EENG582

Artificial Neural Networks

Learning Process

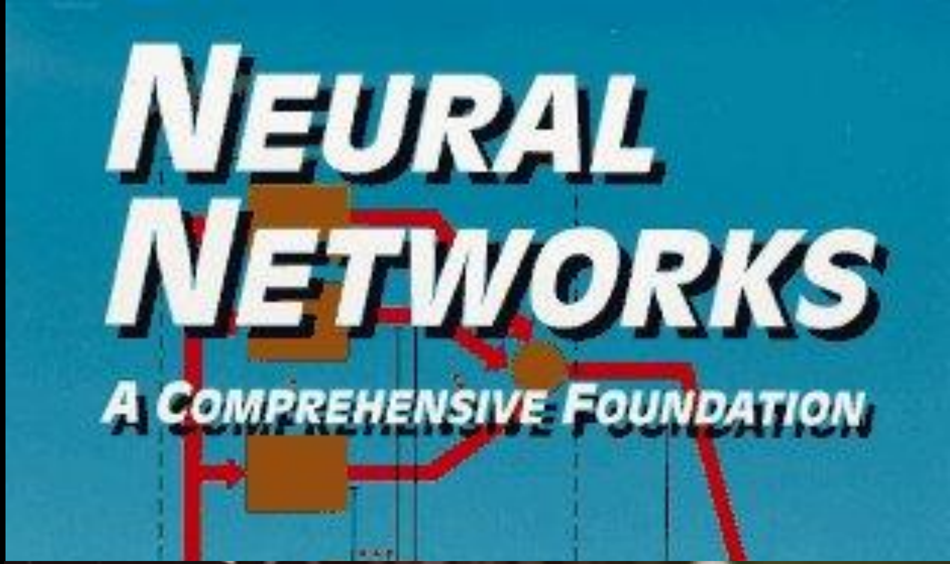
Sections 2.13 – 2.16

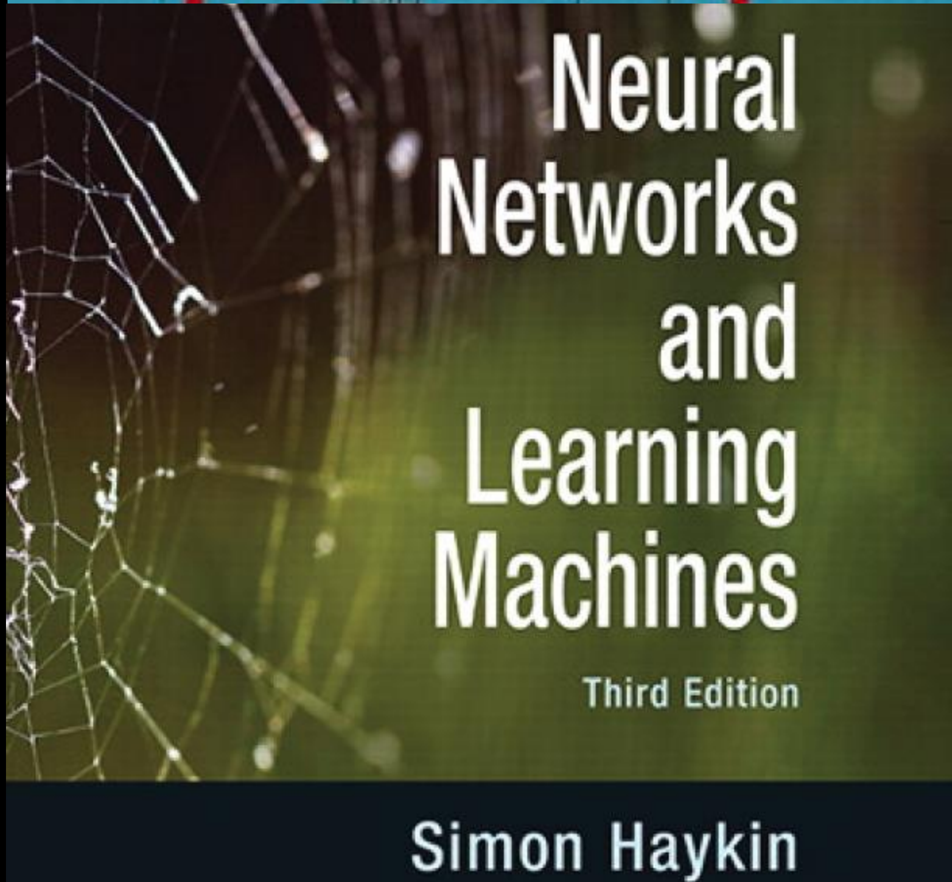
Artificial Neural Networks

Prof. Dr. Hasan AMCA

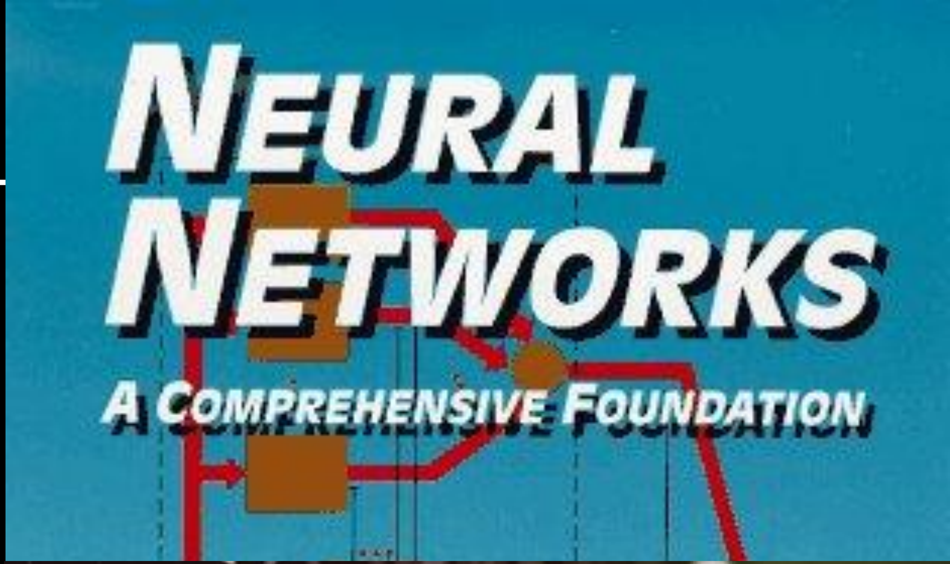
Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

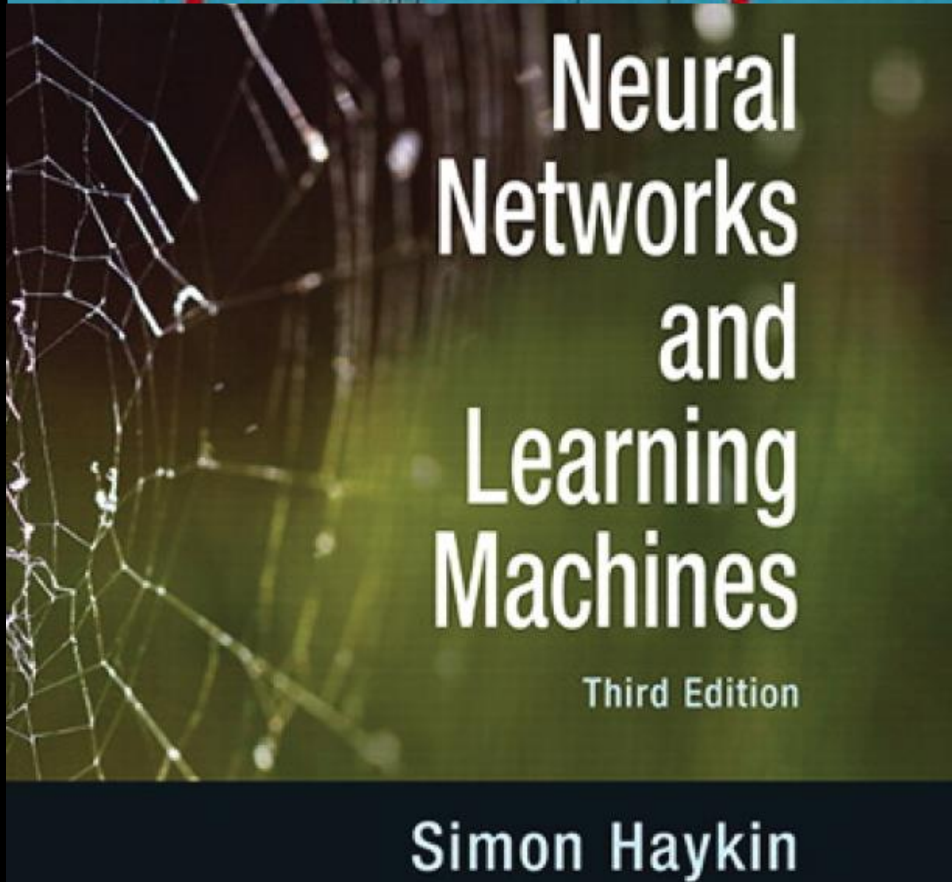
Eastern Mediterranean University
(emu.edu.tr)





2 Learning Process





2.14 Statistical Learning Theory

- A model of supervised learning consists of three interrelated components, illustrated in Fig. 2.22 and abstracted in mathematical terms as follows.

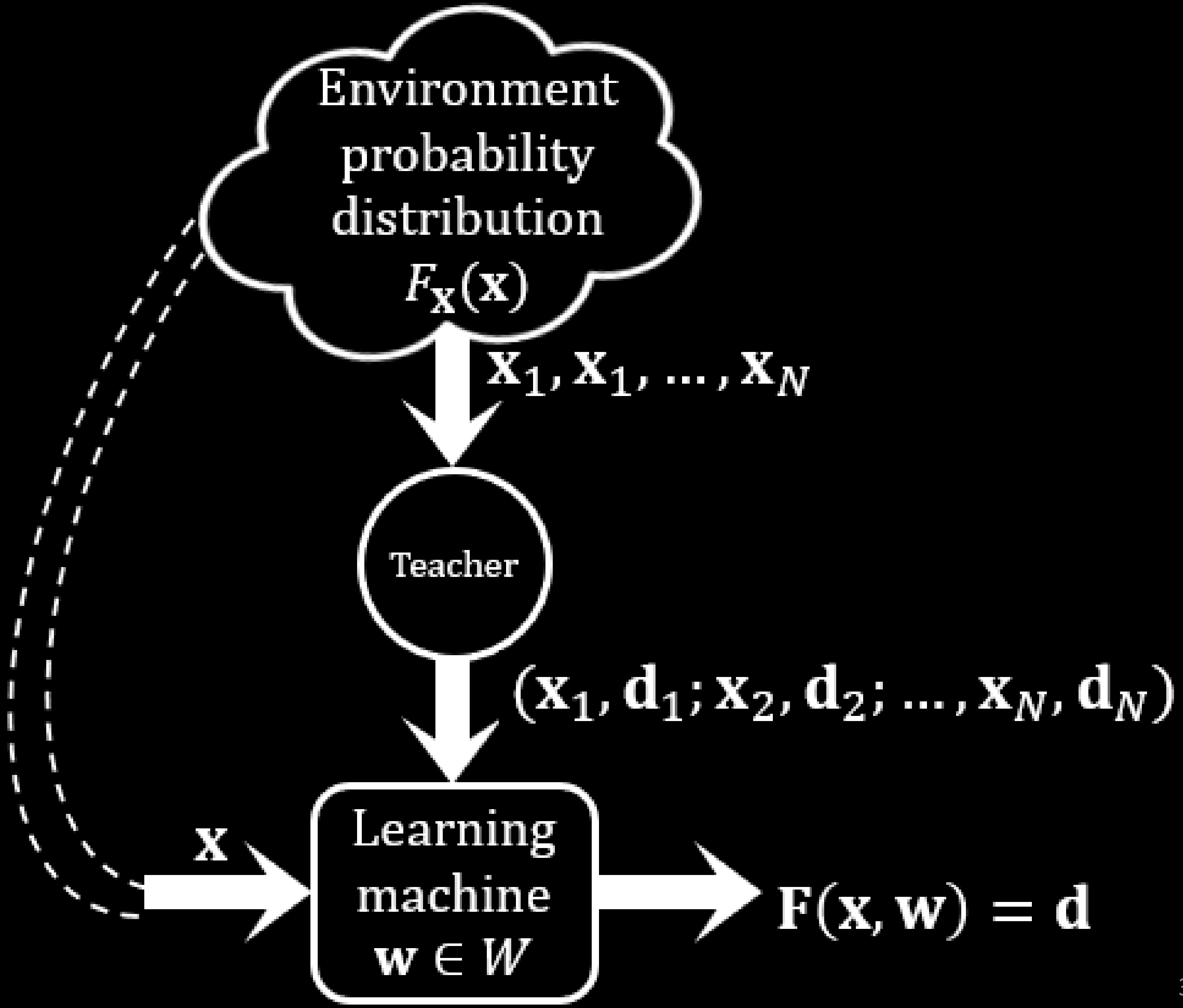


Fig. 2.22 Model of the supervised learning process.

2.14 Statistical Learning Theory

1. Environment. The environment is stationary, supplying a vector $\mathbf{x}$ with a fixed but unknown cumulative (probability) distribution function $F_{\mathbf{x}}(\mathbf{x})$.

2. Teacher. The teacher provides a desired response d for every input vector $\mathbf{x}$ received from the environment, in accordance with a conditional cumulative distribution function $F_{\mathbf{x}}(\mathbf{x}|d)$ that is also fixed but unknown. The desired response d and input vector $\mathbf{x}$ are related by

$$d = f(\mathbf{x}, v) \qquad (2.69)$$

where v is a noise term, permitting the teacher to be "noisy."

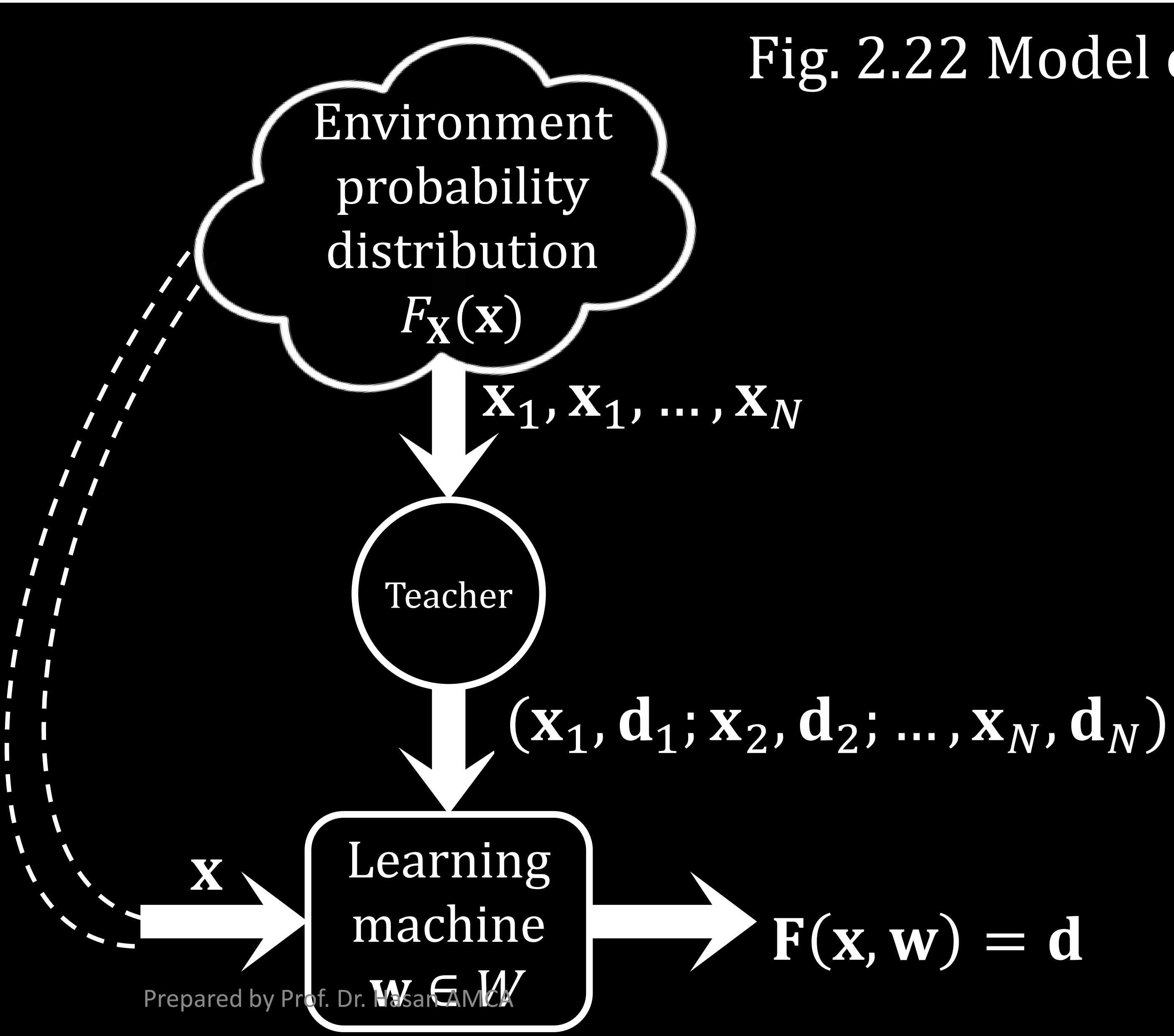
3. Learning machine (algorithm). The learning machine (neural network) is capable of implementing a set of input-output mapping functions described by

$$y = F(\mathbf{x}, \mathbf{w}) \qquad (2.70)$$

where y is the actual response produced by the learning machine in response to the input $\mathbf{x}$, and $\mathbf{w}$ is a set of free parameters (synaptic weights) selected from the parameter (weight) space W.

2.14 Statistical Learning Theory

Fig. 2.22 Model of the supervised learning process.



2.14 Statistical Learning Theory

- The supervised learning problem is that of selecting the particular function $F(\mathbf{x}, \mathbf{w})$ that approximates the desired response d in a statistically optimum fashion.

- The selection itself is based on the set of N independent, identically distributed (iid) training examples described as

$$\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$$

- The feasibility of supervised learning depends on this question:
  - Do the training examples $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ contain sufficient information to construct a learning machine capable of good generalization performance?

2.14 Statistical Learning Theory

- Let $L\big(d, F(\mathbf{x}, \mathbf{w})\big)$ denote a measure of the loss or discrepancy between the desired response d corresponding to an input vector $\mathbf{x}$ and the actual response $F(\mathbf{x}, \mathbf{w})$ produced by the learning machine.
- A popular definition for the loss function defined as the squared distance between $d = f(\mathbf{x})$ and the approximation $F(\mathbf{x}, \mathbf{w})$ as shown by:

$$L\big(d, F(\mathbf{x}, \mathbf{w})\big) = \big(d - F(\mathbf{x}, \mathbf{w})\big)^2 \quad (2.71)$$

- which is the ensemble-average performed over all the example pairs ($\mathbf{x}$, d).
- The expected value of the loss is defined by the risk functional

$$R(w) = \int L\big(d, F(\mathbf{x}, \mathbf{w})\big) dF_{\mathbf{x},D}(\mathbf{x}, d) \quad (2.72)$$

where the integral is taken over all possible values of the example pair ($\mathbf{x}$, d).

2.14 Statistical Learning Theory

- The goal of supervised learning is to minimize the risk functional $R(\mathbf{w})$ over the class of approximating functions $\{F(\mathbf{x}, \mathbf{w}), \mathbf{w} \in \mathrm{W}\}$.
- In supervised learning, the only information available is contained in the training data set $\mathcal{T}$.
- To overcome the mathematical difficulties in complications in evaluating the risk function,
  - empirical risk minimization principle that relies entirely on availability of the training data set $\mathcal{T}$, can be used.

Definitions: Convergence in Probability

- Consider a sequence of random variables $a_1, a_2, \dots, a_N$.
- This sequence of random variables is said to converge in probability to a random variable a_0 if for any $\delta > 0$, the probabilistic relation

$$P(|a_N - a_0|) > \delta \xrightarrow{P} 0 \qquad as\ N \to \infty \tag{2.73}$$

holds

Definitions: Supremum and infimum

- The supremum of a nonempty set $\mathcal{A}$ of scalars, denoted by $\sup \mathcal{A}$, is defined as the smallest scalar x such that $x \geq y$ for all $y \in \mathcal{A}$.
- If no such scalar exists, we say that the supremum of the non-empty set $\mathcal{A}$ is ∞.
- Similarly, the infimum of set $\mathcal{A}$, denoted by $\inf \mathcal{A}$, is defined as the largest scalar x such that $x \leq y$ for all $y \in \mathcal{A}$.
- If no such scalar exists, we say that the infimum of the nonempty set $\mathcal{A}$ is ∞.

http://mathonline.wikidot.com/proofs-regarding-the-supremum-or-infimum-of-a-bounded-set

Definitions: Empirical Risk Functional

- Given the training sample $\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, the empirical risk functional is defined in terms of the loss function Let $L\big(d_i, F(\mathbf{x}_i, \mathbf{w})\big)$ as

$$R_{emp}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\big(d_i, F(\mathbf{x}_i, \mathbf{w})\big) \tag{2.74}$$

Definitions: Strict Consistency

- Consider the set $\mathcal{W}$ of functions $L\big(d, F(\mathbf{x}, \mathbf{w})\big)$ whose underlying distribution is defined by the joint cumulative distribution function $F_{\mathbf{x},D}(\mathbf{x}, d)$.

- Let $\mathcal{W}(c)$ be any non-empty subset of this set of functions, such that

$$\mathcal{W}(c) = \left\{\mathbf{w}: \int L\big(d, F(\mathbf{x}, \mathbf{w})\big) \geq c\right\} \qquad (2.75)$$

where $c \in (-\infty, \infty)$. The empirical risk functional is said to be strictly (nontrivially) consistent if for any subset $\mathcal{W}(c)$ the following convergence in probability

$$\inf_{\mathbf{w}\in\mathcal{W}(c)} R_{emp}(w) \xrightarrow{P} \inf_{\mathbf{w}\in\mathcal{W}(c)} R(w) \quad \text{as } N \to \infty \qquad (2.76)$$

holds.

Definitions: Law of Large Numbers

- According to the law of large numbers, which constitutes one of the main theorems of probability theory,
  - as the size N of the training sample $\mathcal{T}$ is made infinitely large, the empirical mean of the random variable $Z_{\mathbf{w}^*}$ converges to its expected value.

Definitions: Principle of Empirical Risk Minimization

- The basic idea of the principle of empirical risk minimization is to work with the empirical risk functional $R_{emp}(\mathbf{w})$ defined in Eq. (2.74).
- This new functional differs from the risk functional $R(\mathbf{w})$ of Eq. (2.72) in two desirable ways:

1. It does not depend on the unknown distribution function $F_{\mathbf{x},D}(\mathbf{x}, d)$ in an explicit sense.
2. In theory, it can be minimized with respect to the weight vector $\mathbf{w}$.

Definitions: Principle of Empirical Risk Minimization

- For some fixed $\mathbf{w} = \mathbf{w}^*$, the risk functional $R(\mathbf{w}^*)$ determines the mathematical expectation of a random variable defined by

$$Z_{\mathbf{w}^*} = L\big(d, F(\mathbf{x}, \mathbf{w}^*)\big) \tag{2.77}$$

- The empirical risk functional $R_{emp}(\mathbf{w})$ is the empirical (*arithmetic*) mean of the random variable $Z_{\mathbf{w}^*}$
- The two equations define the differences between the true risk and empirical risk functionals at $\mathbf{w} = \mathbf{w}_{emp}$ and $\mathbf{w} = \mathbf{w}_0$, respectively.
- Since $\mathbf{w} = \mathbf{w}_{emp}$ and $\mathbf{w} = \mathbf{w}_0$ are the minimum points of $R_{emp}(\mathbf{w})$ and $R(\mathbf{w})$, respectively, it follows that

$$R_{emp}\big(\mathbf{w}_{emp}\big) \leq R_{emp}(\mathbf{w}_0) \tag{2.83}$$

- Since $R_{emp}(\mathbf{w}_0) - R(\mathbf{w}_0) < \epsilon$, then $R\big(\mathbf{w}_{emp}\big) - R(\mathbf{w}_0) < 2\epsilon$

Definitions: Principle of Empirical Risk Minimization

- We can make a formal statement of the principle of *empirical risk minimization* in three interrelated parts:

1. In place of the risk functional $R(\mathbf{w})$, construct the empirical risk functional

$$R_{emp}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\big(d_i, F(\mathbf{x}_i, \mathbf{w})\big)$$

on the basis of the training set of i.i.d. examples

$$(\mathbf{x}_i, d_i), \qquad i = 1, 2, \dots, N$$

2. Let $\mathbf{w}_{emp}$ denote the weight vector that minimizes the empirical risk functional $R_{emp}(\mathbf{w})$ over the weight space $\mathcal{W}$. Then $R\left(\mathbf{w}_{emp}\right)$ converges in probability to the minimum possible values of the actual risk $R(\mathbf{w})$, $\mathbf{w} \in \mathcal{W}$ $R(\mathbf{w})$ as the size N of the training sample is made infinitely large.

Definitions: Principle of Empirical Risk Minimization

3. Uniform convergence as defined by

$$P\left(\sup_{\mathbf{w}\in\mathcal{W}} \left| R(\mathbf{w}) - R_{emp}(w) \right| > \epsilon \right) \to 0 \quad \text{as } N \to \infty$$

is a necessary and sufficient condition for the consistency of the principle of empirical risk minimization.

Definitions: VC Dimension

- To describe the concept of the VC (Vapnik and Chervonenkis) dimension,
  - which is a measure of the capacity or expressive power of the family of classification functions realized by the learning machine,
  - consider a binary pattern classification problem for which the desired response is written as $d \in \{0,1\}$.
- Let $\mathcal{F}$ denote the ensemble of dichotomies implemented by a learning machine, that is,

$$\mathcal{F} = \{F(\mathbf{x}, \mathbf{w}) : \mathbf{w} \in \mathcal{W}, F : \mathbb{R}^m \mathcal{W} \rightarrow \{0,1\}\} \qquad (2.85)$$

- Let $\mathcal{L}$ denote the set of N points in the m-dimensional space $\mathcal{H}$ of input vectors, that is,

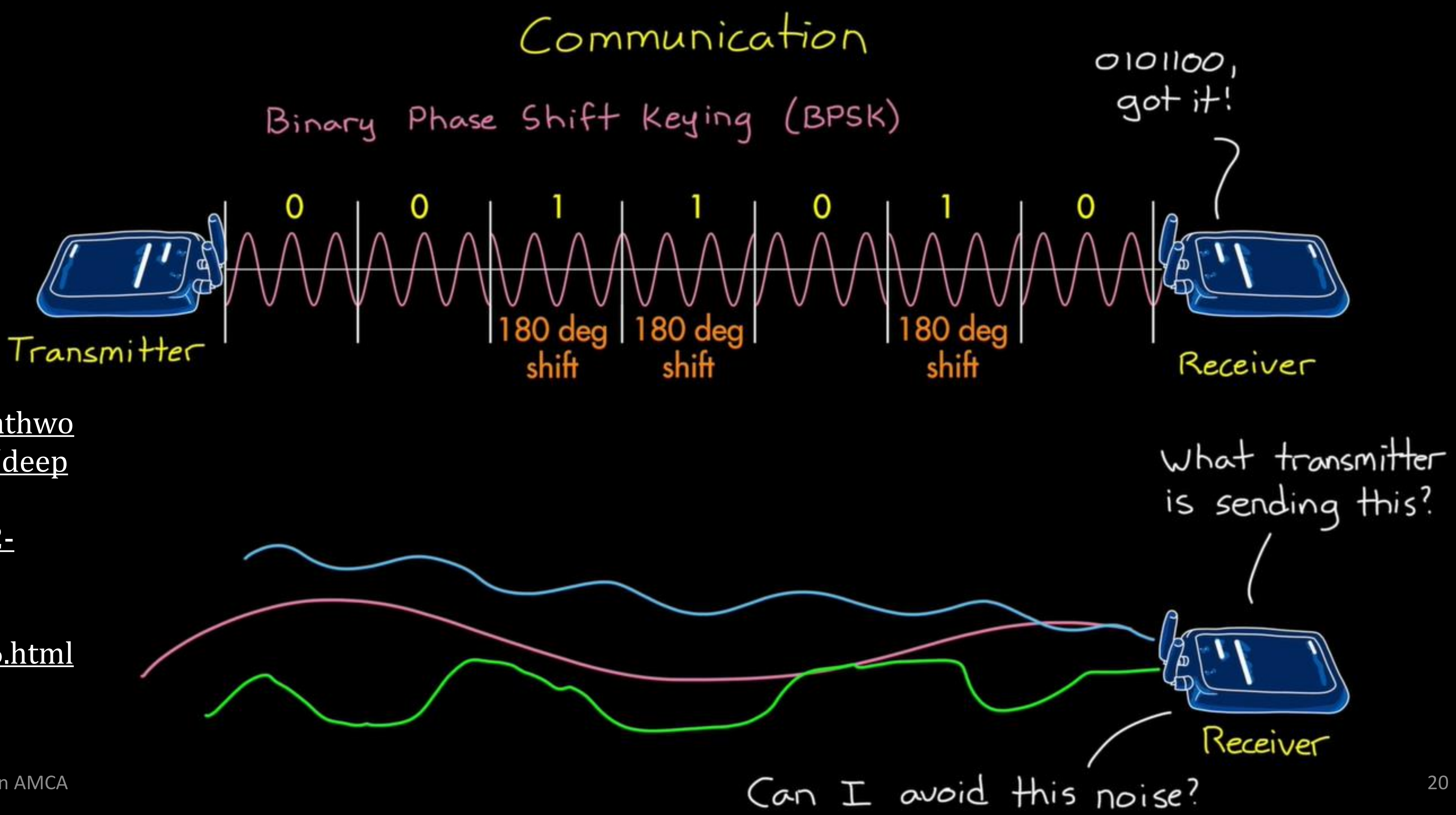
$$\mathcal{L} = \{\mathbf{x}_i \in \mathcal{H}; i = 1, 2, \ldots, N\} \qquad (2.86)$$

Definitions: VC Dimension

- A dichotomy implemented by the learning machine partitions $\mathcal{L}$ into two disjoint subsets $\mathcal{L}_0$ and $\mathcal{L}_1$, such that we may write

$$F(\mathbf{x}, \mathbf{w}) = \begin{cases} 0 & \text{for } \mathbf{x} \in \mathcal{L}_0 \\ 1 & \text{for } \mathbf{x} \in \mathcal{L}_1 \end{cases} \tag{2.87}$$

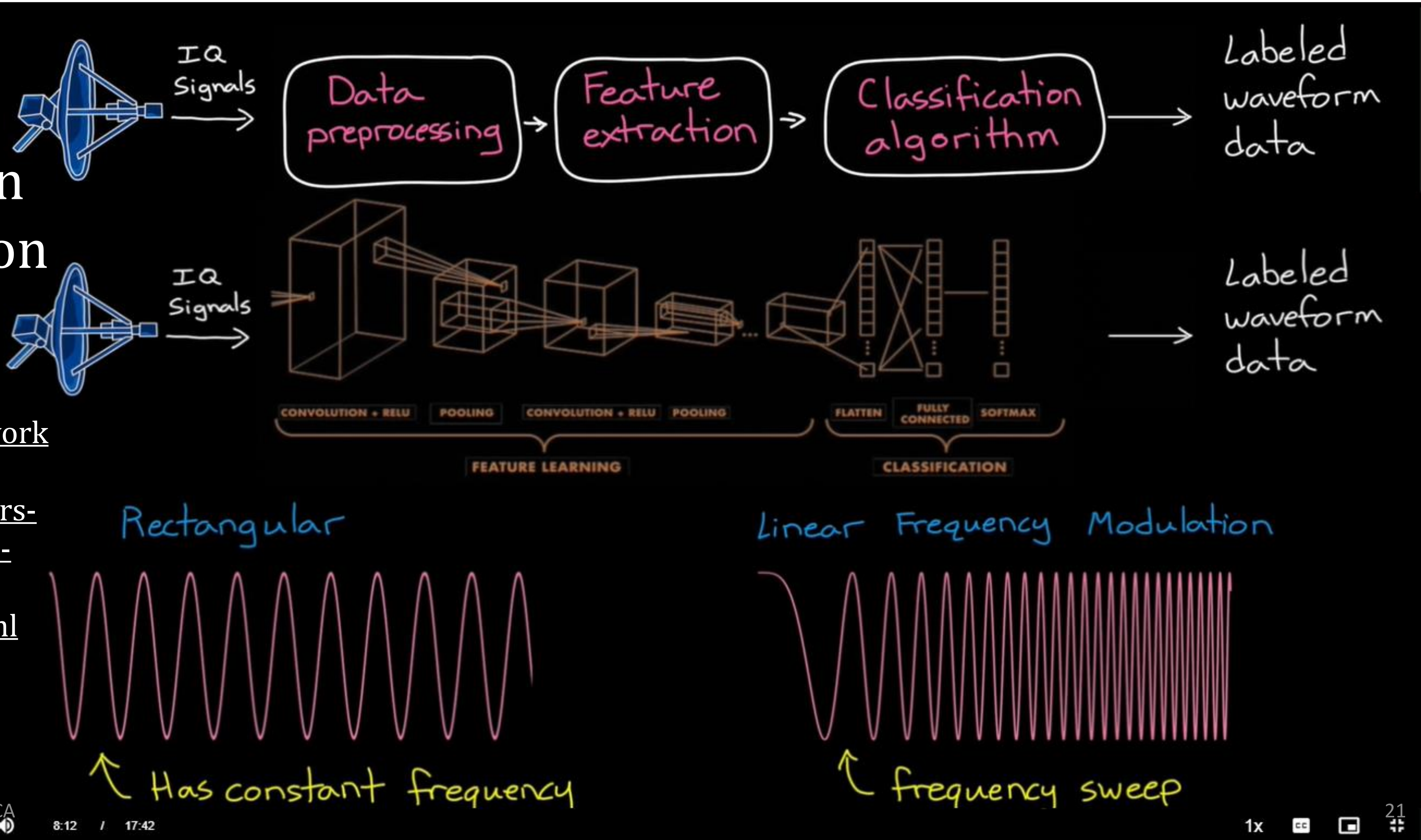
Application: Communication Signal Classification



https://www.mathworks.com/videos/deep-learning-for-engineers-part-2-working-with-synthetic-data-1617266728126.html

Application: Communication Signal Classification

Modulation Identification



https://www.mathworks.com/videos/deep-learning-for-engineers-part-2-working-with-synthetic-data-1617266728126.html

Importance of the VC dimension and its Estimation

- The VC dimension, which plays a central role in statistical learning theory refers to the number of examples needed to learn a class of interest reliably is proportional to the VC dimension of that class.

- The following two results are of special interest:

1. Let $\mathcal{N}$ denote an arbitrary feed forward network built up from neurons with a threshold (Heaviside) activation function:

$$\varphi(v) = \begin{cases} 1 & \text{for } v \geq 0 \\ 0 & \text{for } v < 0 \end{cases}$$

   The VC dimension of $\mathcal{N}$ is O $(W \log W)$ where W is the total number of free parameters in the network.

Importance of the VC dimension and its Estimation

2. Let $\mathcal{N}$ denote a multilayer feedforward network whose neurons use a sigmoid activation function

$$\varphi(v) = \frac{1}{1 + \exp(-v)}$$

The VC dimension of $\mathcal{N}$ is O (W^2), where W is the total number of free parameters in the network.

- A purely linear network has a VC dimension proportional to W as shown in Example 2.2, while a purely threshold neural network has a VC dimension proportional to $W \log W$.

Constructive Distribution-free Bounds on the Generalization Ability of Learning Machines

- Considering a specific case of binary pattern classification, for which the desired response is defined by $d \in \{0,1\}$.
- In a corresponding way the loss function has only two possible values as shown by

$$L\big(d, F(\mathbf{x}, \mathbf{w})\big) = \begin{cases} 0 & if\ F(\mathbf{x}, \mathbf{w}) = d \\ 1 & \text{otherwise} \end{cases}$$

Then, the risk functional $R(\mathbf{w})$ and empirical risk functional $R_{emp}(\mathbf{w})$ will assume the following interpretations:

- The risk functional $R(\mathbf{w})$ is the probability of classification error denoted by P(w).
- The empirical risk functional $R_{emp}(\mathbf{w})$ is the training error (i.e., frequency of errors made during the training session), denoted by $v(\mathbf{w})$.

Constructive Distribution-free Bounds on the Generalization Ability of Learning Machines

- Here, for any weight vector $\mathbf{w}$ and for any precision $\epsilon > 0$, the following condition holds

$$P(|P(\mathbf{w}) - v(\mathbf{w})| > \epsilon \to 0 \quad \text{as } N \to \infty \tag{2.91}$$

where N is the size of the training set.

- For a training set of sufficiently large size N, the proximity between $v(\mathbf{w})$ and $P(\mathbf{w})$ follows from a stronger condition, which stipulates that the following condition holds for any $\epsilon > 0$:

$$P\left(\sup_{\mathbf{w}} |P(\mathbf{w}) - v(\mathbf{w})| > \epsilon\right) \to 0 \quad \text{as } N \to \infty \tag{2.92}$$

- In such a case, we speak of the uniform convergence of the frequency of training errors to the probability that $v(\mathbf{w}) = P(\mathbf{w})$.

Constructive Distribution-free Bounds on the Generalization Ability of Learning Machines

- For the set of classification functions with VC dimension h, the following inequality holds

$$P\left(\sup_{\mathbf{w}} |P(\mathbf{w}) - v(\mathbf{w})| > \epsilon\right) < \left(\frac{2eN}{h}\right)^h \exp(-\epsilon^2 N) \qquad (2.93)$$

where N is the size of the training sample and e is the base of natural logarithm.

- The right-hand side of the (2.93) should be small for large N in order to achieve uniform convergence.
- The right-hand side of (2.93) will go to zero as N goes to infinity; this requirement is satisfied if the VC dimension h is finite.
- In other words, a finite VC dimension is a necessary and sufficient condition for uniform convergence of the principle of empirical risk minimization.

Constructive Distribution-free Bounds on the Generalization Ability of Learning Machines

- Let α denote the probability of occurrence of the event

$$\sup_{\mathbf{w}} |P(\mathbf{w}) - v(\mathbf{w})| \geq \epsilon$$

Then, with probability $1 - \alpha$, we may state that for all weight vectors $\mathbf{w} \in \mathcal{W}$ the following inequality holds:

$$P(\mathbf{w}) < v(\mathbf{w}) + \epsilon \tag{2.94}$$

From (2.93) we have

$$\alpha = \left(\frac{2eN}{h}\right)^h \exp(-\epsilon^2 N) \tag{2.95}$$

Constructive Distribution-free Bounds on the Generalization Ability of Learning Machines

- Then the *confidence interval* $\epsilon_0(N, h, \alpha)$ will be given by

$$\epsilon_0(N, h, \alpha) = \sqrt{\frac{h}{N}\left[\log\left(\frac{2N}{h}\right) + 1\right] - \frac{1}{N}\log\alpha} \qquad (2.96)$$

Structural Risk Minimization

- The *training error* $v_{train}(\mathbf{w})$ or $v(\mathbf{w})$ is the frequency of errors made by a learning machine of some weight vector $\mathbf{w}$ during the training session.

- Similarly, the generalization error $v_{gene}(\mathbf{w})$ is defined as the frequency of errors made by the machine when it is tested with examples not seen before.

- Let h be the VC dimension of a family of classification functions $\{F(\mathbf{x}, \mathbf{w})\mathbf{w} \in \mathcal{W}\}$ with respect to the input space $\mathcal{H}$.

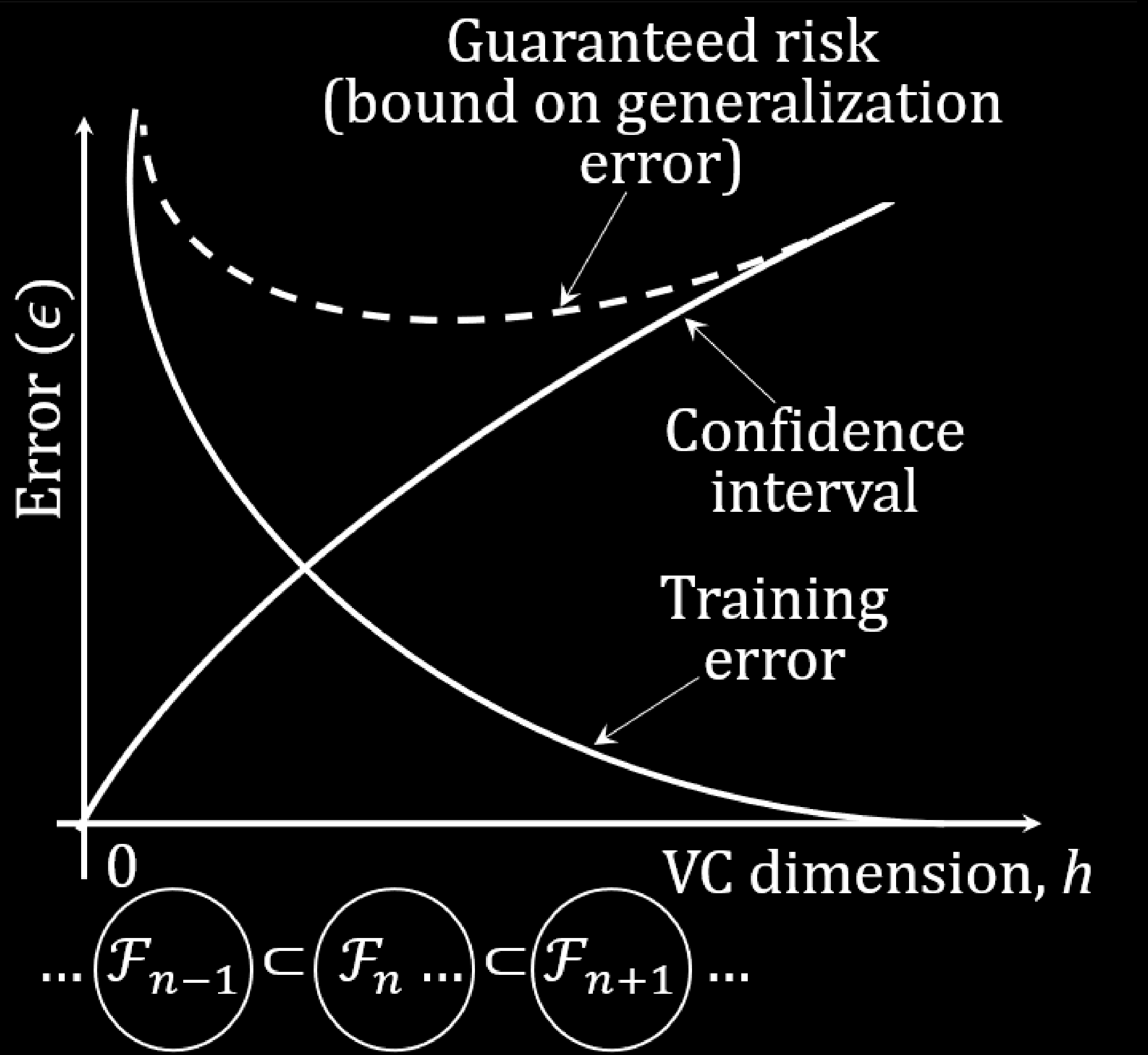
- Then, we may state that with probability $1 - \alpha$, for a number of training examples $N > h$, and simultaneously for all classification functions $F(\mathbf{x}, \mathbf{w})$, the generalization error $v_{gene}(\mathbf{w})$ is lower than a guaranteed risk defined by the sum of a pair of competing terms.

$$v_{guarant}(\mathbf{w}) = v_{train}(\mathbf{w}) + \epsilon_1(N, h, \alpha, v_{train})$$

Structural Risk Minimization

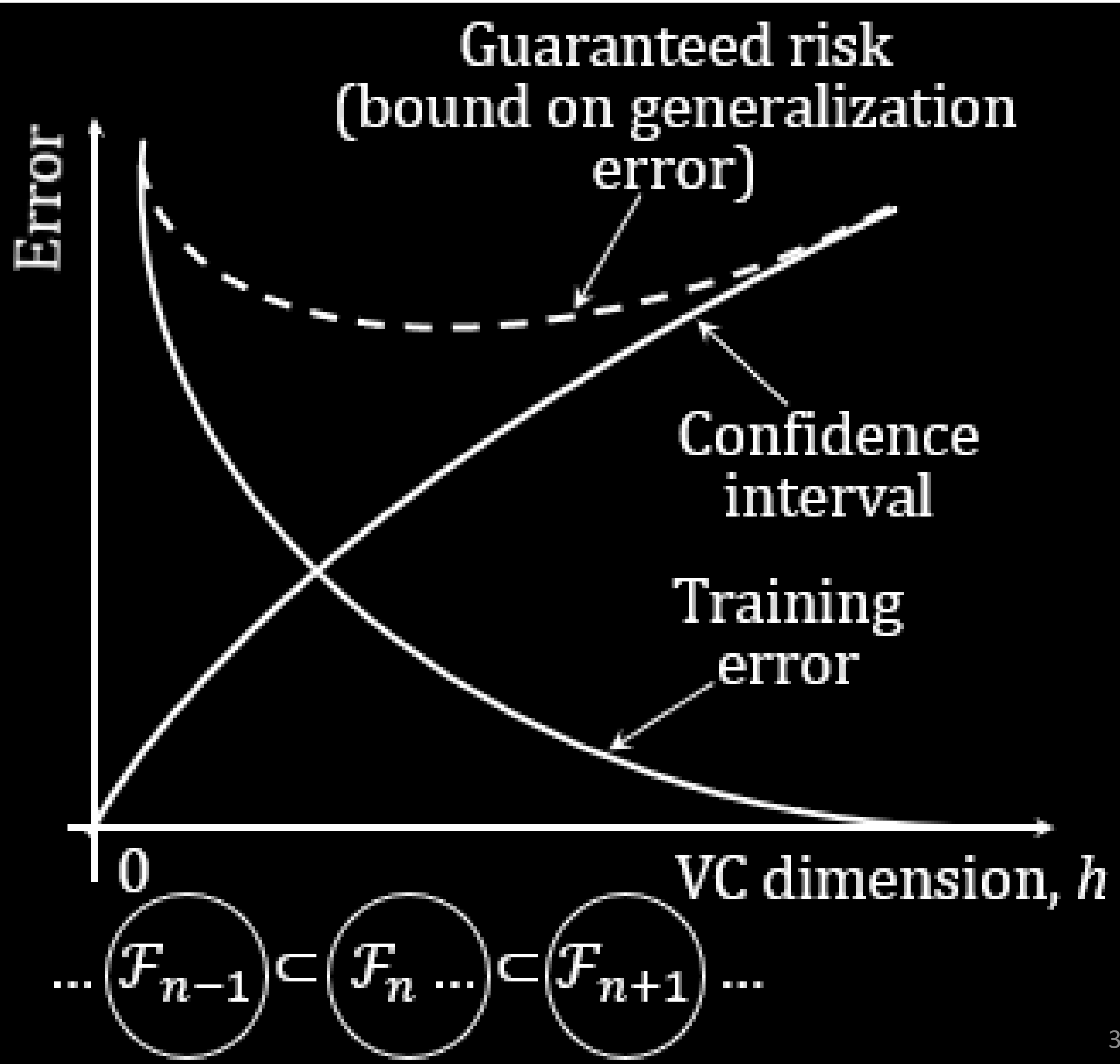
- Illustration of the relationship between training error, confidence interval and guaranteed risk are shown in Fig. 2.25.

Fig. 2.25 Illustration of the relationship between training error, confidence interval, and guaranteed risk.



Structural Risk Minimization

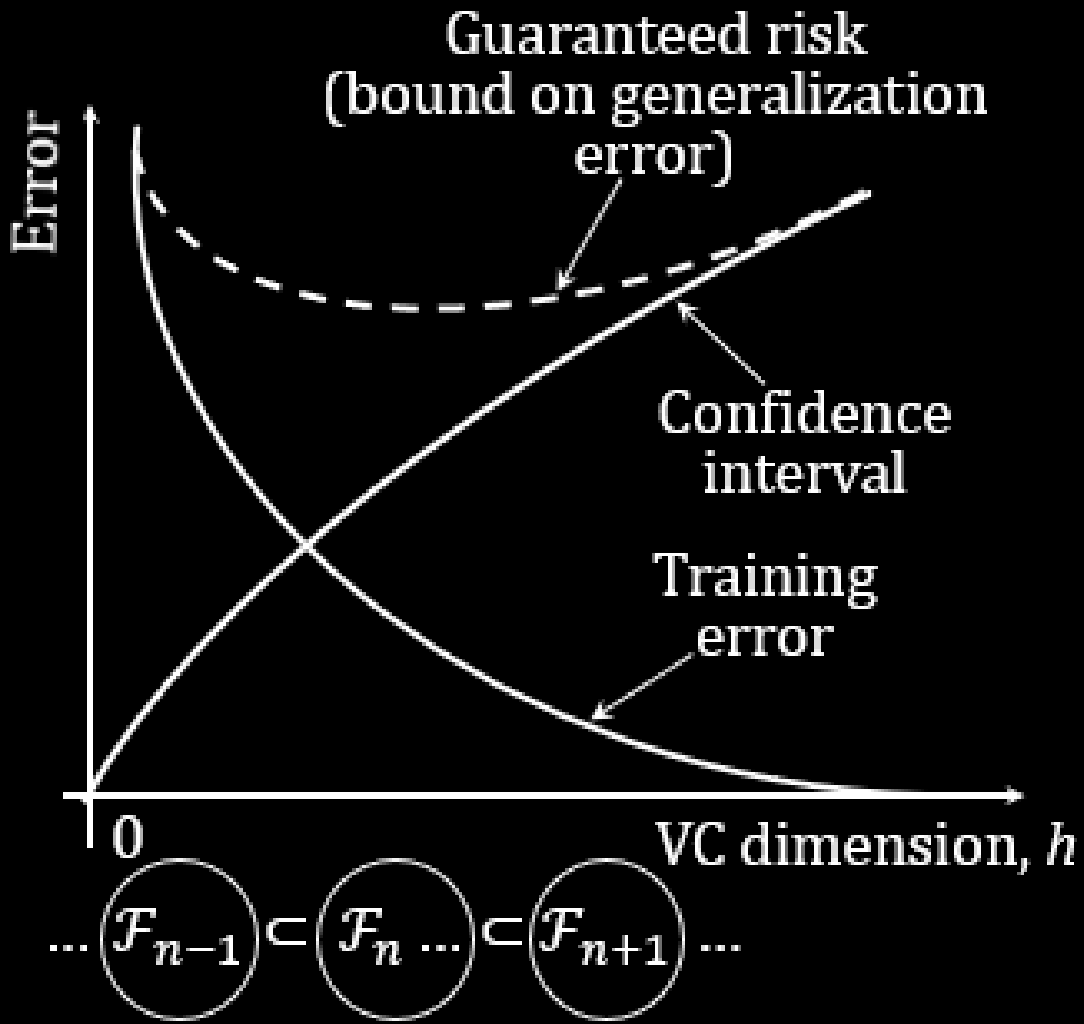
- Before the minimum point is reached, the *learning problem* is *overdetermine*d since the *machine capacity h is too small* for the amount of training data



Structural Risk Minimization

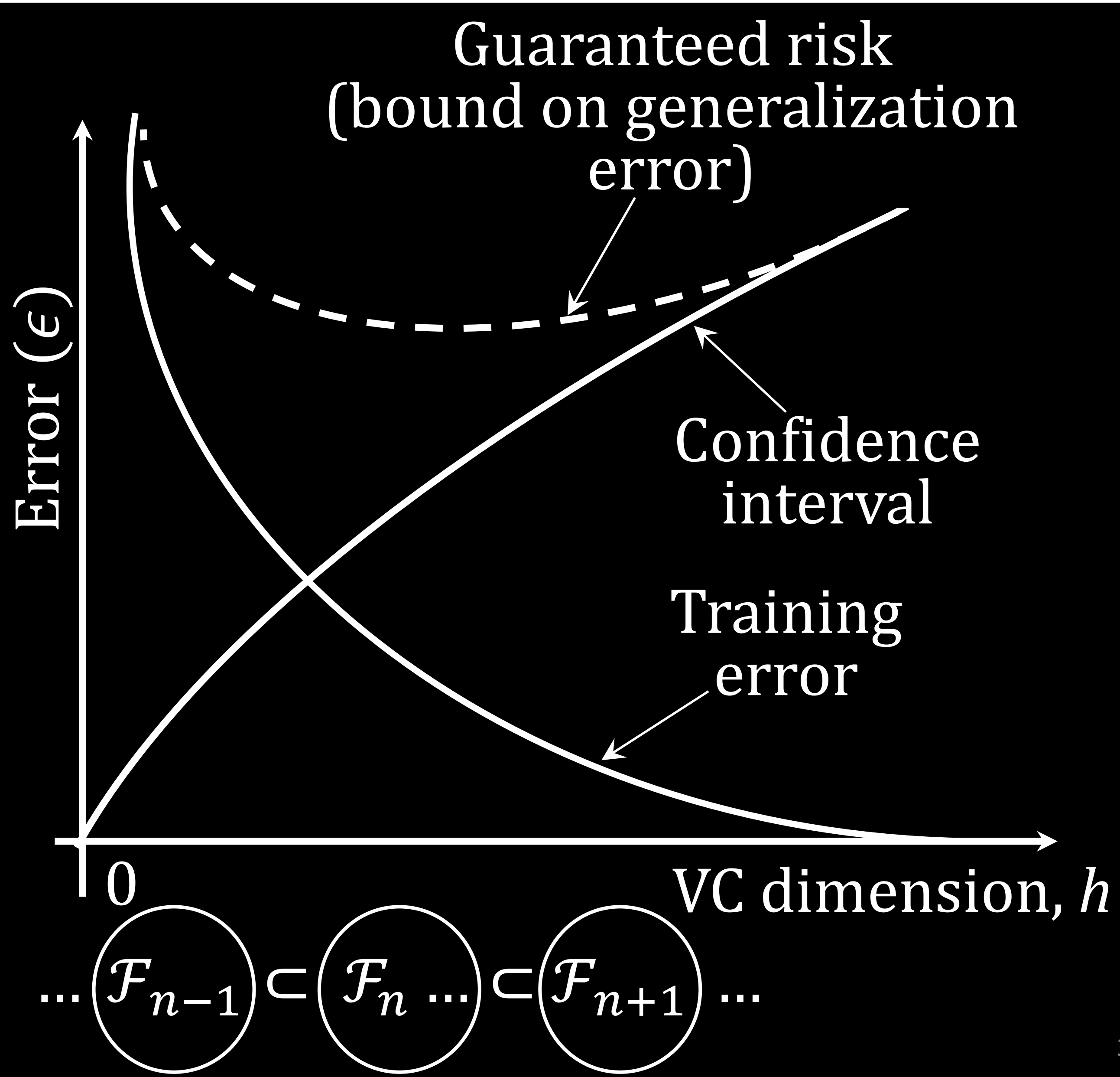
- Beyond the minimum point, the learning problem is *underdetermined* because the machine capacity is too large for the amount of training data.

- The challenge in solving a *supervised learning* problem is to realize the best generalization performance by matching the **machine capacity** to the available **amount of training data** for the problem at hand.



Structural Risk Minimization

Fig. 2.25 Illustration of the relationship between training error, confidence interval, and guaranteed risk.



Structural Risk Minimization

- To be specific, consider an ensemble of pattern classifiers $\{F(\mathbf{x}, \mathbf{w}); \mathbf{w} \in \mathcal{W}\}$ and define a nested structure of n such machines

$$\mathcal{F}_k = \{F(\mathbf{x}, \mathbf{w}); \mathbf{w} \in \mathcal{W}\}, \qquad k = 1, 2, 3, \dots, n \tag{2.102}$$

$$\text{As in Fig.2.25}, \mathcal{F}_1 \subset \mathcal{F}_2 \ \dots \ \subset \mathcal{F}_n \tag{2.103}$$

where the symbol $\subset$ signifies "is contained in."

- Correspondingly, the VC dimensions of the individual pattern classifiers satisfy the condition

$$h_1 \leq h_2 \leq \cdots \leq h_n \tag{2.104}$$

which implies that the VC dimension of each pattern classifier is finite.

Structural Risk Minimization

- The method of *structural risk minimization* may proceed as follows:
  - ~ The empirical risk (i.e., training error) for each pattern classifier is minimized.
  - ~ The pattern classifier $\mathcal{F}^*$ with the smallest guaranteed risk is identified; this particular machine provides the best compromise between the training error (i.e., quality of approximation of the training data) and the confidence interval, (i.e., complexity of the approximating function) which compete with each other.
- Our goal is to find a network structure such that decreasing the VC dimension occurs at the expense of the smallest possible increase in training error.

Structural Risk Minimization

- The principle of structural risk minimization may be implemented in a variety of ways.
- For example, we may vary the VC dimension h by varying the number of hidden neurons.
- The principle of structural risk minimization states that the best network in this ensemble is the one for which the guaranteed risk is the minimum.

2.15 Probably Approximately Correct Model of Learning

- The Probably Approximately Correct (PAC) learning model is a probabilistic framework for the study of learning and generalization in binary classification systems.
- It is closely related to supervised learning.
- We begin with an environment $\mathcal{H}$. A set of $\mathcal{H}$ is called a concept, and a set of subsets of $\mathcal{H}$ is called a concept class.

2.15 Probably Approximately Correct Model of Learning

- An example of a concept is an object in the domain of interest, together with a class label.
- If the example is a member of the concept, we refer to it as a positive example; if the object is not a member of the concept, we refer to it as a negative example.
- A concept for which examples are provided is called a target concept.

2.15 Probably Approximately Correct Model of Learning

- We may acquire a sequence of training data of length N for a target concept c as shown by

$$\mathcal{T} = \{(\mathbf{x_i}, c(\mathbf{x}_i))\}_{i=1}^{N} \tag{2.105}$$

which may contain repeated examples.

- The examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are drawn from the environment $\mathcal{H}$ at random, according to some fixed but unknown probability distribution.

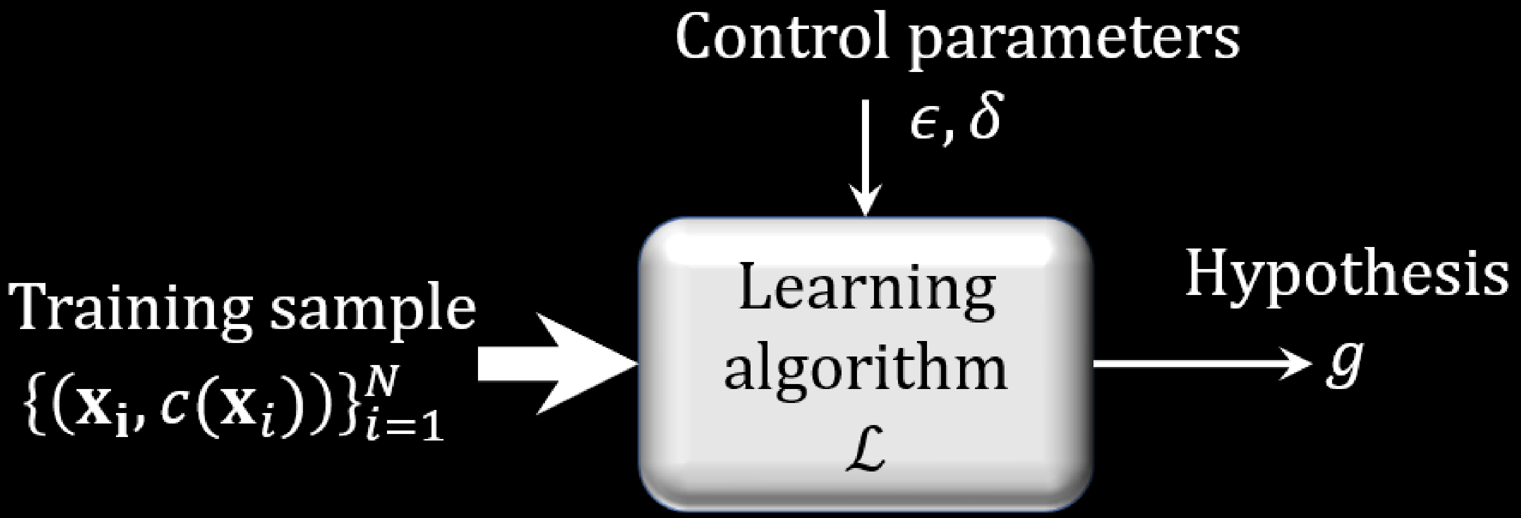


Fig. 2.26 Block diagram illustrating the PAC learning model.

2.15 Probably Approximately Correct Model of Learning

- The following points are also noteworthy in Eq. (2.105):
  - ~ Target concept $c(\mathbf{x}_i)$ (unknown) is treated as a function from $\mathcal{H}$ to $\{0, 1\}$.
  - ~ The examples are usually assumed to be statistically independent, which means that the joint probability density function of any two examples, $\mathbf{x}_i$ and $\mathbf{x}_j$, are equal to the product of their individual probability density functions.

2.15 Probably Approximately Correct Model of Learning

- The concept space $\mathcal{C}$ derived from the environment $\mathcal{H}$ may contain "the letter A", "the letter B," and so on.
- Concepts may be coded to generate a set of positive and negative examples.
- A learning machine typically represents a set of functions, with each function corresponding to a specific state.
- For example, the machine may be designed to recognize "the letter A," "the letter B," and so on.
- The set of all functions (i.e., concepts) determined by the learning machine is referred to as a hypothesis space $\mathcal{C}$.

2.15 Probably Approximately Correct Model of Learning

- One way of assessing the success of the learning process is to measure how close the hypothesis $g(\mathbf{x})$ is to the target concept $c(\mathbf{x})$.
- There will be errors incurred, making $g(\mathbf{x}) \neq c(\mathbf{x})$ since we are trying to learn a function on the basis of limited information available about function.
- The probability of training error is defined by

$$v_{train} = P(x \in \mathcal{H}: g(\mathbf{x}) \neq c(\mathbf{x})) \tag{2.106}$$

2.15 Probably Approximately Correct Model of Learning

- The learning algorithm is supplied with two control parameters:
  - ~ Error parameter $\epsilon \in (0,1]$ which specifies the error allowed in a good approximation of the target concept $c(\mathbf{x})$ by the hypothesis $g(\mathbf{x})$.
  - ~ Confidence parameter $\delta \in (0,1]$ which controls the likelihood of constructing a good approximation.

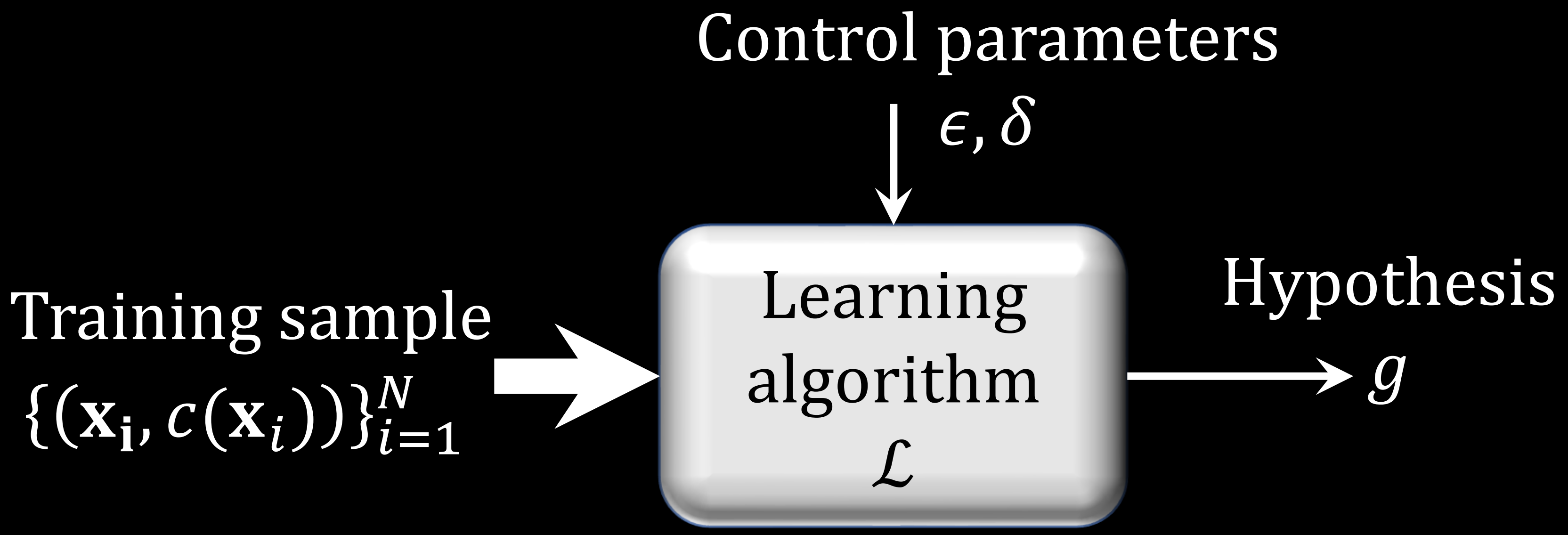


Fig. 2.26 Block diagram illustrating the PAC learning model.

Sample Complexity

- The focus in this issue is on

  ~how many random examples should be presented to the learning algorithm for it to acquire sufficient information to learn an unknown target concept c chosen from the concept class $\mathcal{C}$. Or,

  ~how large should the size N of the training set $\mathcal{T}$ be?

  sample comblexity $\propto$ VC dimension

- Let $\mathcal{T} = \{(\mathbf{x_i}, d_i)\}_{i=1}^{N}$ be any set of labeled examples, where each $x_i \in \mathcal{H}$ and each $d_i \in (0, 1)$. Let c be a target concept over the environment $\mathcal{H}$.
- Then, concept c is said to be consistent with the training set $\mathcal{T}$ if for all $1 \leq i \leq N$, we have $c(\mathbf{x}_i) = d_i$.

Sample Complexity

- When PAC learning is concerned, we have a key result presented in two parts
- Consider a neural network with a finite VC dimension $h \geq 1$.
  1. Any consistent learning algorithm for that neural network is a PAC learning algorithm.
  2. There is a constant K such that a sufficient size of training set $\mathcal{T}$ for any such algorithm is

$$N = \frac{K}{\epsilon}\left(hlog\left(\frac{1}{\epsilon}\right) + \log\left(\frac{1}{\delta}\right)\right) \qquad (2.107)$$

where ϵ is the error parameter and δ is the confidence parameter.

Computational Complexity

- *Computational complexity* deals with the worst-case "running time" needed to train a neural network (learning machine), given a set of labeled examples of some finite size N.
- The running time and therefore computational complexity, is usually measured in terms of the number of operations (additions, multiplications, and storage) needed to perform the computation.
- In assessing the computational complexity of a learning algorithm, we like to know how it varies with the example size m (i.e., size of the input layer of the neural network being trained).
- The running time should be $\mathrm{O}(m')$ for some fixed integer $r \geq 1$. Hence,the running time is said to increase polynomially with m and the algorithm itself is said to be a *polynomial time algorithm*.

Computational Complexity

- The other parameter that requires attention is the error parameter ϵ.
- In sample complexity of a learning algorithm , the parameter ϵ is fixed,
- In computational complexity the running time polynomial is $\propto 1/\epsilon$.
- The formal statement on computational complexity is the given as
  - ~ A learning algorithm is computationally efficient with respect to error parameter ϵ, example size m, and size N of the training set if its running time is polynomial in N and if there is a value of $N_0(\delta, \epsilon)$ sufficient for PAC learning that is polynomial in both m and ϵ^{-1}.

2.16 Summary and Discussion

- R1, pp. 126. In PAC learning theory, an issue of particular interest with