Measuring Generalization in Machine Learning

CASIS, 2019

Gerald Friedland (ENG, UC Berkeley)
with input from Berkeley Institute of Data Science
"Uncertainty and Information" discussion group

May 15, 2019

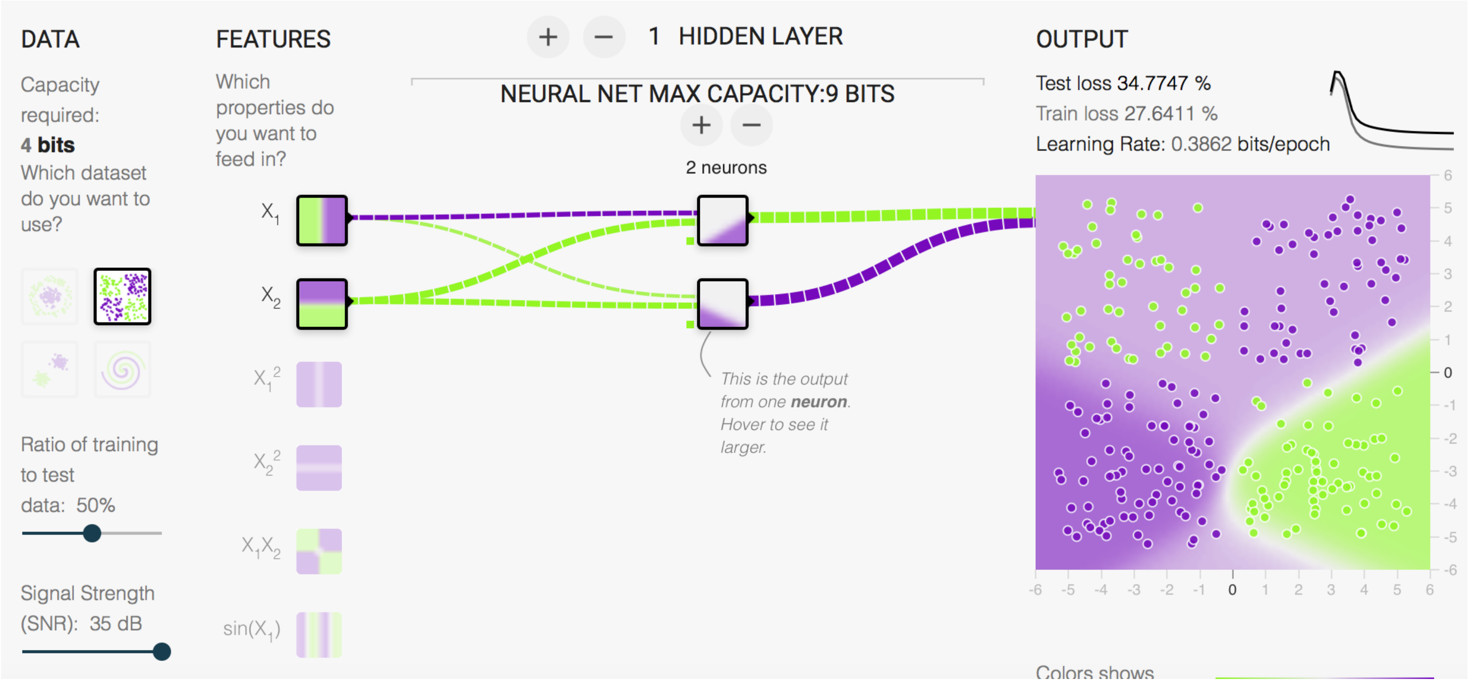
LLNL-PRES-774205

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Background

- SIAM CSE: On the Capacity of Neural Networks
  See: tfmeter.icsi.berkeley.edu



Information Theory and Signal Processing make Machine Learning (ML) more efficient!

Thought Framework: Machine Learning

- Intelligence: *The ability to adapt* (Binet and Simon, 1904)
- Machine learning *adapts a state machine 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.

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

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

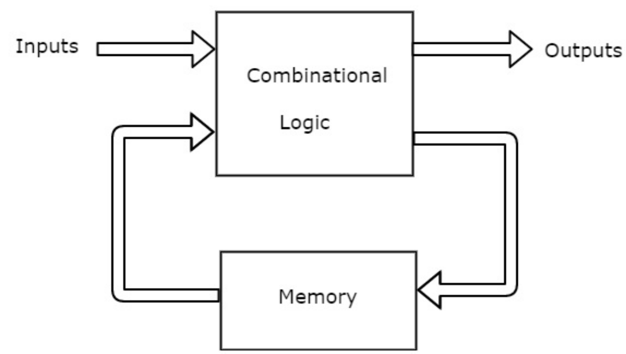
Thought Framework: Machine Learning

- 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

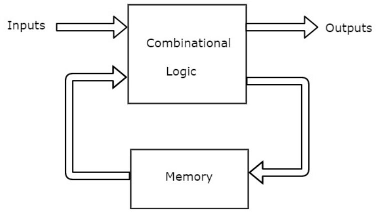
Assume

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

(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?

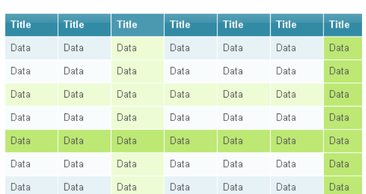
Maximally: #rows (lookup table)
Minimally: ? (Shannon Entropy of significant digits)

Thought Framework: Machine Learning

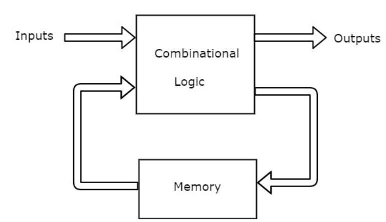
- ***Intellectual Capacity:*** *The number of unique target functions a machine learner is able to represent (as a function of the number of model parameters).*

- ***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



Generalization in Machine Learning

Memorization is worst-case generalization.

For binary classifiers:

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

G<1 => *M* needs more training/data
G=1 => *M* is memorizing = overfitting
G>1 => *M* is generalizing

Generalization in Machine Learning

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

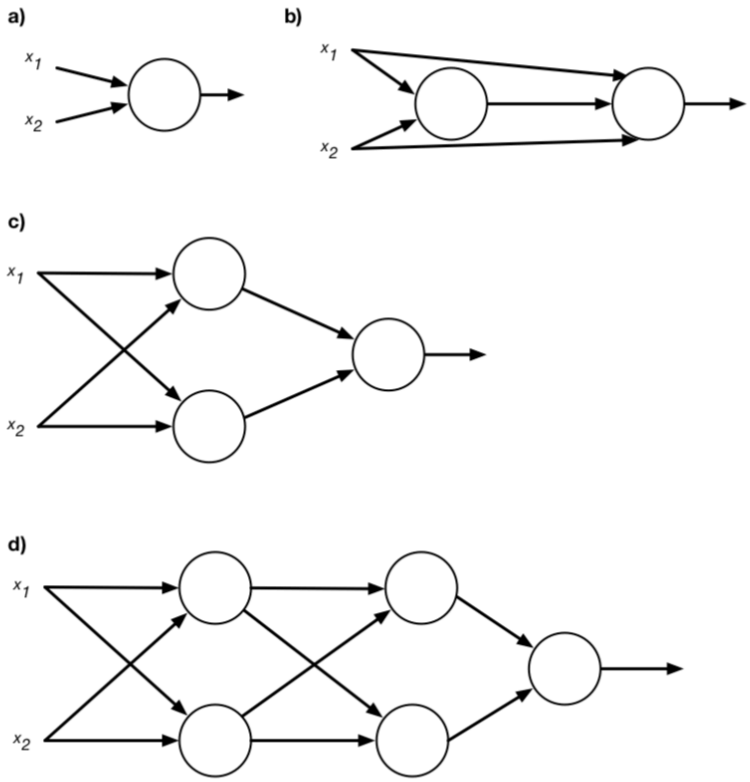
Advantages of this definition:

- 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 (see next slide)
- Random Forrest: TBD
- SVN: TBD
- k-NN: TBD
- GMMs: TBD

Memory Equivalent Capacity for NNs is like Circuit Analysis



1. The output of a single perceptron yields maximally one bit of information.
2. The capacity of a single perceptron is the number of its parameters (weights and bias) in bits.
3. The total capacity C_{tot} of M perceptrons in parallel is $C_{\text{tot}} = \sum_{i=1}^{M} C_i$ where C_i is the capacity of each neuron.
4. For perceptrons in series (e.g., in subsequent layers), the capacity of a subsequent layer cannot be larger than the output of the previous layer.

a) 3bits, b) RESNET: 3bits+4bits=7bits, c) 2*3bits+3bits=9bits
d) 2*3+max(2*3,2+2)+max(3,2+1)=6+4+3=13bits

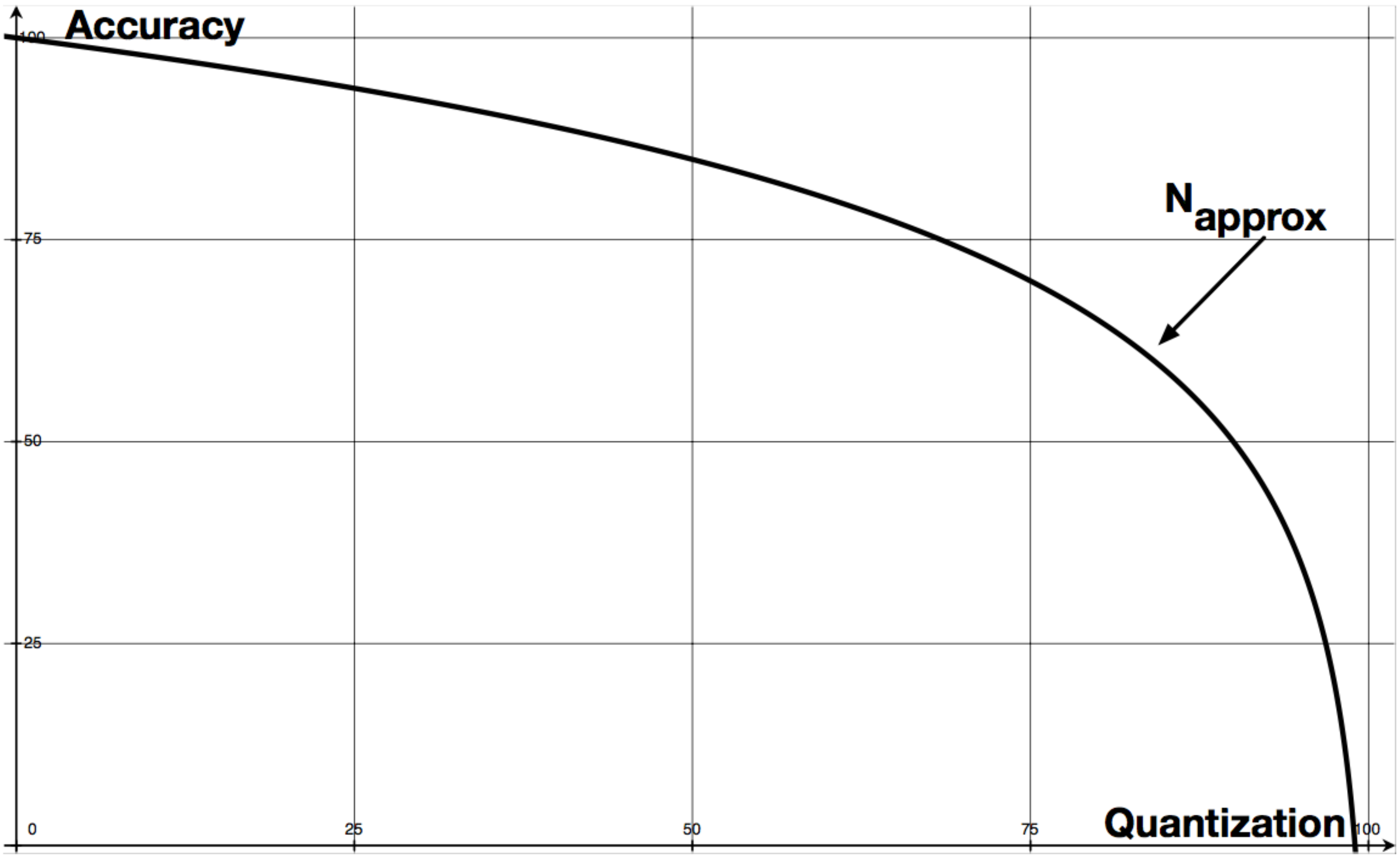
Generalization for Regression

- Assume an n-row table with header:
- Memorization is worst-case generalization

$$G = \frac{\#correctly\ predicted\ rows}{\#rows\ that\ can\ be\ memorized}$$

G<1 => *M* needs more training/data
G=1 => *M* is memorizing = overfitting
G>1 => *M* is generalizing

Process: Reduce MEC of Machine Learner while Training



Conclusion

- Information definition of generalization for Machine Learning that uses less assumptions and is therefore easier to implement.
- Creates an engineering process. Start at MEC=#instances!
- Allows comparisons of approaches beyond accuracy.
- Provides and understanding of data/training needs.
- Smallest MEC, highest accuracy = best machine learner. (Occam's Razor)

Future Work

- MEC for various other classifiers and tasks:
  - SVN, Random Forrests, GMMs, k-nn?
  - Impact of regularization?
  - Impact of imperfect training?
  - Regression, generative modeling
- Tools, tools, tools.

Thank You!

Questions?