The Mathematical Foundations of Artificial Intelligence

W.D. Rawle, PhD
Senior Member IEEE
Chair, IEEE Maine Section
September 16, 2020

Deep Conversations on Deep Learning

A technical series hosted by IEEE Maine Section

IEEE
Advancing Technology for Humanity

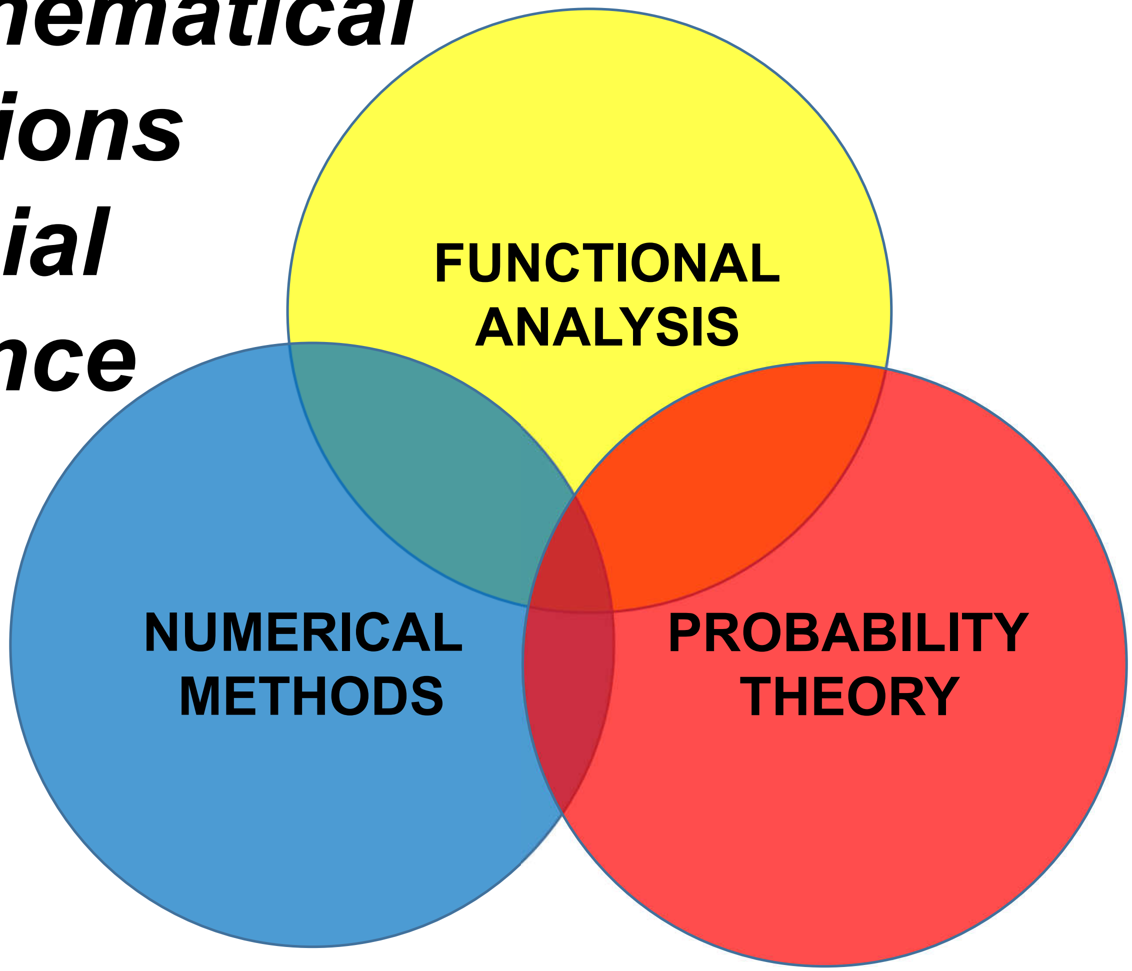
Artificial Intelligence

In computer science, artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans. Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

..

Russell, Stuart J.; Norvig, Peter (2009). Artificial Intelligence: A Modern Approach (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 978-0-13-604259-4

The Mathematical Foundations of Artificial Intelligence



The Mathematical Foundations Of Artificial Intelligence

- ***Functional Analysis:***
  - ***Establishes to domain of the model or "hypothesis"***
  - ***Defines operations within the domain and transformations into adjacent domains***
  - ***Provides for measures of completeness: orthonormal function sets, vector projection***
  - ***Simplifies to more tractable implementations: linear algebra, matrix arithmetic, Fourier series.***

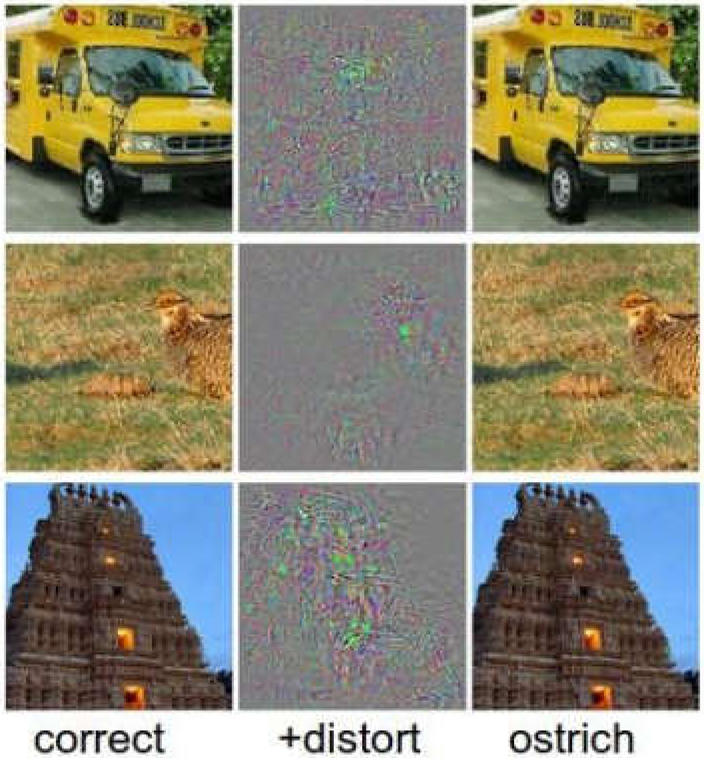
The Mathematical Foundations Of Artificial Intelligence

- ***Numerical Methods***
  - ***Solutions to multivariate classification problems often require optimization routines:***
    1. ***Establishment of cost and gradient functions***
    2. ***Numerical search strategies***
    3. ***Linearization/determinism of stochastic process***
    4. ***Application of heuristics and ontologies***
    5. ***Numerical integration and differentiation required for ill defined data or "complicated" regions***

The Mathematical Foundations Of Artificial Intelligence

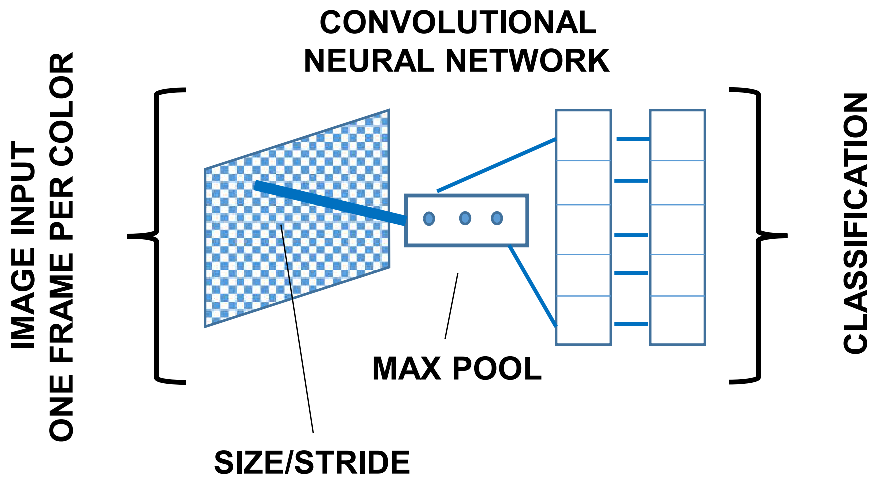
- ***Probability Theory***
  - ***Establishes performance bounds upon stochastic classifiers: Bayesian networks, Particle Filters, Markov Chains, Maximum Likelihood, Parameter Estimation, Statistical Analysis of Physical Parameters***
  - ***Accommodates stochastic processes and multi-variate data - employing measures such as Mahalanobis Distance and Mahalanobis-Bregman divergence***

CNNs and Functional Analysis



Fooled by a little distortion

MIT 6.S094 Lex Friedman Deep Learning Lecture 1



CNN Approximations

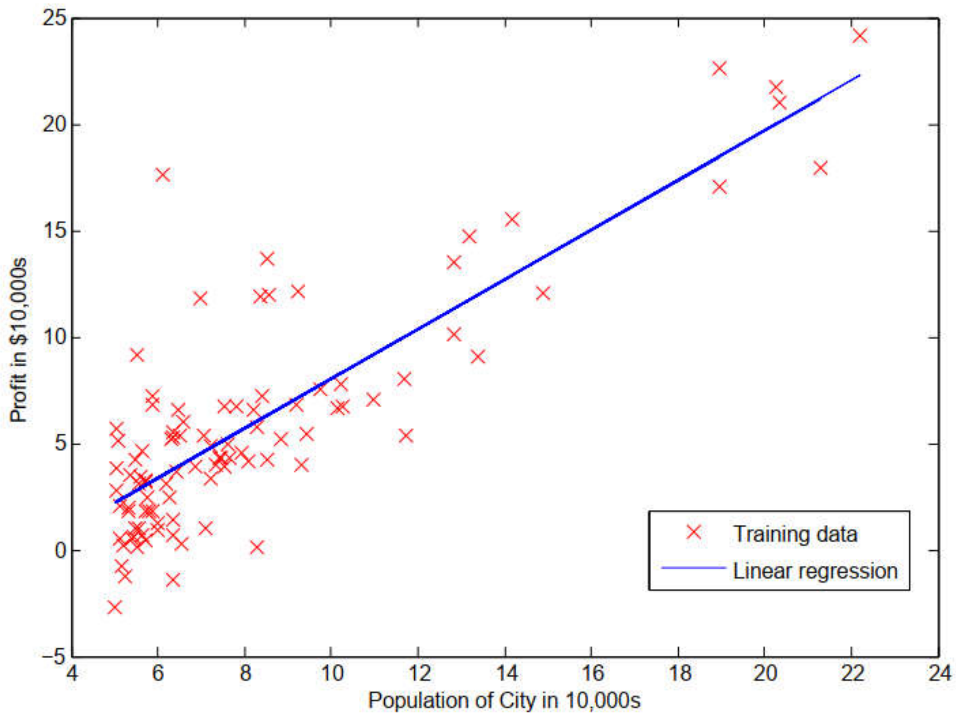
- **Size/stride- convolution approx.**
- **Max Pool – data loss**
- **Training – insufficient samples**

CNNs and Functional Analysis

Questions??

- **Is the image sufficiently sampled to capture "high frequency" effects- Nyquist criteria**
- **Does the discretization of the convolution function compromise the output**
- **How much data is lost when using max pool compression**
- **Is fidelity of training data sufficient**
- **Would alternate approaches (DCT, for example) provide sufficient compression and maintain fidelity**
- **What would be the difference in compute resource requirements**

NNs and Numerical Methods



Linear Regression

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Cost function

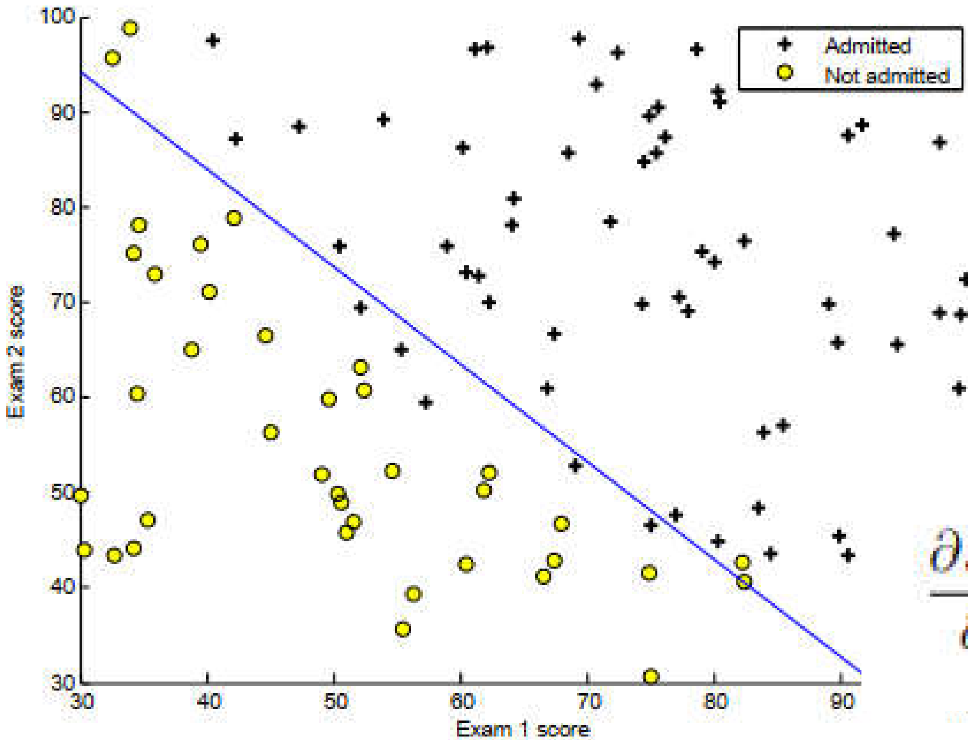
$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

hypothesis

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Numerical "gradient"

NNs and Numerical Methods



Logistic Regression

$$h_\theta(x) = g(\theta^T x),$$

Activated hypothesis

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Non linear activation

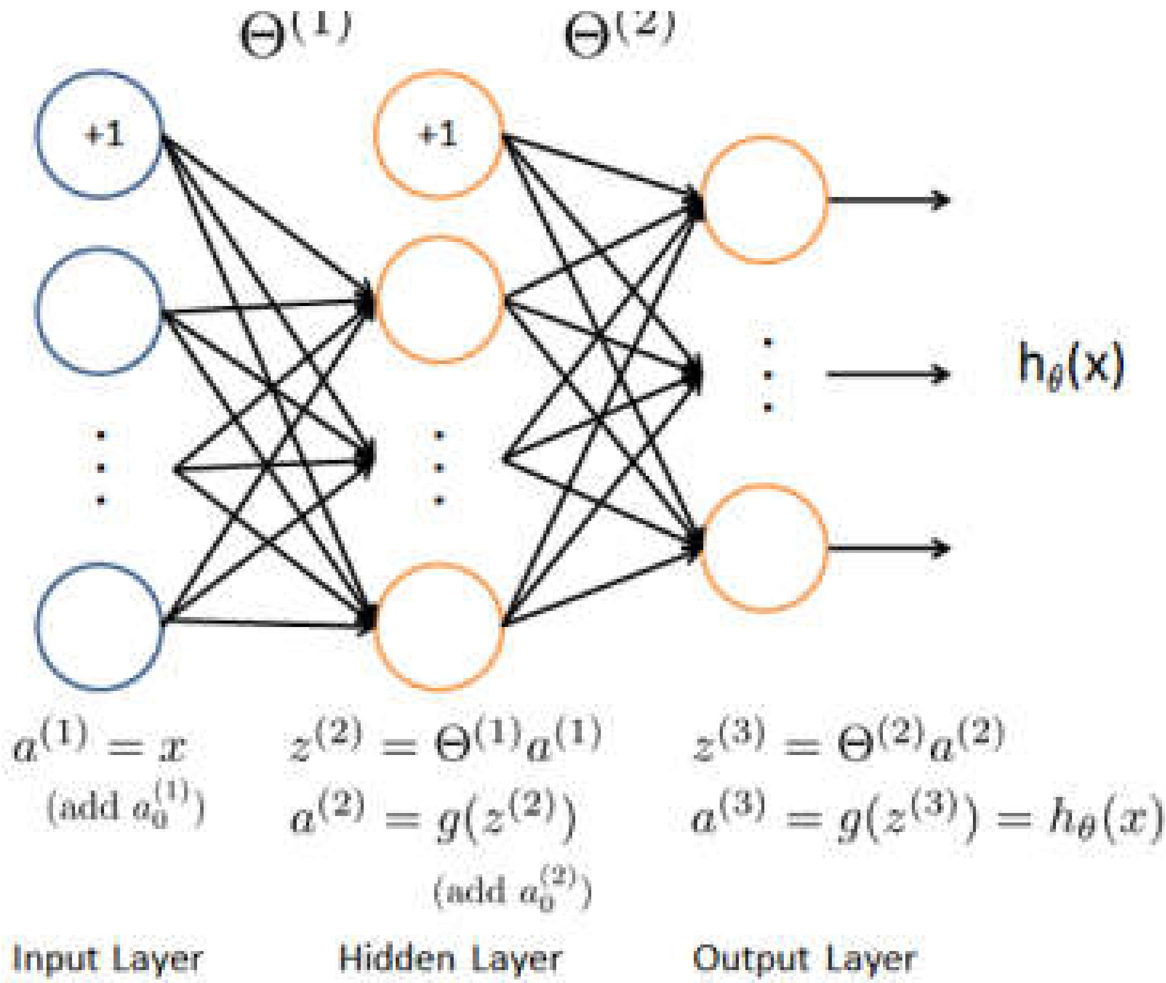
$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Parameter Gradient

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right],$$

Cost function

NNs and Numerical Methods



IEEE

NNs and Numerical Methods

Forward Propagation

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\theta(x^{(i)}))_k)\right] +$$

$$\frac{\lambda}{2m}\left[\sum_{j=1}^{25}\sum_{k=1}^{400}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2\right].$$

Back Propagation, minimize wrt θ, gradient derivatives

$\delta_j^{(l)}$ = "error" of node j in layer l.

Given

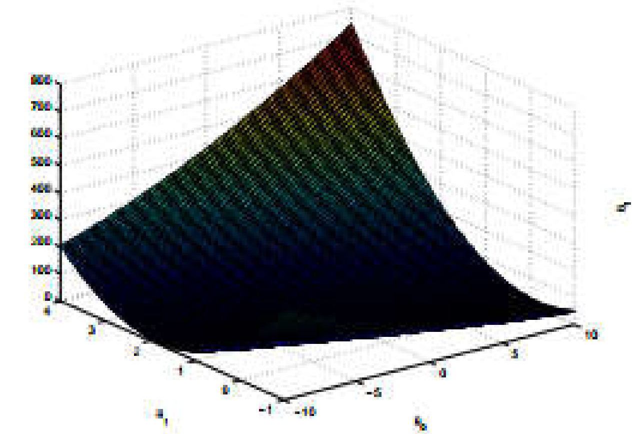
$$\delta^{(3)} = (\Theta^{(3)})^T\delta^{(4)} .* g'(z^{(3)})$$

$$a^{(4)} = h_\Theta(x) = g(z^{(4)})$$

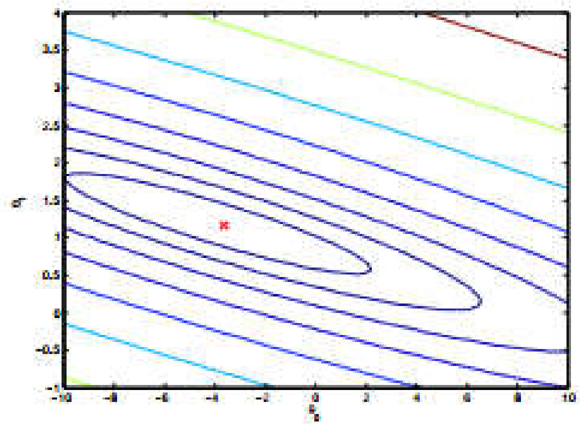
$$\delta^{(2)} = (\Theta^{(2)})^T\delta^{(3)} .* g'(z^{(2)})$$

NNs and Numerical Methods

Searching for the minima
- Classic optimization theory
  - Conjugate gradient
  - Simplex
  - Direct search
  - Stochastic Gradient



(a) Surface

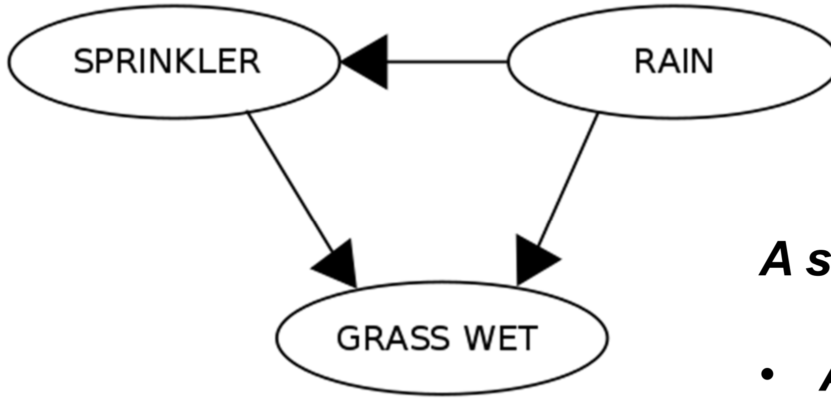


(b) Contour, showing minimum

Challenges
- Well behaved – global minima
- Oscillatory behavior
- Regularization
- Convergence Rate

BBNs and Probability Theory

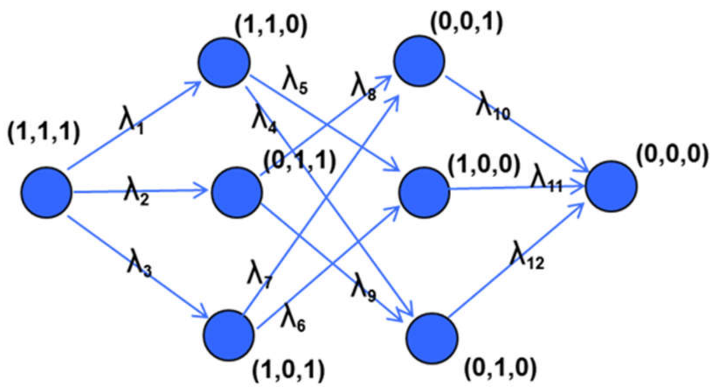
	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

A simple example from Wikipedia

- *Acyclic graph*



Markov Model

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

BBNs and Probability Theory

Given $p(C_k \mid x_1, \ldots, x_n)$ $\qquad p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$

$$\boxed{\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}}$$

$$\begin{aligned} p(C_k, x_1, \ldots, x_n) &= p(x_1, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\, p(x_2, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\, p(x_2 \mid x_3, \ldots, x_n, C_k)\, p(x_3, \ldots, x_n, C_k) \\ &= \ldots \end{aligned}$$

BBNs and Probability Theory

Naïve Bayes Probability Condition

$$p(x_i \mid x_{i+1}, \ldots, x_n, C_k) = p(x_i \mid C_k)\,.$$

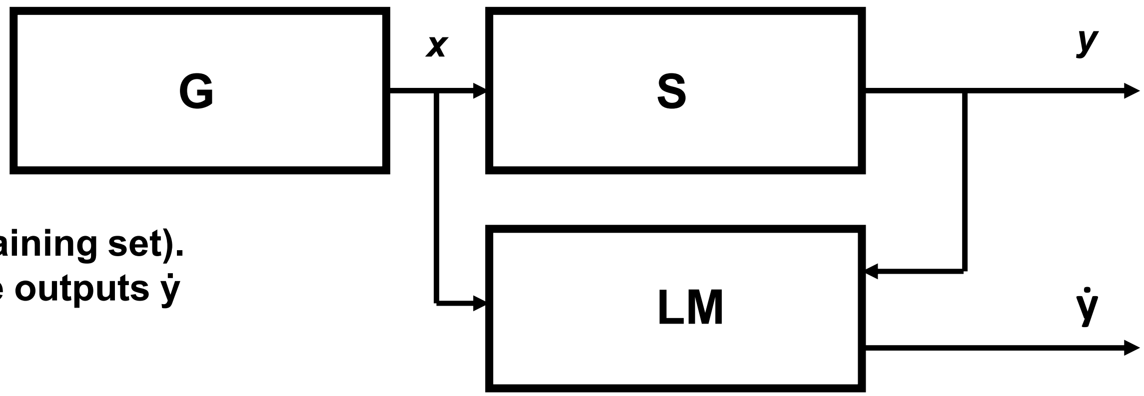
$$\begin{aligned} p(C_k \mid x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\ &= p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k)\, \cdots \\ &= p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,, \end{aligned}$$

Constructing a classifier from the probability model

$$\boxed{\hat{y} = \underset{k \in \{1, \ldots, K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).}$$

Vapnik's Learning Model

1. ***A generator of random vectors*** $x \in R^n$, ***drawn independently from a fixed but unknown probability distribution function*** $F(x)$.
2. ***A supervisor who returns an output value*** y ***to every input vector*** x ***according to a conditional distribution function*** $F(y|x)$ ***also fixed but unknown***
3. ***A learning machine capable of implementing a set of functions*** $f(x, \alpha), \alpha \in \Lambda$, ***where*** Λ ***is a set of parameters.***

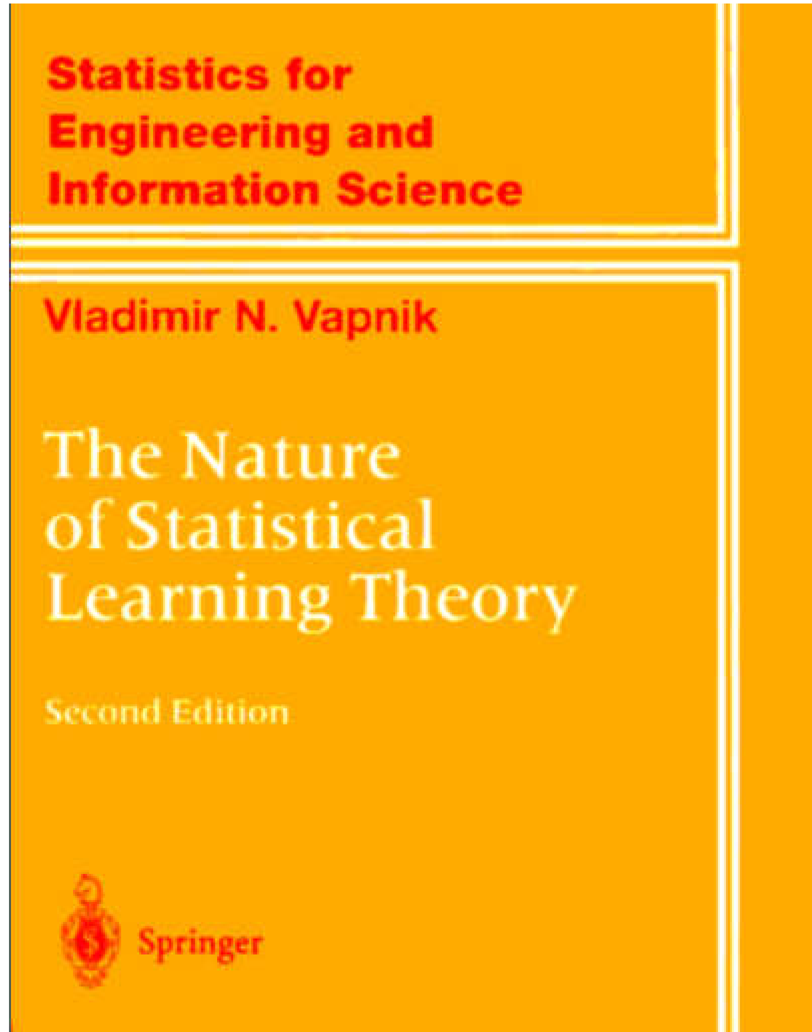


LM observes pairs (x,y) (training set). Subsequently, the machine outputs ẏ for any given x.

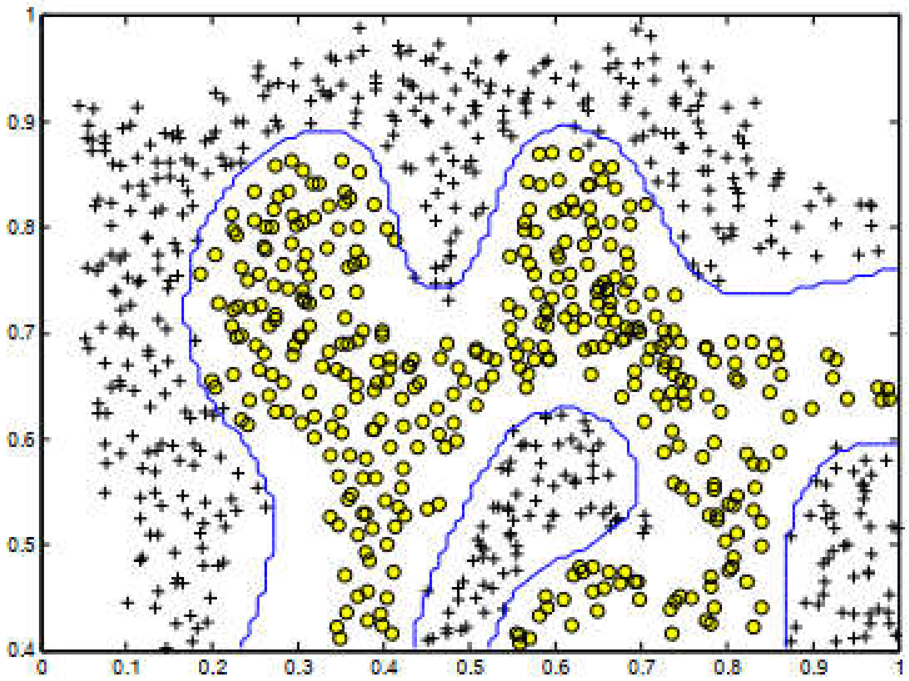
The Nature of Statistical Learning

- ***What are (necessary and sufficient) conditions for consistency of a learning process based on the ERM principle?***
- ***How fast is the rate of convergence of the learning process?***
- ***How can one control the rate of convergence (the generalization ability) of the learning process***
- ***How can one construct algorithms that can control the generalization ability?***

Empirical Risk Minimization Inductive Principle



The Nature of Statistical Learning



Support Vector Machines

SVM Kernels

- ***homogeneous polynomial,***
- ***inhomogeneous polynomial,***
- ***Gaussian radical basis,***
- ***hyperbolic tangent***

$$K_{gaussian}(x^{(i)}, x^{(j)}) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{k=1}^{n}(x_k^{(i)} - x_k^{(j)})^2}{2\sigma^2}\right).$$

Thank you

Many thanks to Dr. Ali Abedi, NE Area Chair, IEEE Region 1 for providing Webex Resources

W.D Rawle PhD
Senior Member, IEEE
Chair, IEEE Maine Section
wrawle@hotmail.com