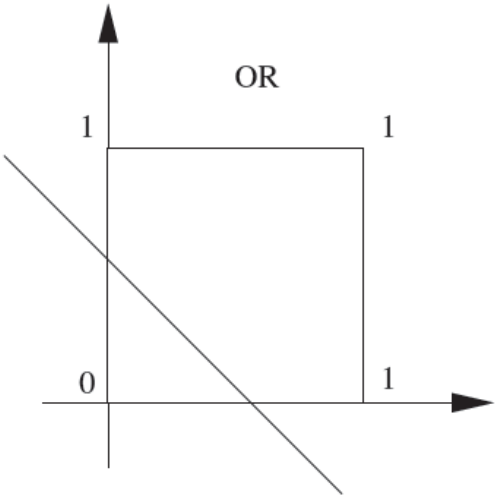


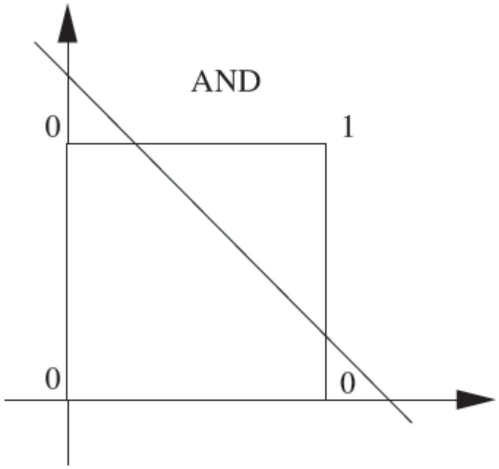
x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

- *2^{2^v} possible* labelings of *v* boolean variables

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

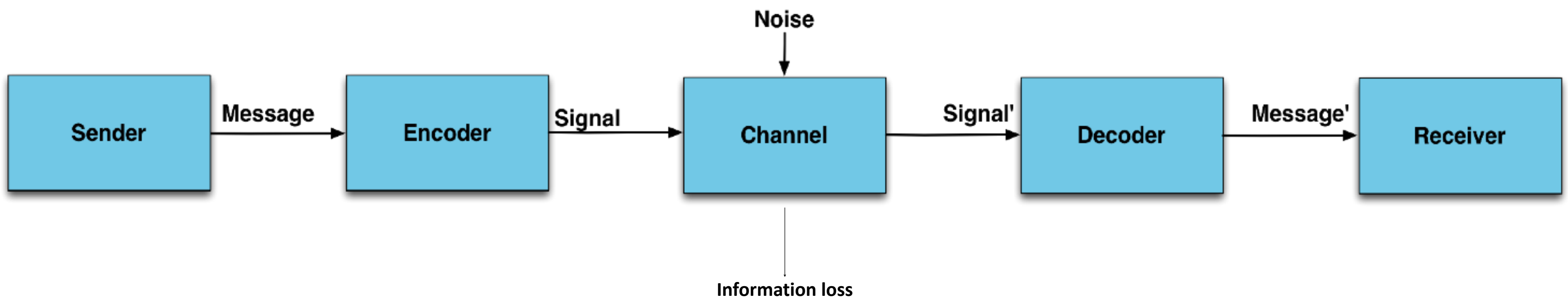
- For *v=2*, all but *2* functions work: XOR, NXOR

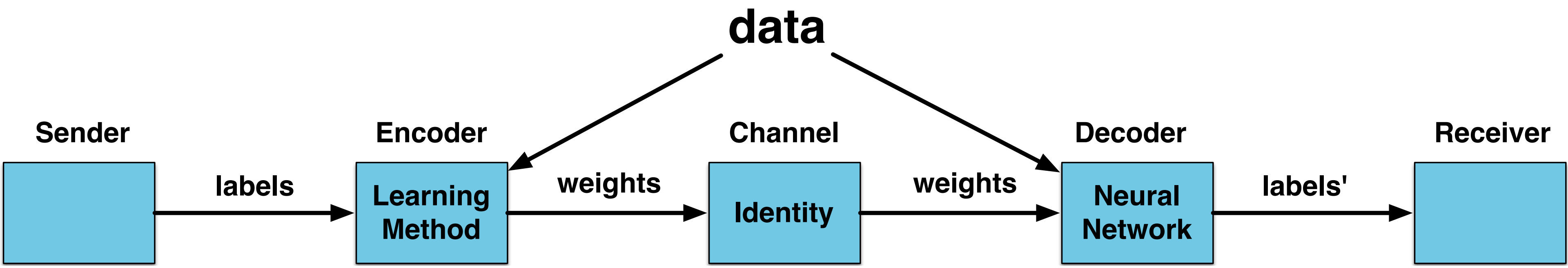




Source: R. Rojas, Intro to Neural Networks

Machine Learning as an Encoder/Decoder

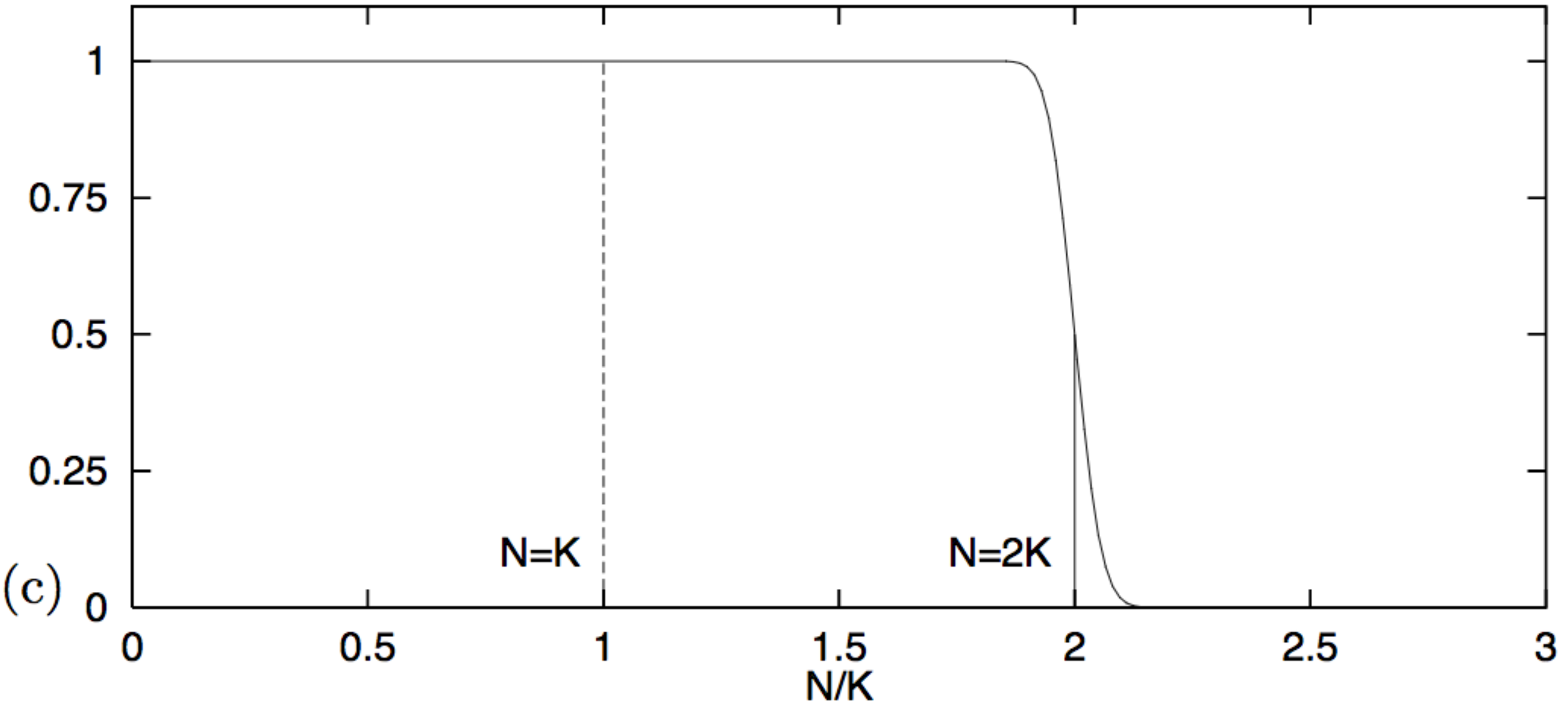




Main trick: Let the Machine Learner label random points!

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

From a Perceptron to Perceptron Networks

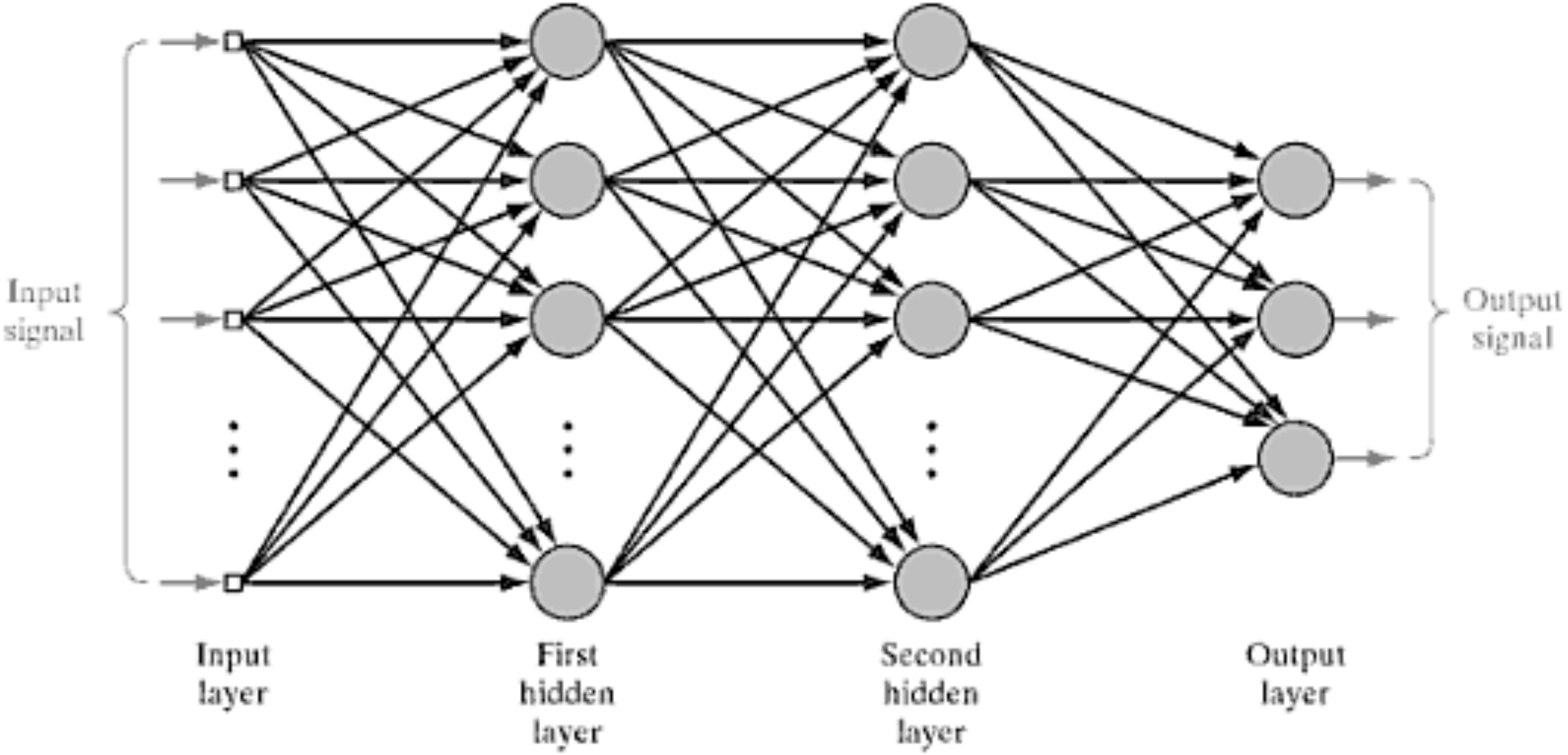
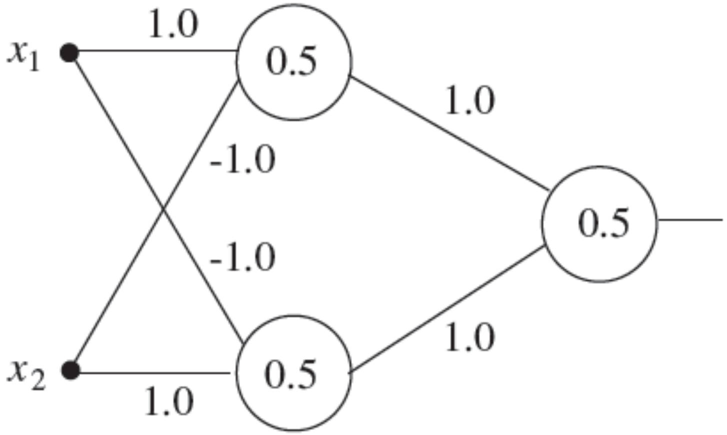


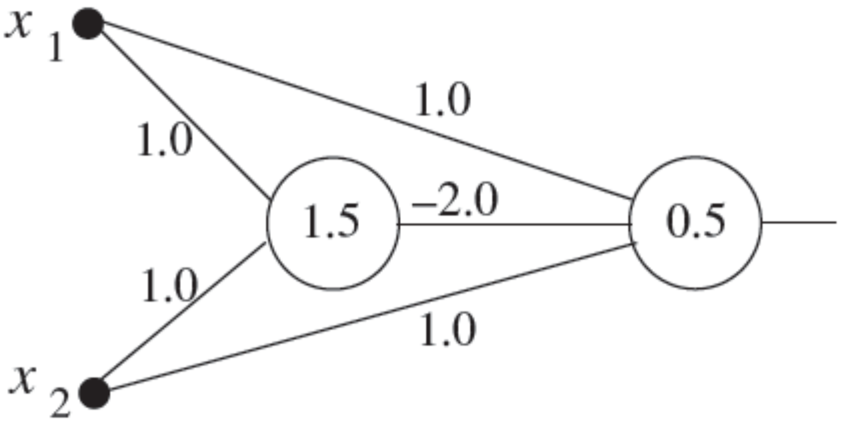
FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

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: $f(x_1,\ldots,x_n)=x_1,\ldots,x_n.$ (identity function).

Machine Learner:

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

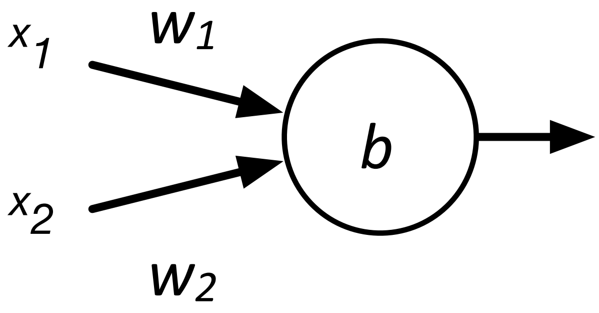
multi-class/regression: $f(x_1,\ldots,x_n)=y_{1,\ldots,}y_m$

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

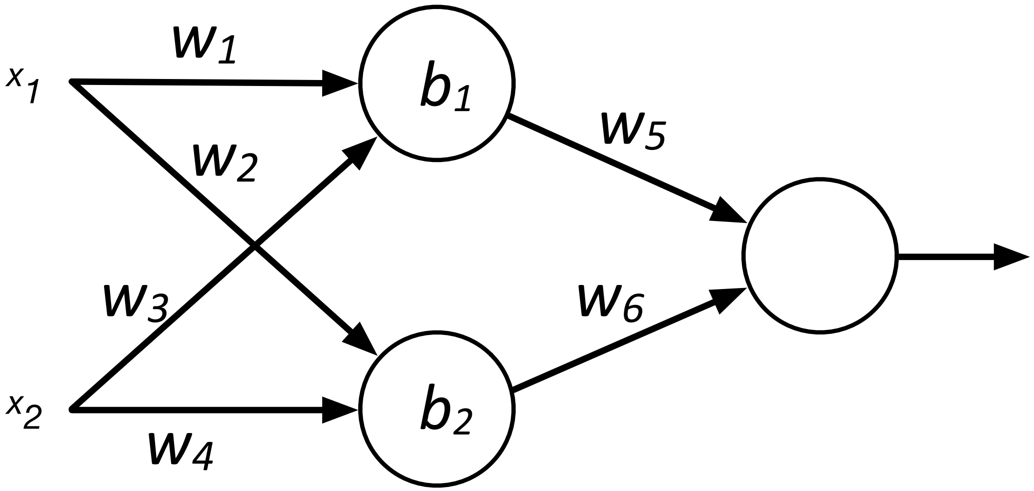
Memory Equivalent Capacity for Neural Networks

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 a neural network is additive, except 4.
   (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?

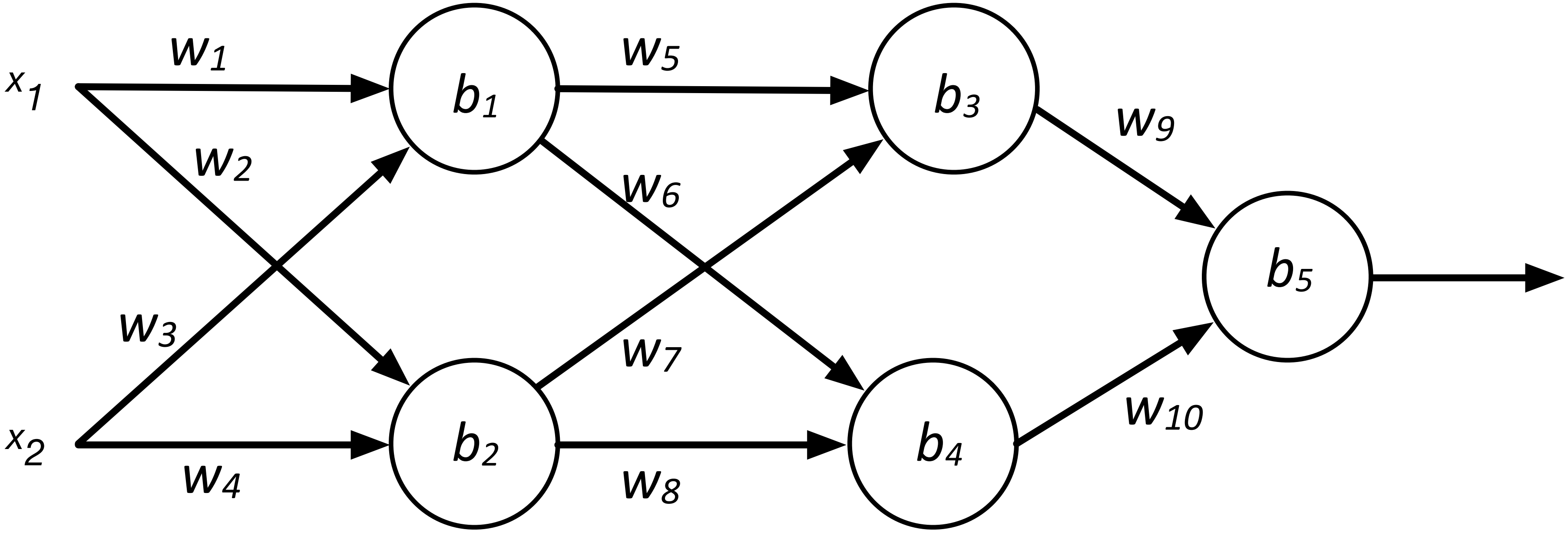


3 bits



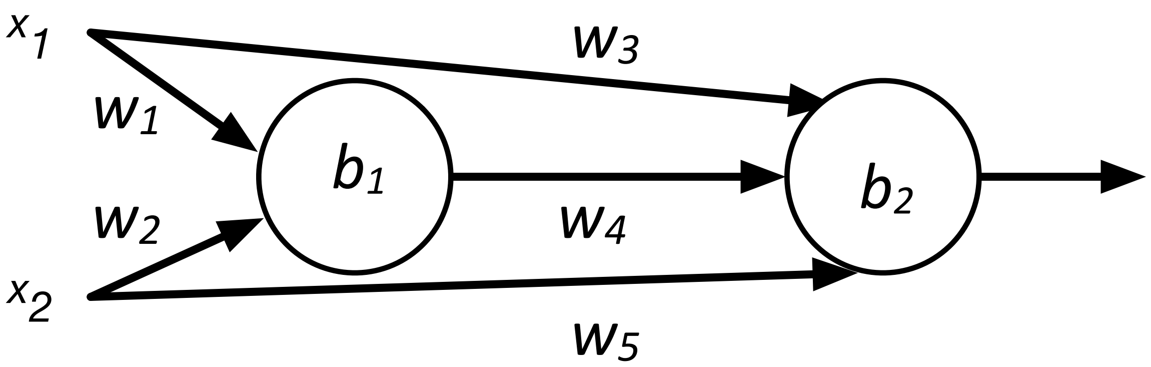
2*3 bits+min(2,3) bits=8 bits

Examples: How many bits of maximal capacity?



2*3 bits+min(2,2*3) bits+min(2,3) bits=10 bits

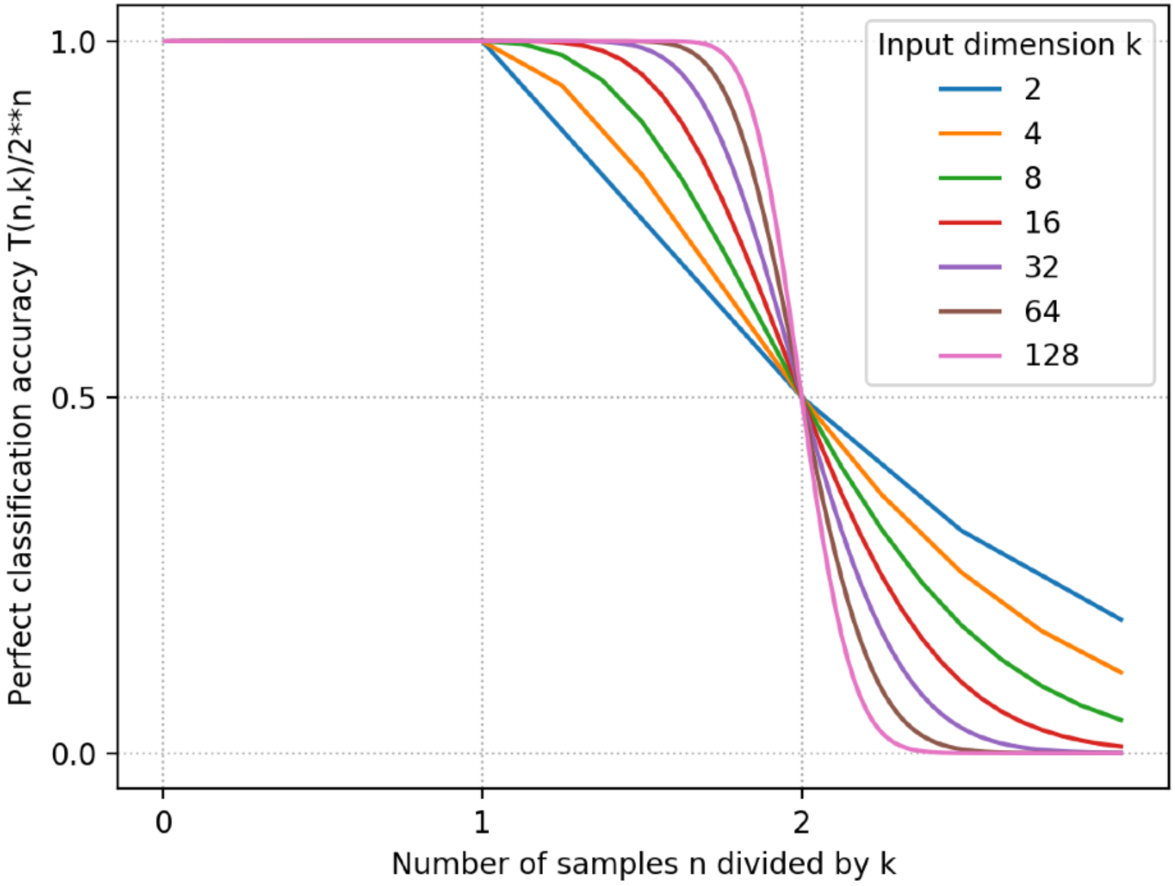
Examples: How many bits of maximal capacity?



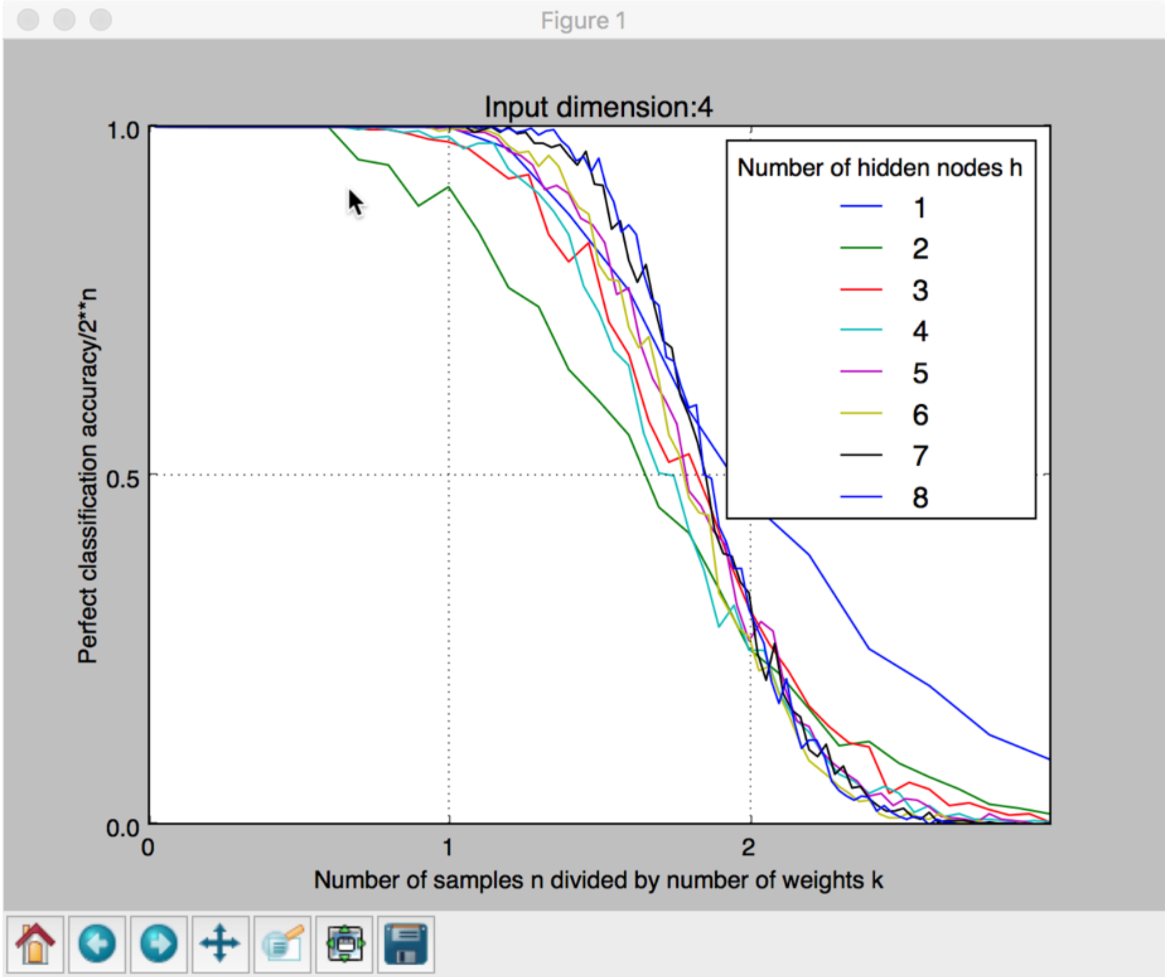
3 bits+4 bits=7 bits

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

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

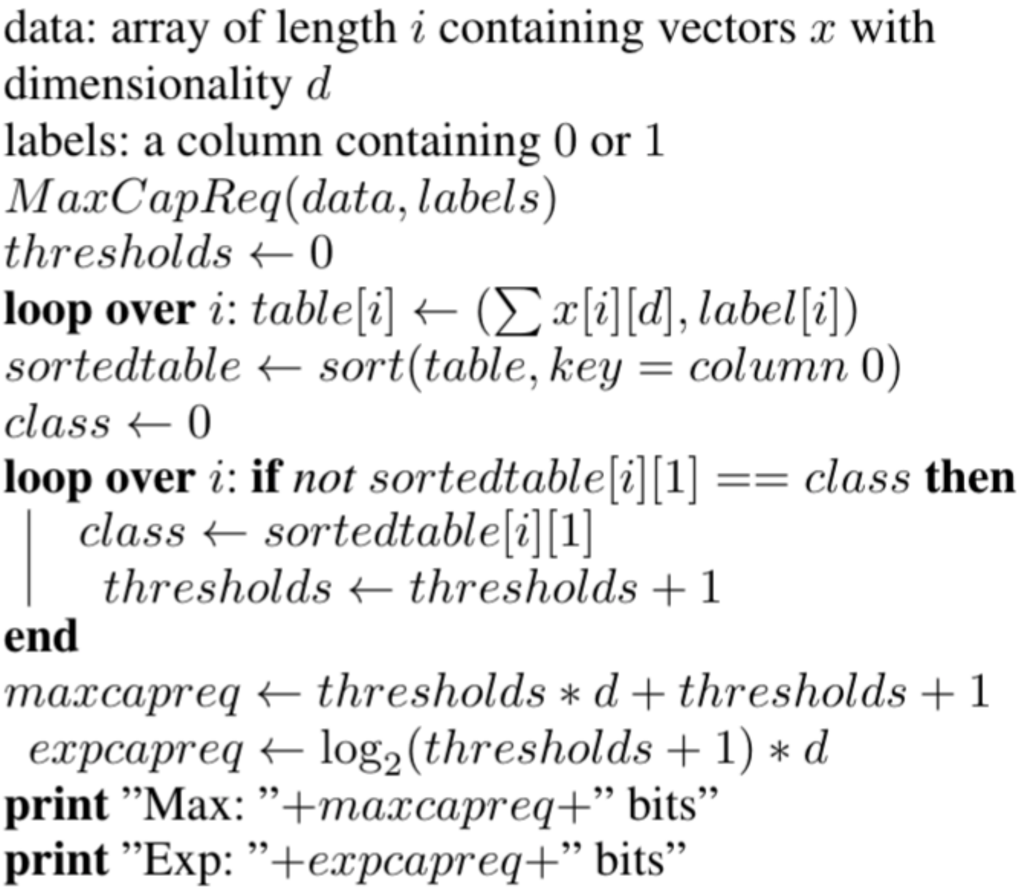
Predicting Capacity Requirements

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

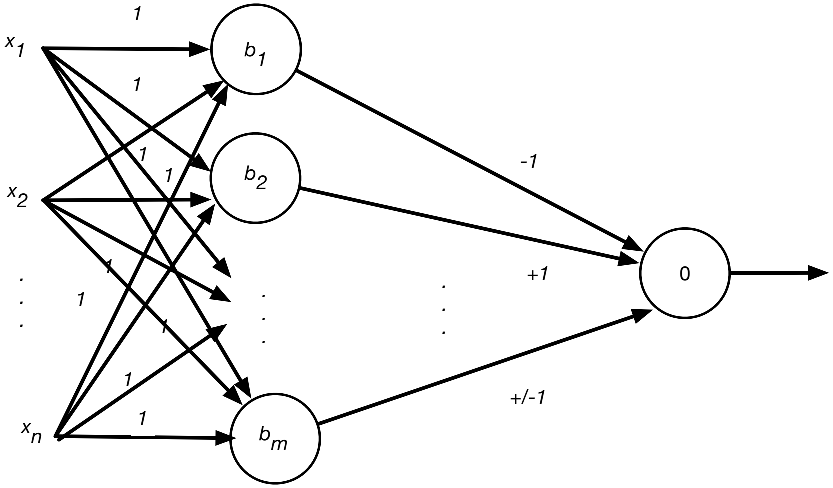
Idea:

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)*

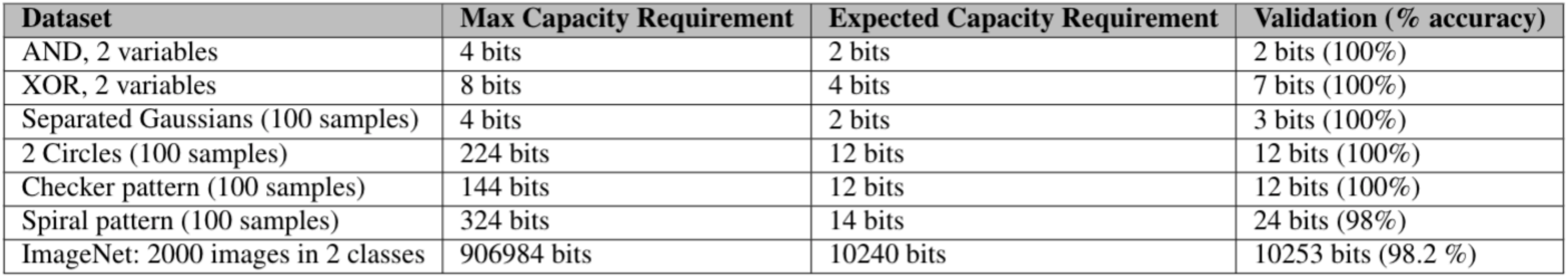
Predicting Expected Minimum Memory Equivalent Capacity

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 >1bit per parameter (Cover, MacKay).

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

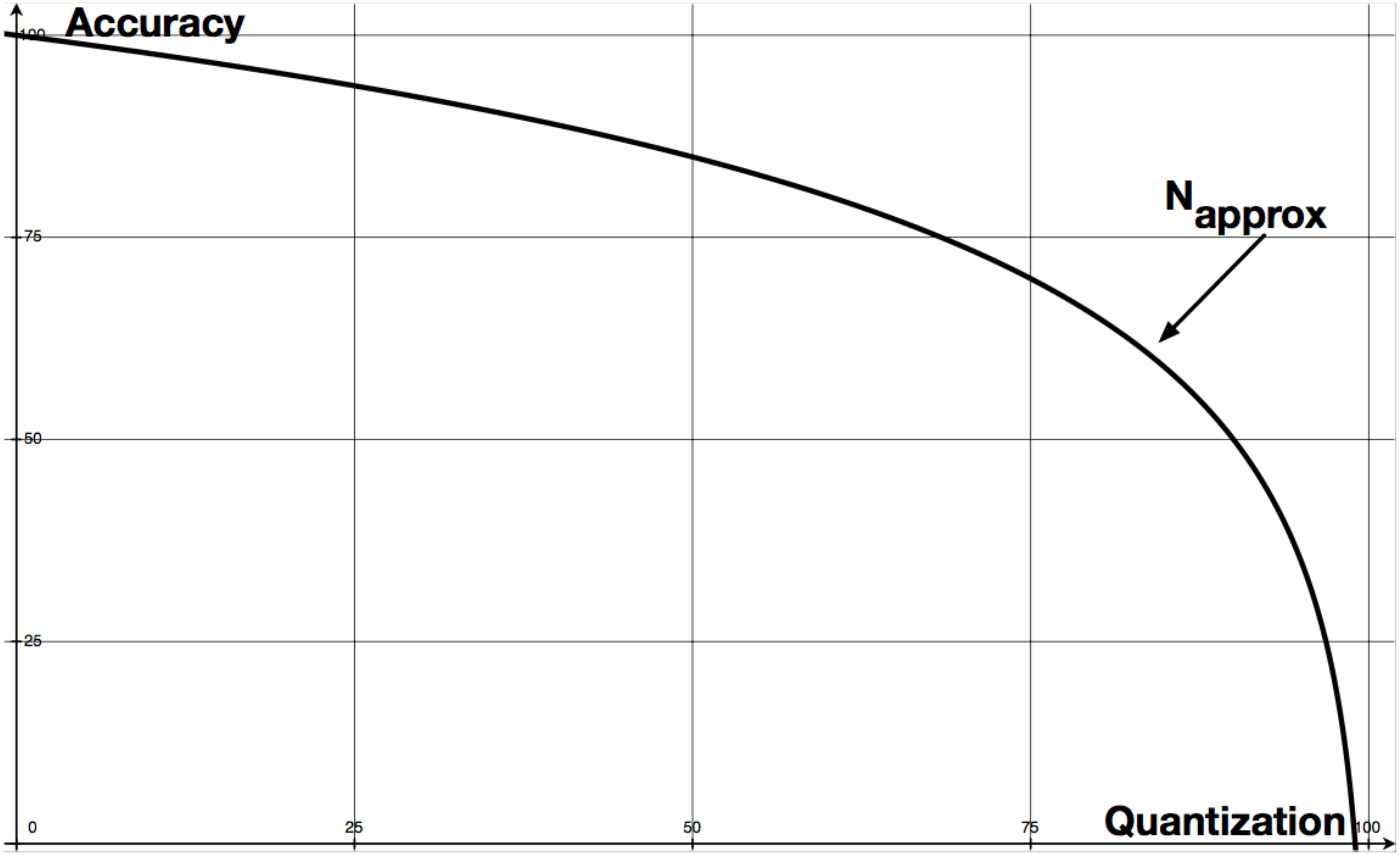
Suggested Engineering Process for Generalization

- 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):

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

- Multi-class/regression:

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

Non-Statistical Definition (Literature)

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

$x \in$ Training data

$x' \in$ Test data

$| \circ |$ semi-metric

f machine learner

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

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

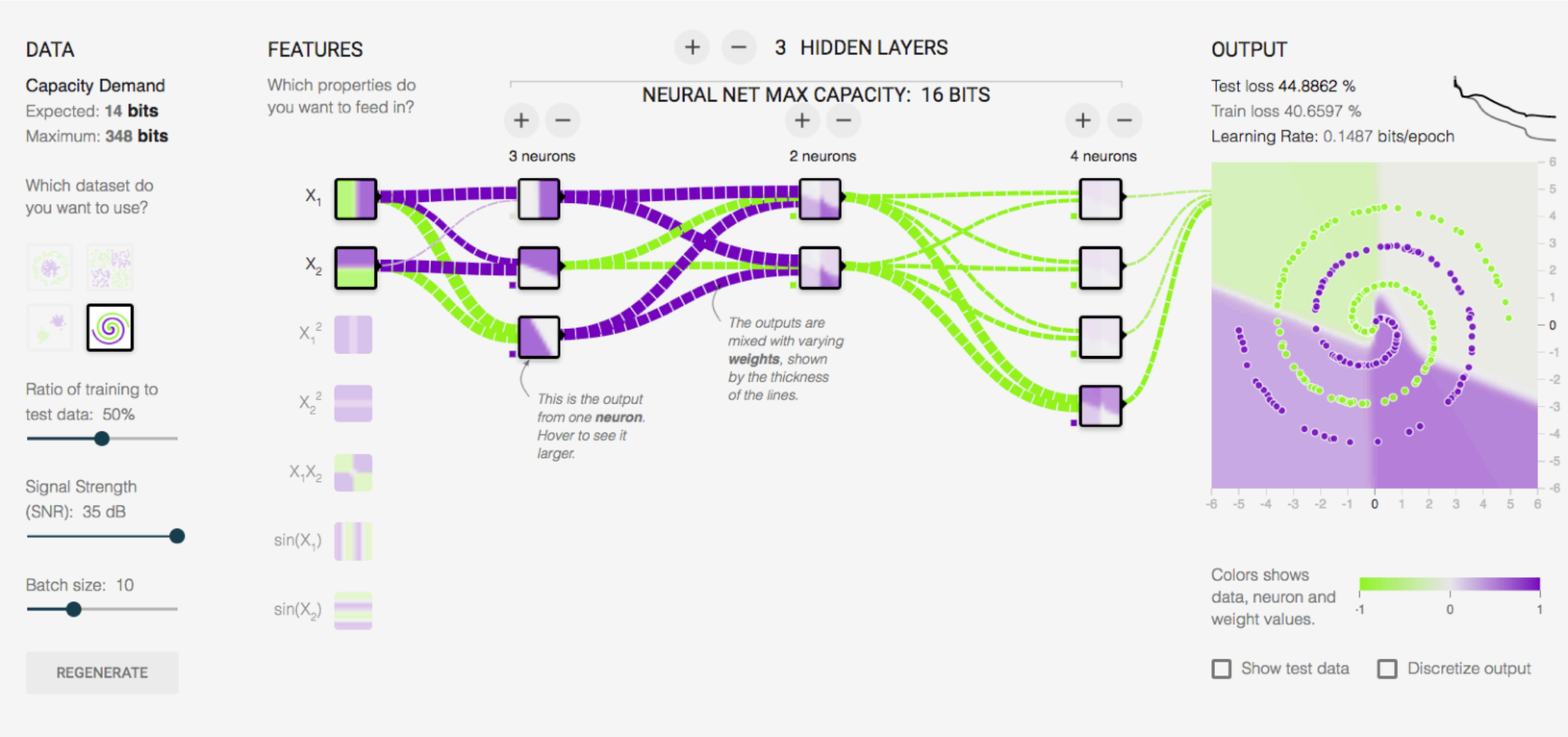
Statistical equivalent: How many bits per bit can be ignored on average (see *G* measure).

Central Hypothesis

Given perfect training, capacity makes a machine learning experiment reproducible independent of other hyperparameters.

- This is, initialization, learning rate, batch size, and architecture do not matter!
- TODO: Improve training!

Demo: Experimental Design for TensorFlow



DATA

Capacity Demand
Expected: **14 bits**
Maximum: **348 bits**

Which dataset do you want to use?

Ratio of training to test data: 50%

Signal Strength (SNR): 35 dB

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

3 HIDDEN LAYERS

NEURAL NET MAX CAPACITY: 16 BITS

3 neurons

2 neurons

4 neurons

*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

Test loss 44.8862 %
Train loss 40.6597 %
Learning Rate: 0.1487 bits/epoch

Colors shows data, neuron and weight values.

□ Show test data

□ Discretize output

http://tfmeter.icsi.berkeley.edu

Exercises (joint)

Please go to

https://tinyurl.com/y3xn8dnf

and work on the homework 1, 2, 3, 4, 5, and 6 in groups.

We will discuss them in about 45 min.