EENG582
Artificial Neural Networks

Learning Process

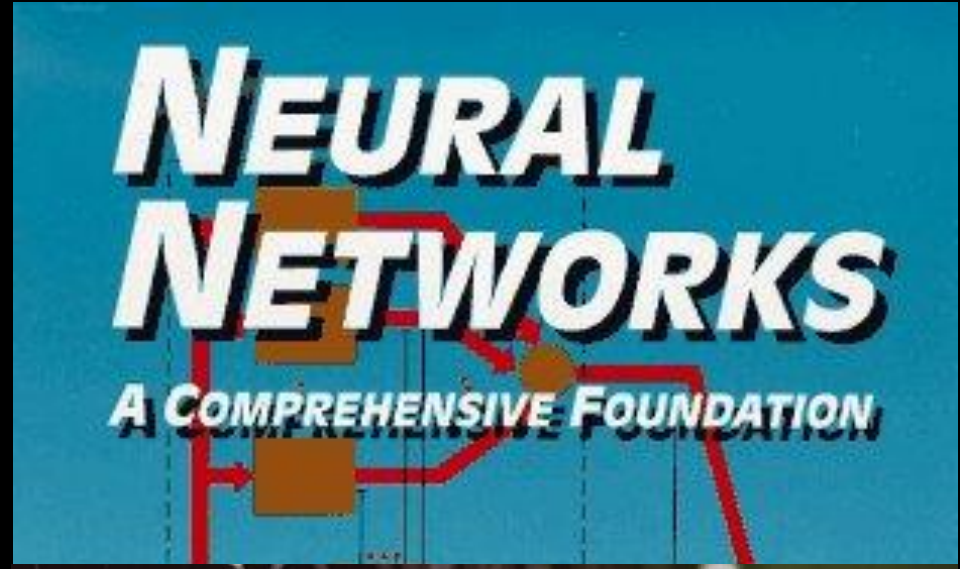
Sections 2.8 – 2.10
Learning with – without a Teacher + Learning Tasks

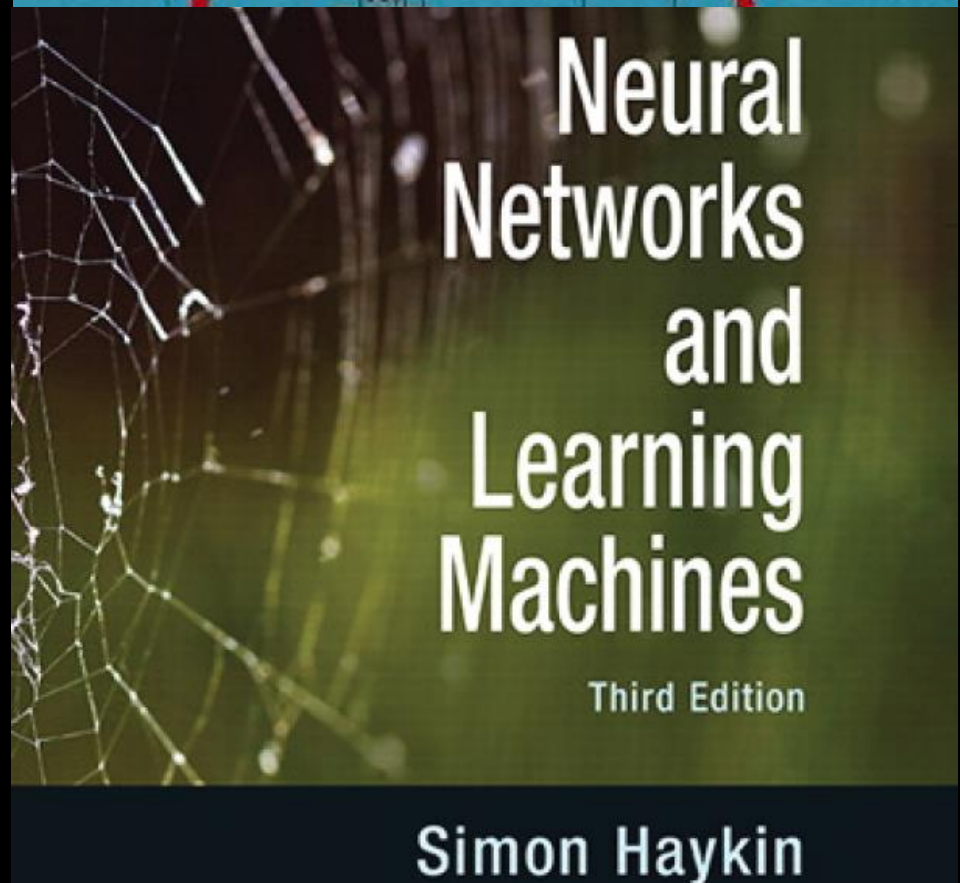
Artificial Neural Networks

Prof. Dr. Hasan AMCA

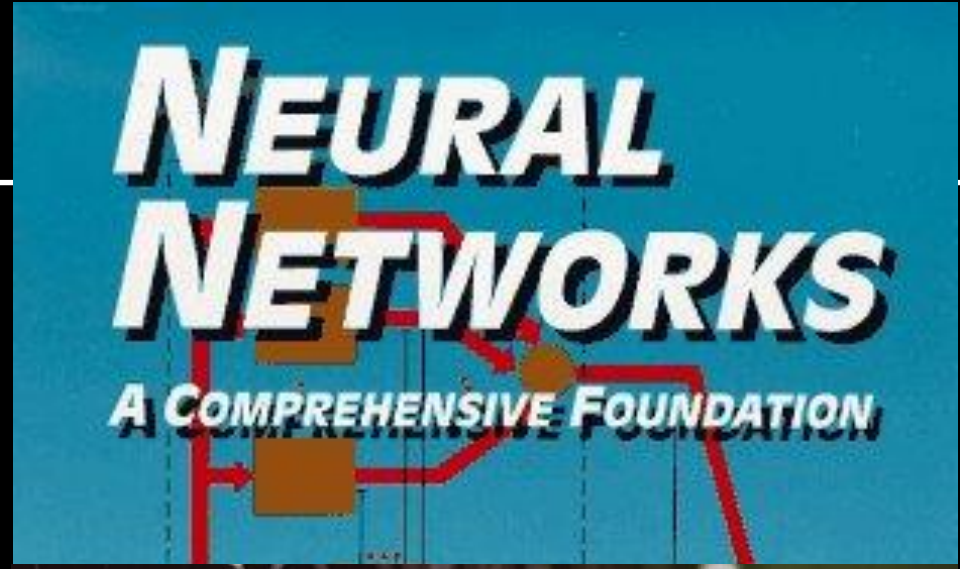
Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

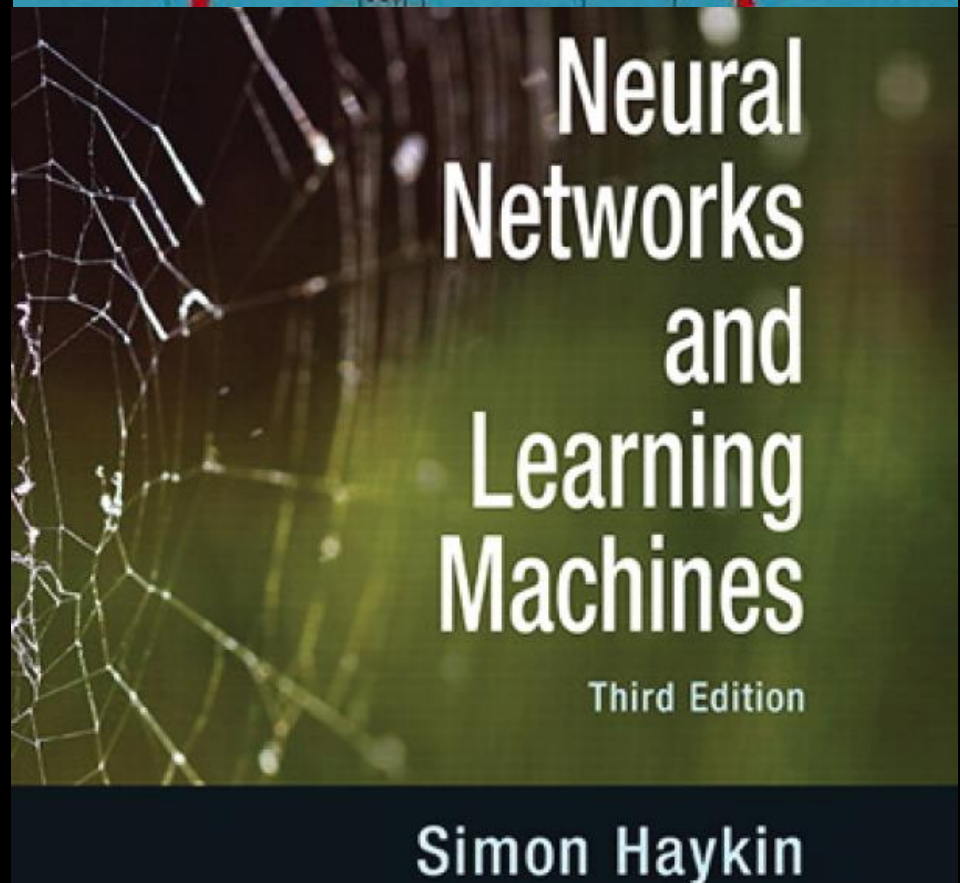
Eastern Mediterranean University
(emu.edu.tr)





2 Learning Process





2.8 Learning with a Teacher

- Learning with teacher is also referred to as supervised learning as shown in Fig. 2.6
- Think of the teacher as having knowledge of the environment represented by a set of input-output examples.
- The environment is, however, unknown to the neural network of interest.
- Suppose that the teacher and the neural network are both exposed to a training vector (i.e., example) drawn from the environment.
- By virtue of built-in knowledge, the teacher is able to provide the neural network with a desired response for that training vector.

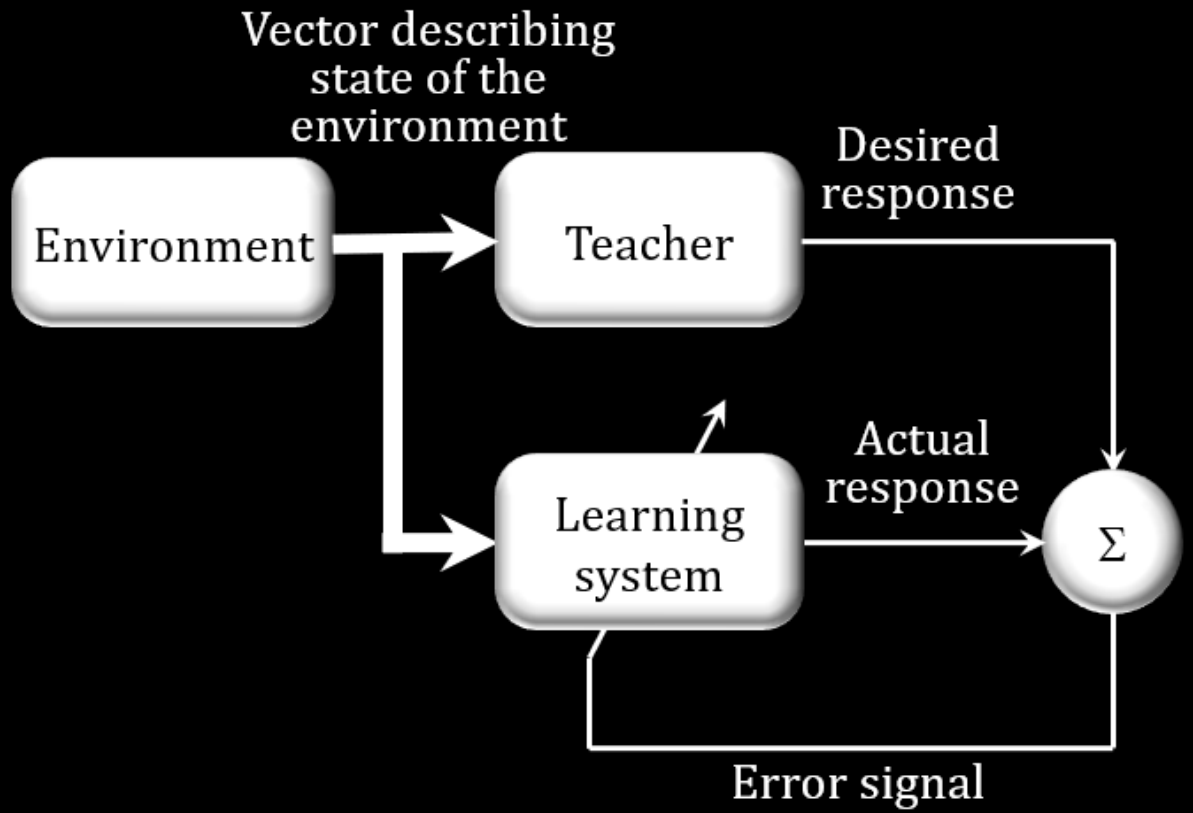


Fig. 2.6 Block diagram of learning with a teacher.

2.8 Learning with a Teacher

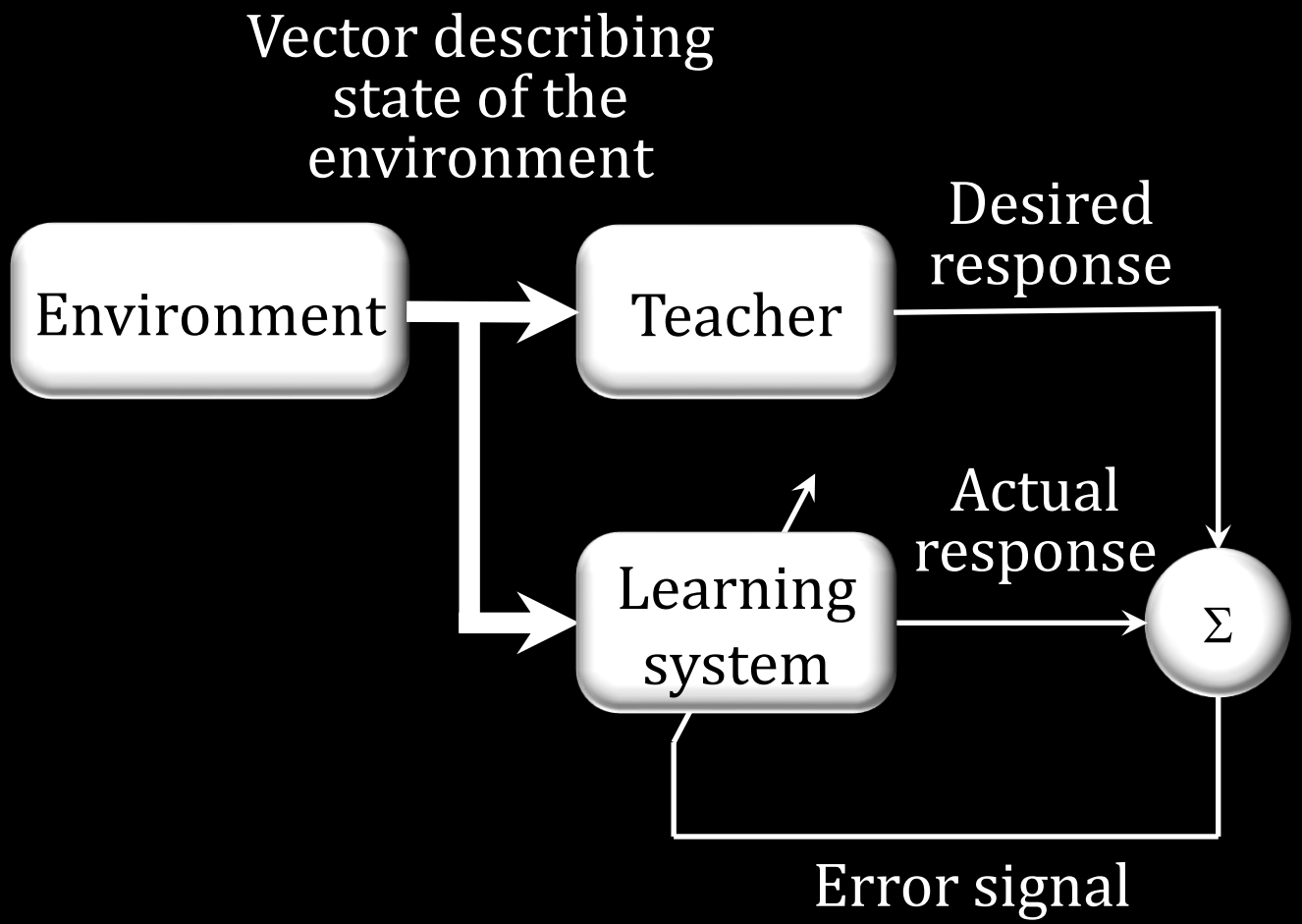


Fig. 2.6 Block diagram of learning with a teacher.

2.8 Learning with a Teacher

- The network parameters are adjusted under the combined influence of **training vector** and **error signal** which is the difference between the **desired response** and the **actual response** of the network.
- This adjustment is done iteratively in a step-by-step fashion with the aim of making network emulate the teacher
- Hence, knowledge of the environment available to teacher is transferred to the neural network through training
- When this condition is reached, we then dispense with the teacher and let the neural network deal with the environment completely by itself.

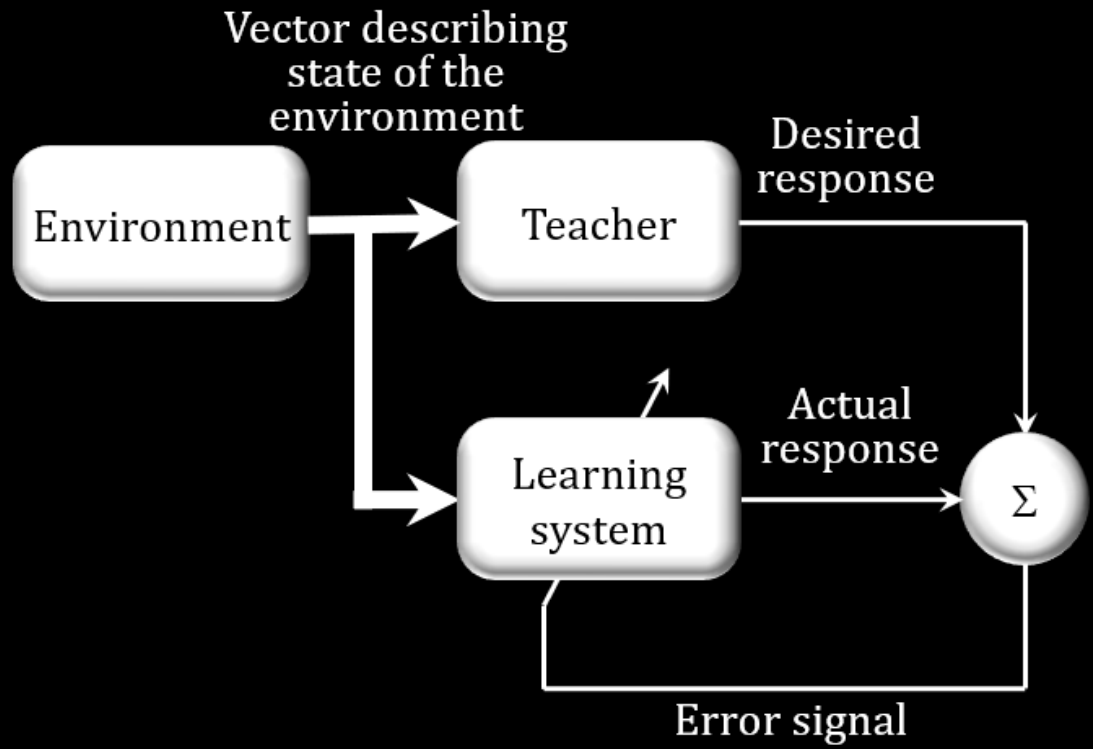
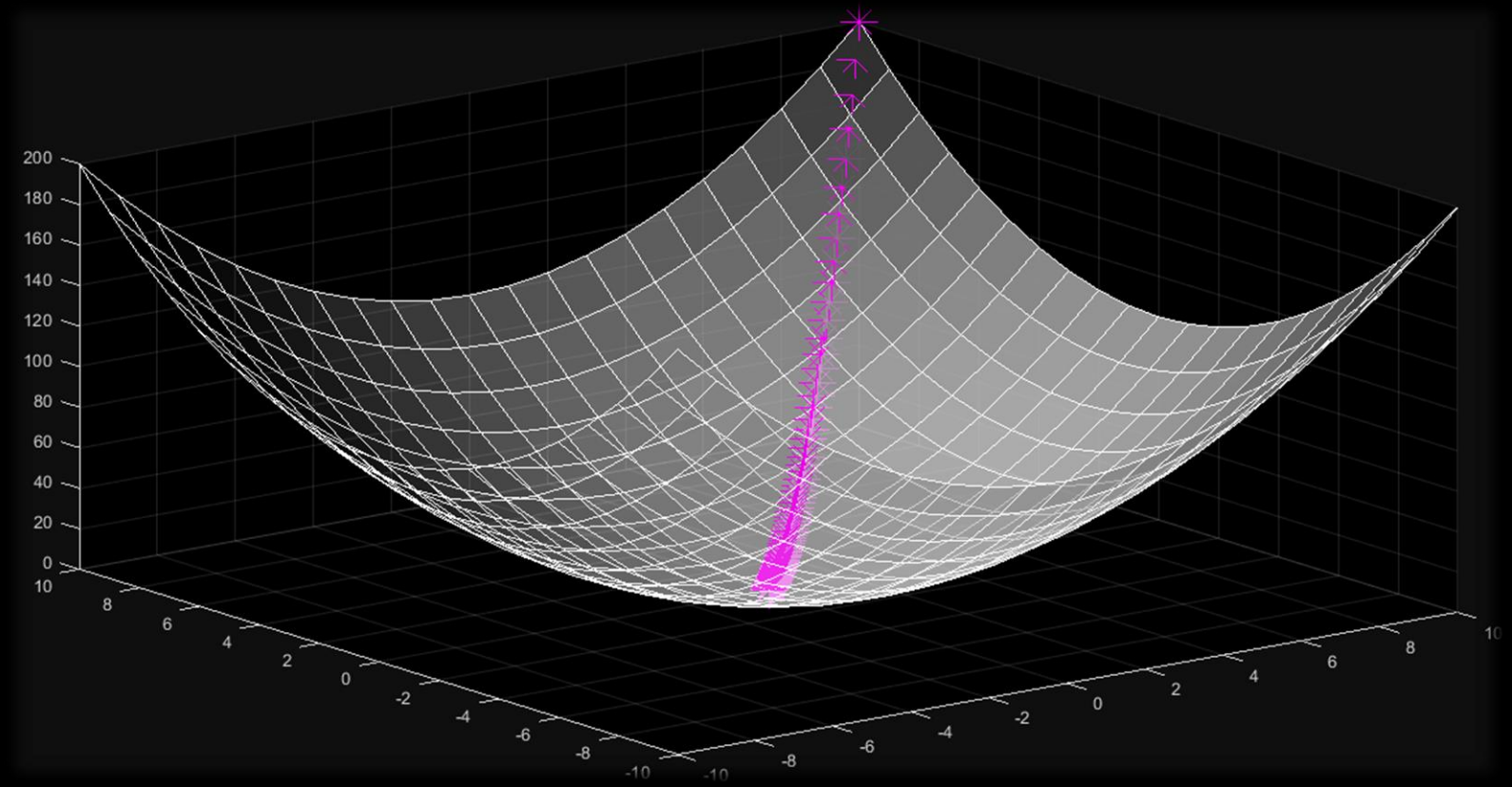


Fig. 2.6 Block diagram of learning with a teacher.

2.8 Learning with a Teacher

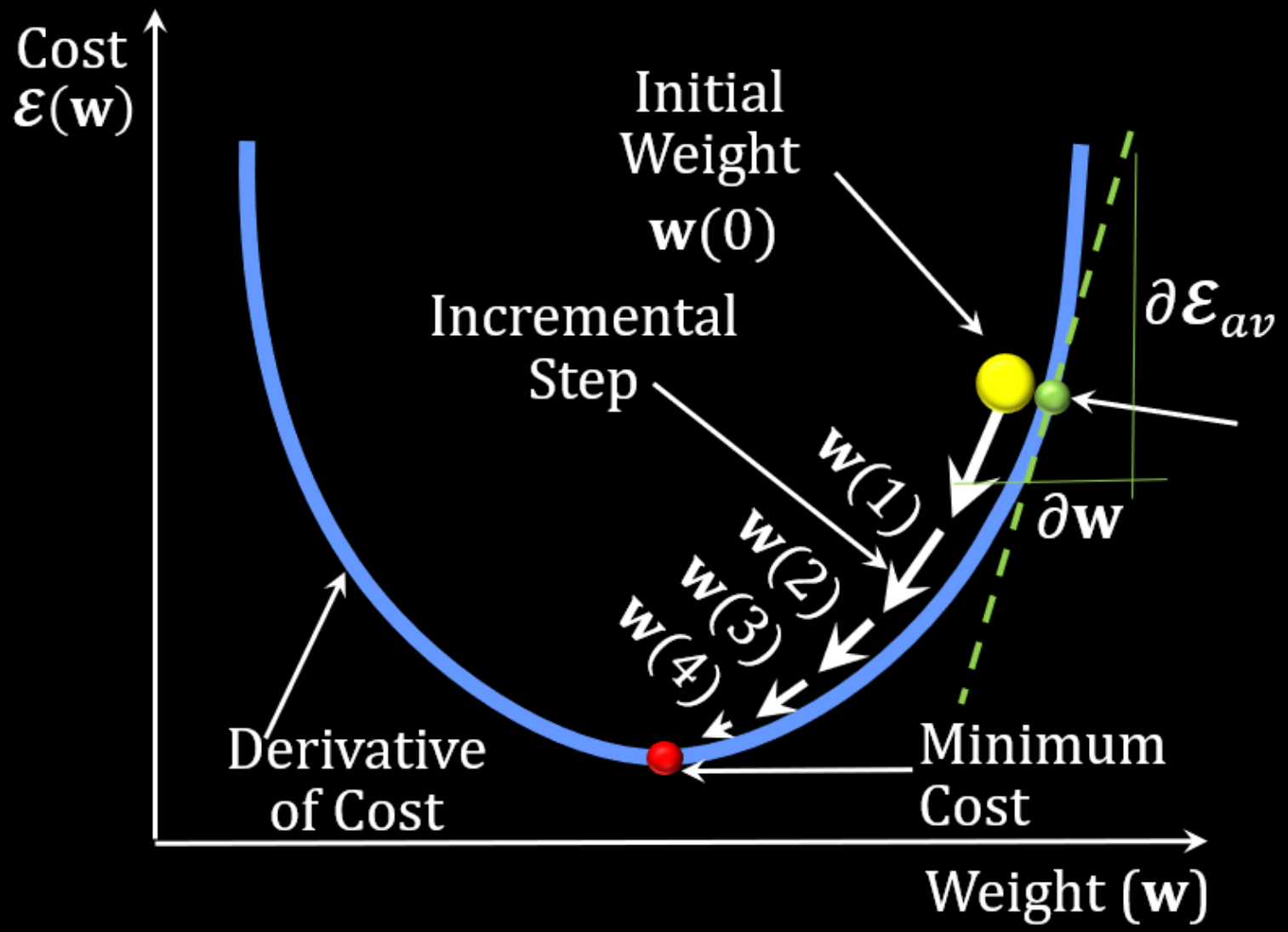
- A performance measure for the system is the mean-square error over training
- The true **error surface** is averaged over all possible input-output examples.
- Any given operation of the system under the teacher's supervision is represented as a point on the error surface.

- To improve performance over time and learn from the teacher, the operating point has to move down toward a minimum point of the error surface;
- Hence a local minimum or a global minimum.

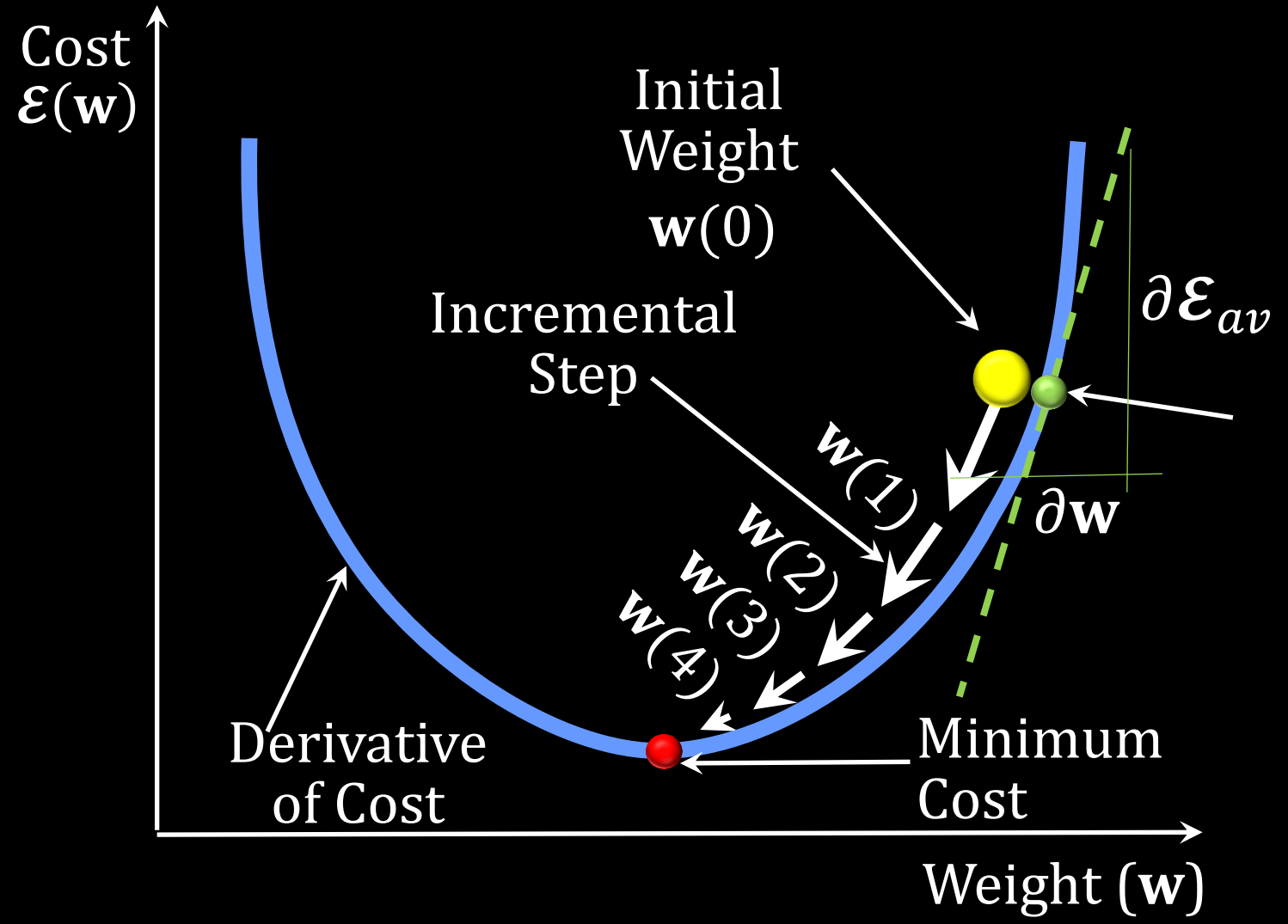


2.8 Learning with a Teacher

- In supervised learning, the system may use an instantaneous estimate of the gradient vector, with the example indices presumed to be those of time.
- With a good algorithm to minimize cost function, adequate input-output examples, and enough time to do the training, a supervised learning system is usually able to perform such tasks as pattern classification and function approximation.



2.8 Learning with a Teacher



2.9 Learning without a Teacher

- In supervised learning, the learning process takes place under the guidance of a teacher.
- However, in learning without a teacher, there is no teacher to oversee the learning process.
- Hence, there are no labeled examples of the function to be learned by the network.
- Under this second paradigm, two subdivisions are identified.
  1. Reinforcement learning/Neurodynamic programming.
  2. Unsupervised learning

1. Reinforcement Learning/Neurodynamic Programming

- In reinforcement learning, the learning of an input-output mapping is performed through continued interaction with the environment in order to minimize a scalar index of performance.
- Figure 2.7 shows the block diagram of one form of a reinforcement learning system built around a primary reinforcement signal received from the environment into a higher quality reinforcement signal called the heuristic reinforcement signal.

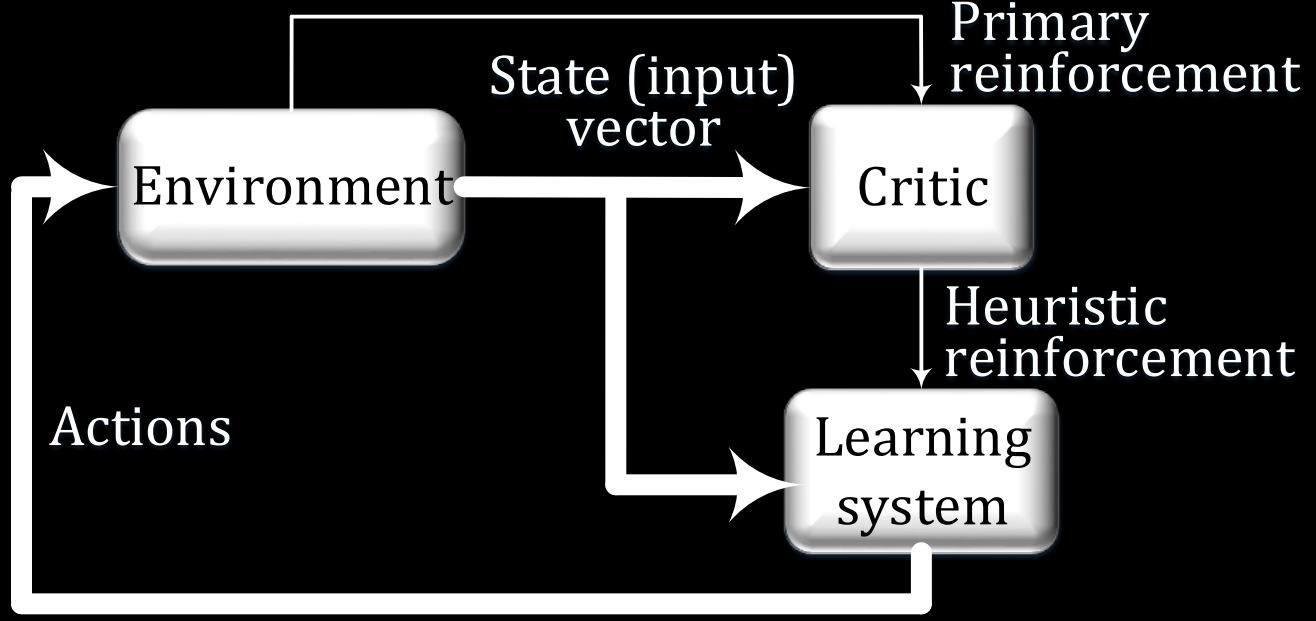


Fig. 2.7 Block diagram of a reinforcement learning.

1. Reinforcement Learning/Neurodynamic Programming

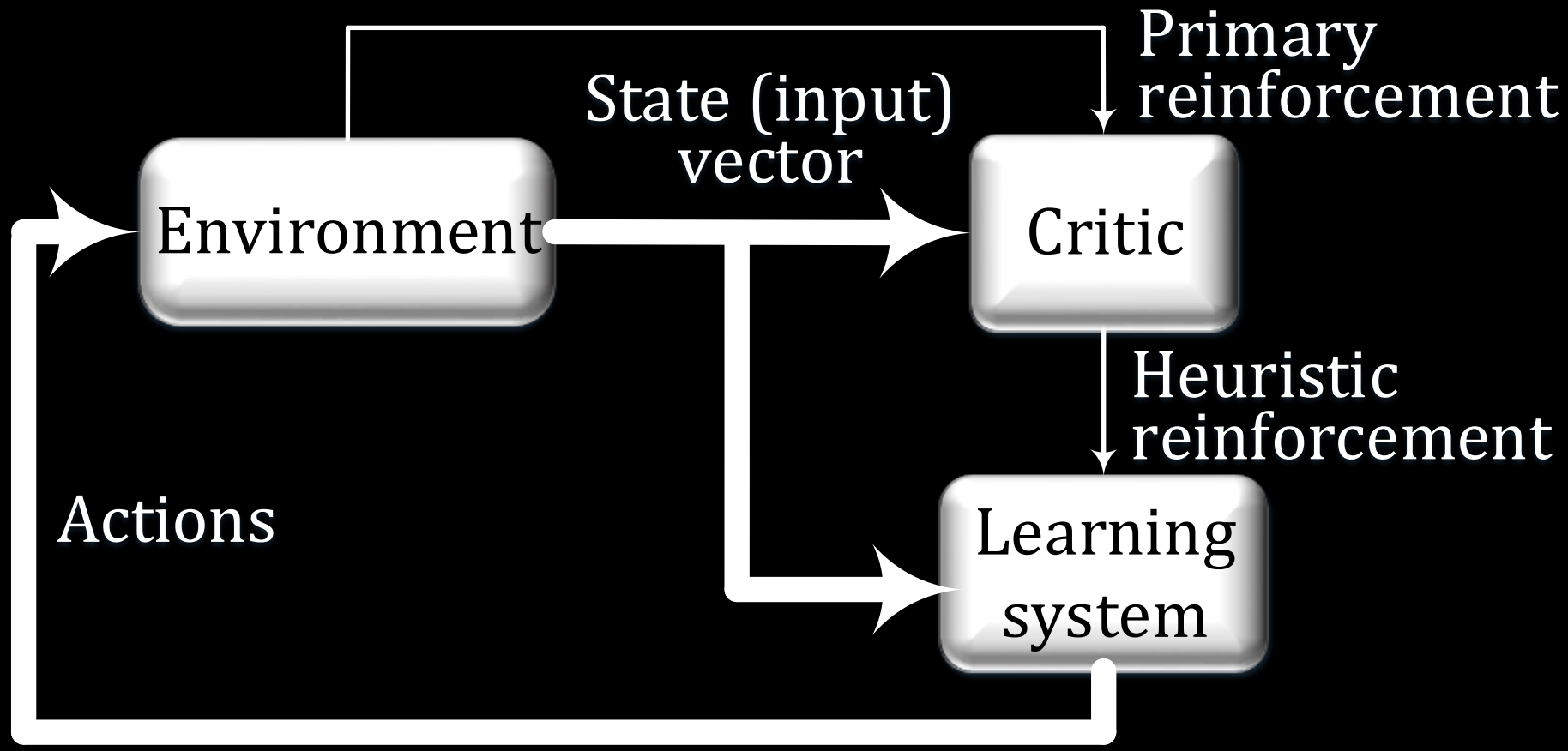
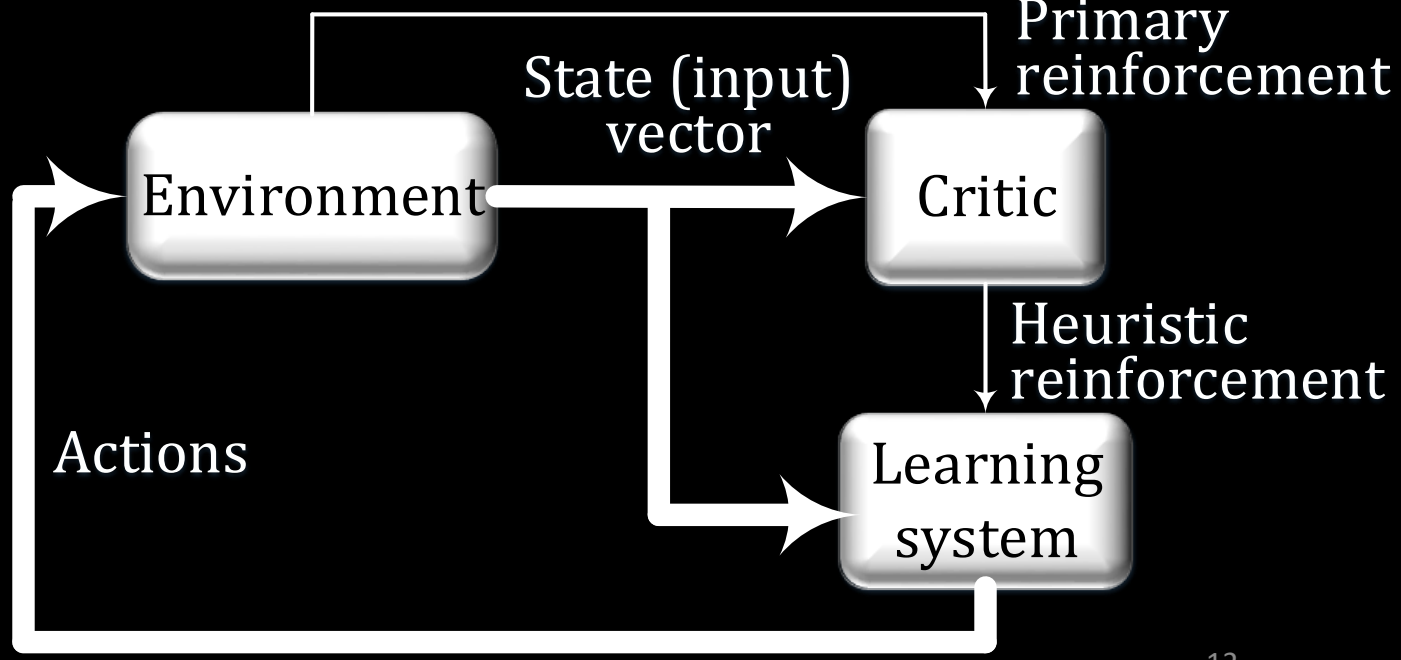


Fig. 2.7 Block diagram of a reinforcement learning.

1. Reinforcement Learning/Neurodynamic Programming

- The system learns under delayed reinforcement by observing a temporal sequence of stimuli (i.e., state vectors) also received from the environment, which eventually result in generation of the heuristic reinforcement signal.
- The goal of learning is to minimize a cost-to-go function, defined as the expectation of the cumulative cost of actions taken over a sequence of steps.
- The function of the learning machine, which constitutes the second component of the system, is to discover these actions and to feed them back to the environment.



1. Reinforcement Learning/Neurodynamic Programming

- Delayed-reinforcement learning is difficult to perform for two basic reasons:
  - ~ There is no teacher to provide a desired response at each step of the learning process.
  - ~ The learning machine must be able to assign credit and blame individually to each action in the sequence of time steps that led to the final outcome, while the primary reinforcement may only evaluate the outcome.

2. Unsupervised Learning

- In *unsupervised* or *self-organized* learning there is no external teacher or critic to oversee the learning process, as indicated in Fig. 2.8. below.
- Rather, provision is made for a task independent measure of the quality and the free parameters of the network are optimized w.r.t. that measure.

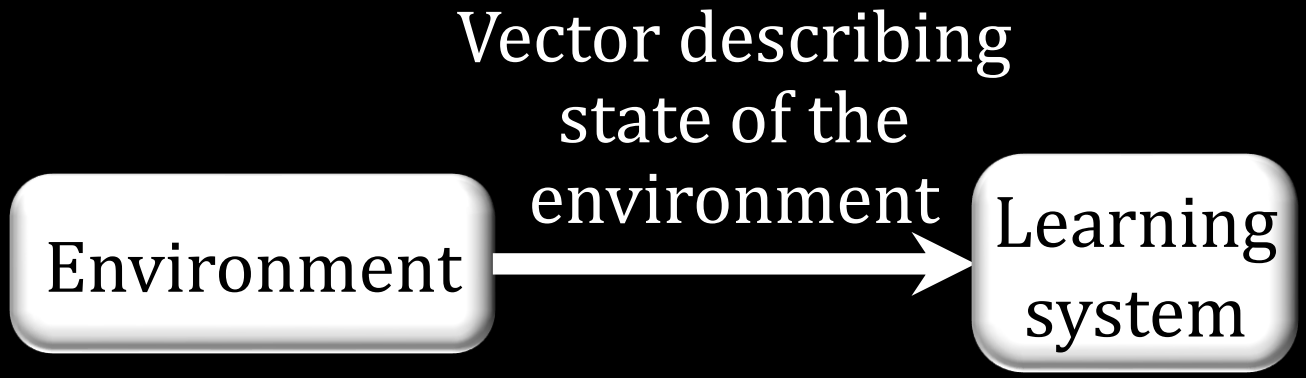
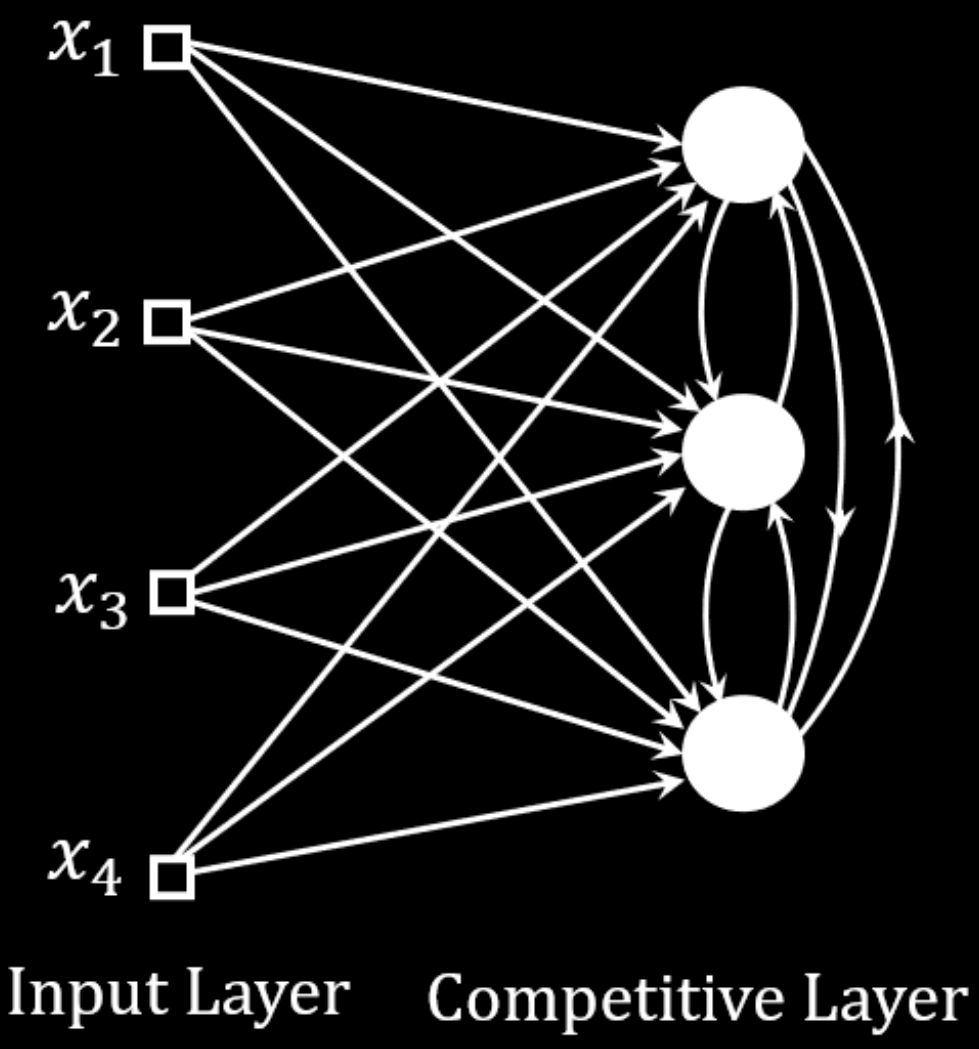


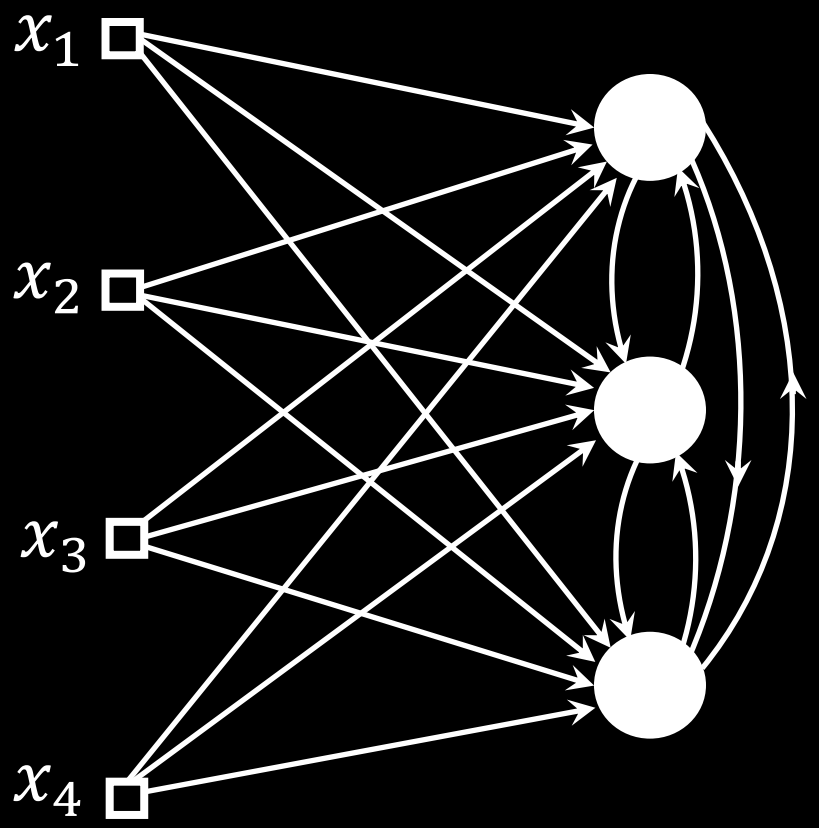
Fig. 2.8 Block diagram of unsupervised learning .

2. Unsupervised Learning

- Unsupervised learning uses a competitive learning rule.
- For example, a neural network that consists of two layers an input layer and a competitive layer could be used.
- The input layer receives the available data.
- The competitive layer consists of neurons that compete with each other for the "opportunity" to respond to features contained in the input data.
- the neuron with the greatest total input "wins" the competition and turns on; all the othe!" neurons then switch off.



2. Unsupervised Learning



2.10 Typical Problem Areas / Learning Tasks

- The choice of a particular learning algorithm is influenced by the learning task that a neural network is required to perform
- In this context we identify following learning tasks/problem areas

1. Pattern Association / Completion
2. Pattern Recognition
3. Function Approximation
4. Control
5. Filtering
6. Beamforming
7. Classification
8. Optimization
9. Feature Detection
10. Data Compression
11. Approximation
12. Prediction

2.10 Learning Tasks: Pattern Association

- Association is a prominent feature of human memory and all models of cognition use association as the basic operation.
- Association takes one of two forms:

Autoassociation	Heteroassociation
In autoassociation, a neural network is required to store a set of patterns (vectors) by repeatedly presenting them to the network. The network is then presented a partial description or distorted (noisy) version of an original pattern stored in it, and the task is to retrieve (recall) that particular pattern.	Heteroassociation differs from autoassociation in that an arbitrary set of input patterns is paired with another arbitrary set of output patterns.
Autoassociation involves the use of unsupervised learning.	Heteroassociation involves the use of supervised learning.

2.10 Learning Tasks: Pattern Association

- Let $\mathbf{x}_k$ denote a key pattern (vector) applied to an associative memory and $\mathbf{y}_k$ denote a memorized pattern (vector).
- The pattern association performed by the network is described by

$$\mathbf{x}_k \rightarrow \boldsymbol{y}_k \quad \text{k=1, 2, 3, … , } q \tag{2.18}$$

where q is the number of patterns stored in the network.

- The key pattern $\mathbf{x}_k$ acts as a stimulus that not only determines the storage location of memorized pattern $\mathbf{y}_k$ but also holds the key for its retrieval.
- In an autoassociative memory, $\mathbf{y}_k = \mathbf{x}_k$, so input and output (data) have the same dimensionality.
- In a heteroassociative memory, $\mathbf{y}_k \neq \mathbf{x}_k$; hence, the dimensionality of the output space may or may not equal to the dimensionality of the input space.

2.10 Learning Tasks: Pattern Association

- There are two phases involved in the operation of an associative memory:
  - **Storage phase**, which refers to the training of the network in accordance with Eq. (2.18).
  - **Recall phase**, which involves the retrieval of a memorized pattern in response to presentation of a noisy or distorted version of a key pattern to the network.

2.10 Learning Tasks: Pattern Association

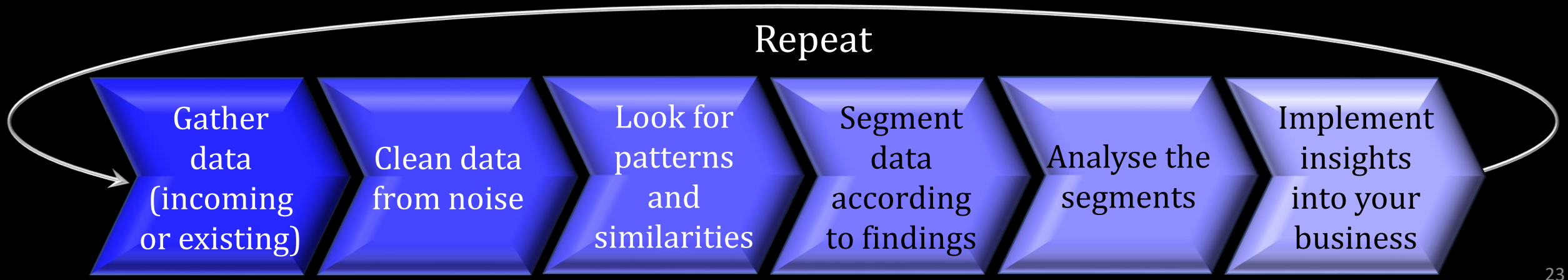
- Let the stimulus (input) $\mathbf{x}$ represent a noisy or distorted version of a key pattern $\mathbf{x}_j$.
- This stimulus produces a response (output) $\mathbf{y}$, as indicated in Fig. 2.9.
- For perfect recall, we should find that $\mathbf{y} = \mathbf{y}_j$, where $\mathbf{y}_j$ is the memorized pattern associated with the key pattern $\mathbf{x}_j$.
- When $\mathbf{y} \neq \mathbf{y}_j$ for $\mathbf{x} = \mathbf{x}_j$, the associative memory is said to have made an error in recall.

2.10 Learning Tasks: Pattern Recognition

- Humans can
  - recognize familiar face of a person even though that person aged since last encounter,
  - identify a familiar person by his or her voice on the telephone despite a bad connection,
  - and distinguish a boiled egg that is good from a bad one by smelling it.
- Pattern recognition is defined as the process whereby a received pattern signal is assigned to one of a prescribed number of classes (categories).

2.10 Learning Tasks: Pattern Recognition

- Neural networks do pattern recognition by first undergoing a training session
  - during which the network is repeatedly presented a set of input patterns along with the category to which each particular pattern belongs.
- Later, a new pattern is presented to network that has not been seen before,
  - but which belongs to the same population of patterns used to train the network.
  - The network is able to identify the class of that particular pattern because of the information it has extracted from the training data.



What Is Pattern Recognition in Machine Learning (huspi.com)

2.10 Learning Tasks: Pattern Recognition

- Pattern-recognition machines using neural networks may take one of two forms
  1. The machine is split into two parts, an unsupervised network for feature extraction and a supervised network for classification, as shown in Fig. 2.10a.
     ~ a pattern is represented by a set of m observables, which may be viewed as a point $\mathbf{x}$ in an m-dimensional observation (data) space.
     ~ Feature extraction is described by a transformation that maps the point $\mathbf{x}$ into an intermediate point $\mathbf{y}$ in a q-dimensional feature space with $q < m$, as indicated in Fig. 2.10b.
  2. The machine is designed as a single multilayer feedforward network using a supervised learning algorithm. In this second approach, the task of feature extraction is performed by the computational units in the hidden layer(s) of the network.

2.10 Learning Tasks: Pattern Recognition

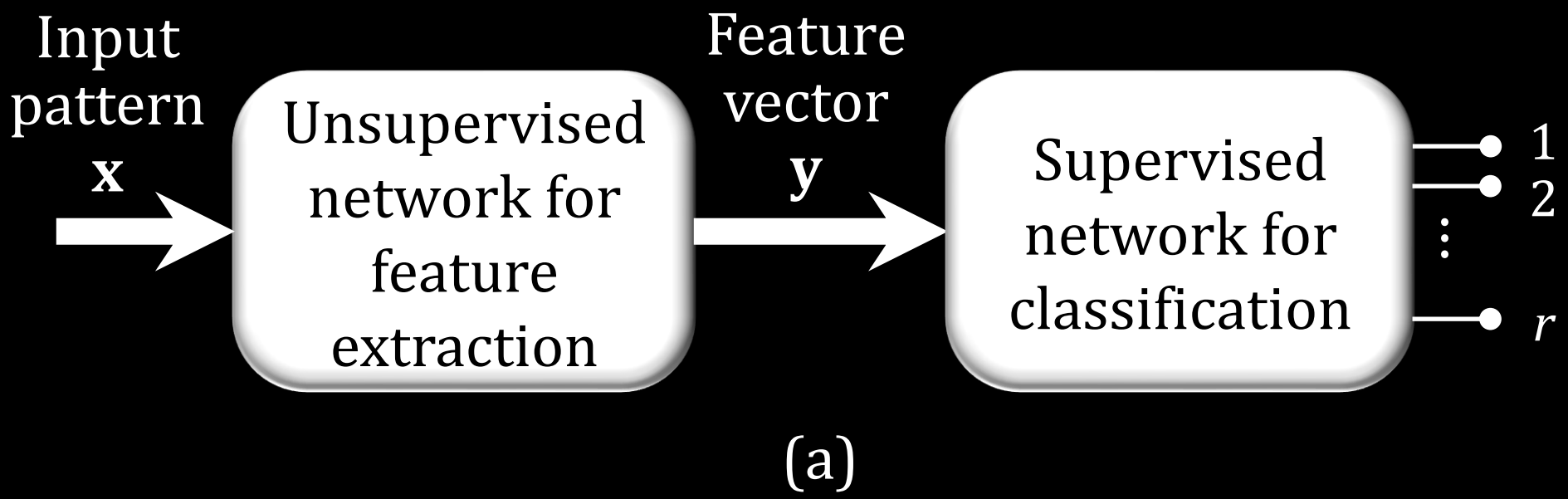


Fig. 2.10 Illustration of the classical approach to pattern classification.

2.10 Learning Tasks: Pattern Recognition

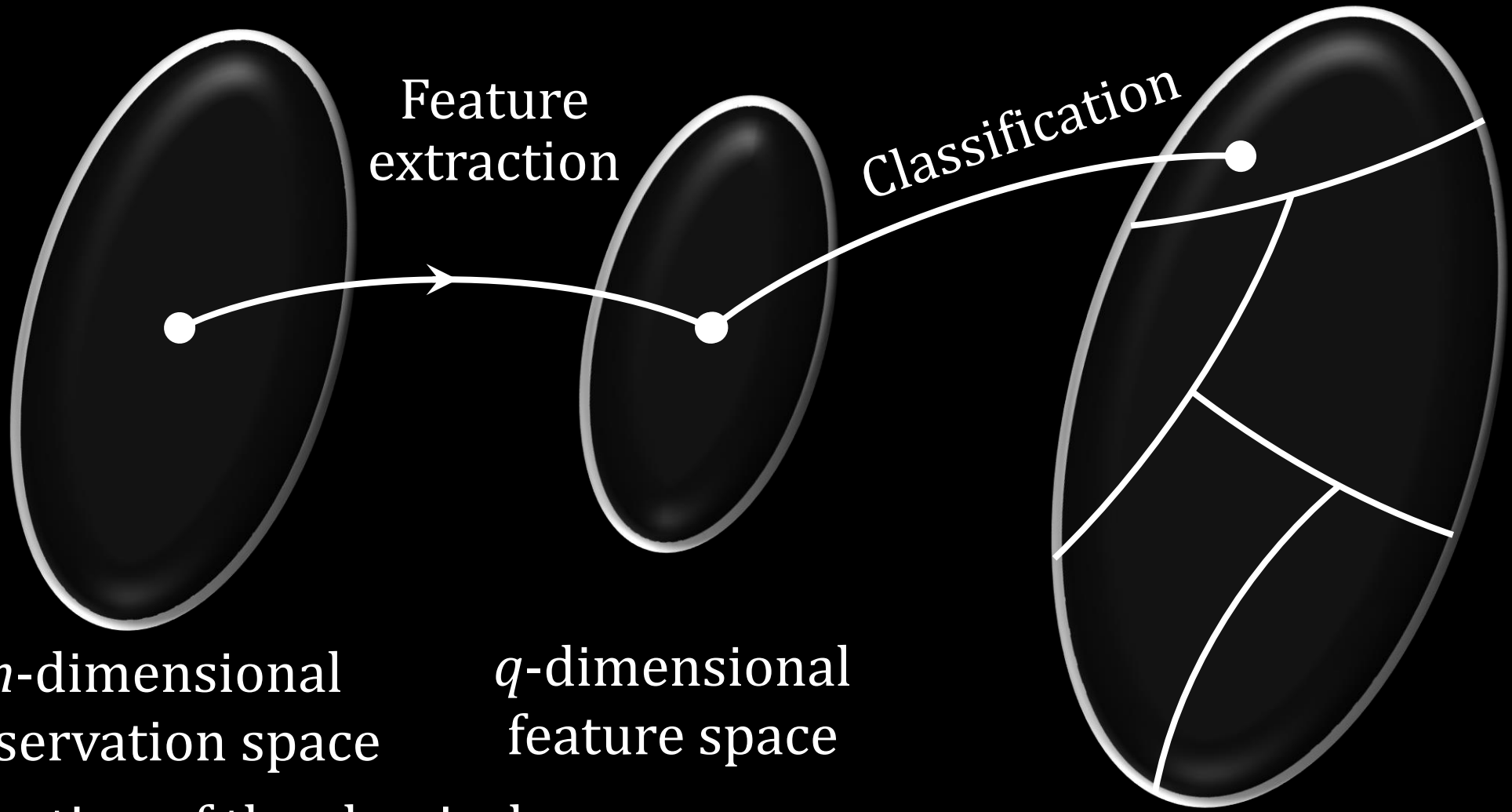


Fig. 2.10 Illustration of the classical approach to pattern classification.

2.10 Learning Tasks: Function Approximation

- Consider a nonlinear input-output mapping described by

$$\mathbf{d} = \mathbf{f}(\mathbf{x}) \tag{2.19}$$

where vector $\mathbf{x}$ and $\mathbf{d}$ **are the input and output respectively**. The vector-valued function $\mathbf{f}(.)$ is assumed to be unknown.

- Given the set of labeled examples:

$$\mathcal{J} = \{\mathbf{x}_i, d_i\}_{i=1}^{N} \tag{2.20}$$

- The requirement is to design $\mathbf{f}(.)$ such that the Eucledean distance

$$\|\mathbf{F}(.) - \mathbf{f}(.)\| < \epsilon \quad \text{for all } \mathbf{x} \tag{2.21}$$

where the approximation error ϵ is a small positive number.

2.10 Learning Tasks: Function Approximation

- The ability of a neural network based on supervised learning to approximate an unknown input-output mapping may be exploited in two important ways:

- **System identification.**
  ~ Let Eq. (2.19) describe the i/o relation of an unknown memoryless MIMO system;
  ~ using set of labeled examples in Eq. (2.20) to train a neural network as a system in Fig. 2.11.

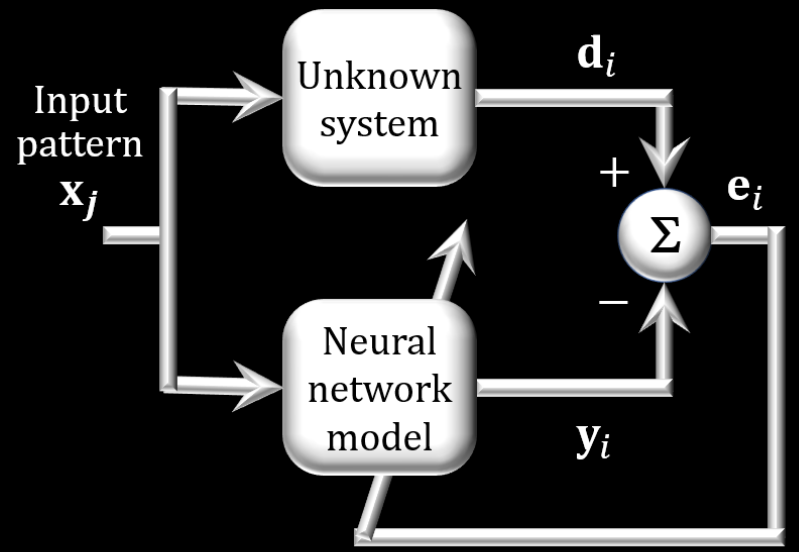


Fig. 2.11 Block diagram of system identification

- **Inverse system.**
  ~ Suppose next we are given a known memoryless MIMO system whose input-output relation is described by Eq. (2.19).
  ~ To construct an inverse system that produces the vector $\mathbf{x}$ in response to the vector $\mathbf{d}$, described by

$$\mathbf{x} = \mathbf{f}^{-1}(\mathbf{d}) \tag{2.22}$$

2.10 Learning Tasks: Function Approximation

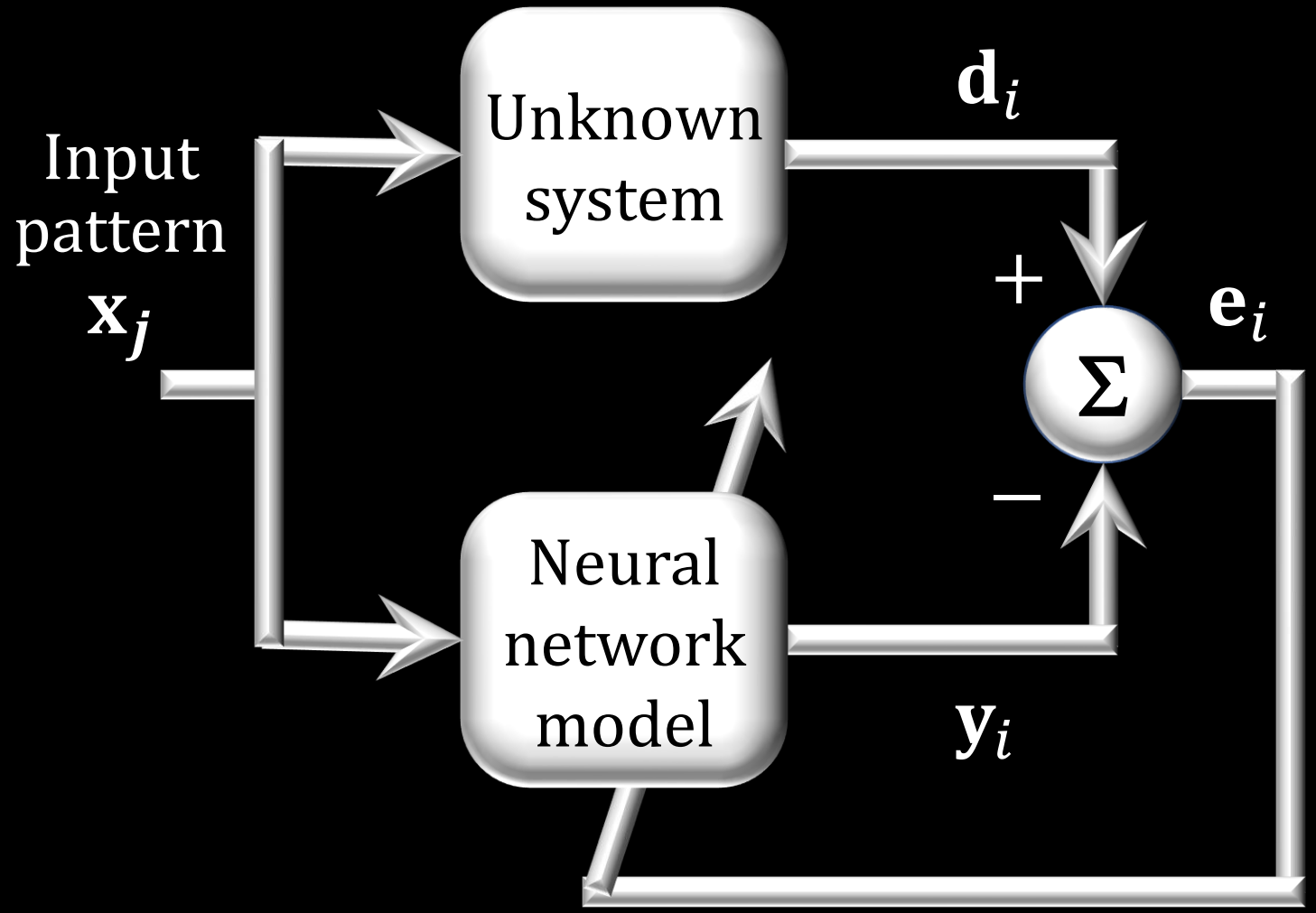


Fig. 2.11 Block diagram of system identification.

2.10 Learning Tasks: Function Approximation

- Given the set of labeled examples in Eq. (2.20), we may construct a neural network approximation of $\mathbf{f}^{-1}(.)$ by using the scheme shown in Fig. 2.12.

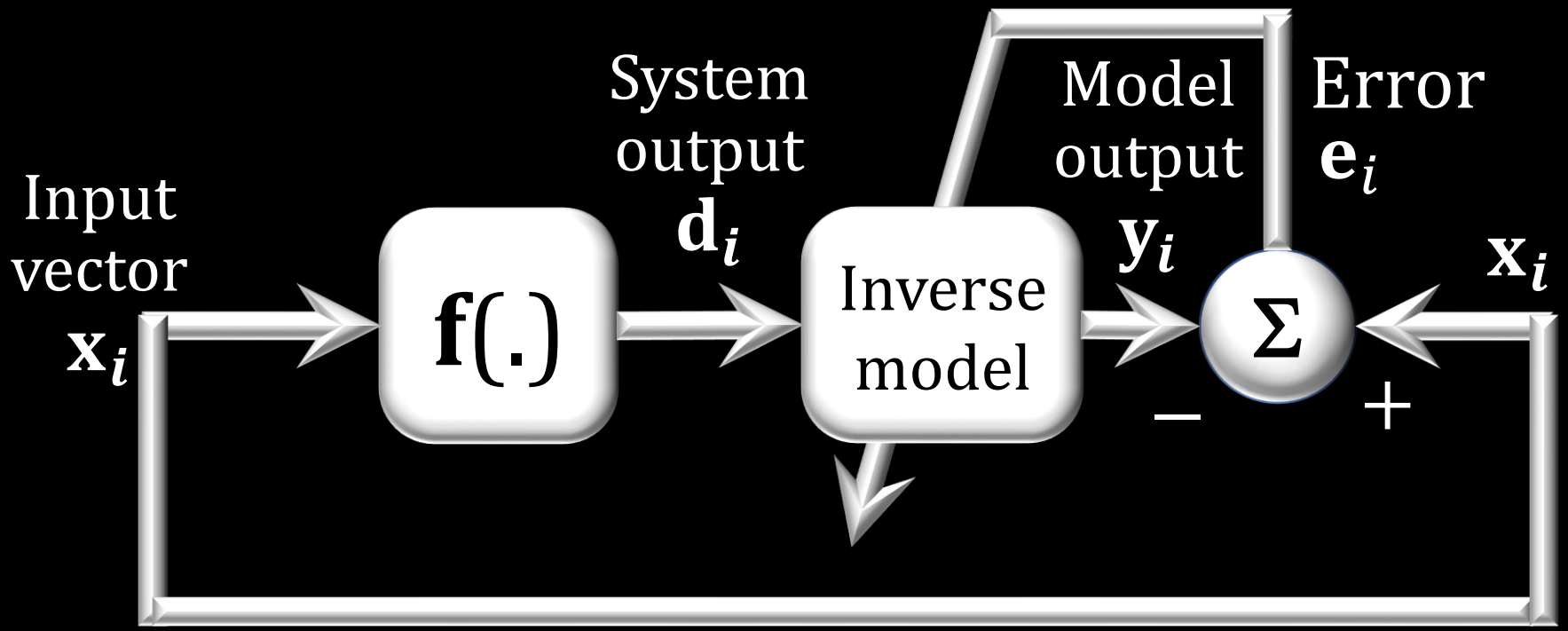
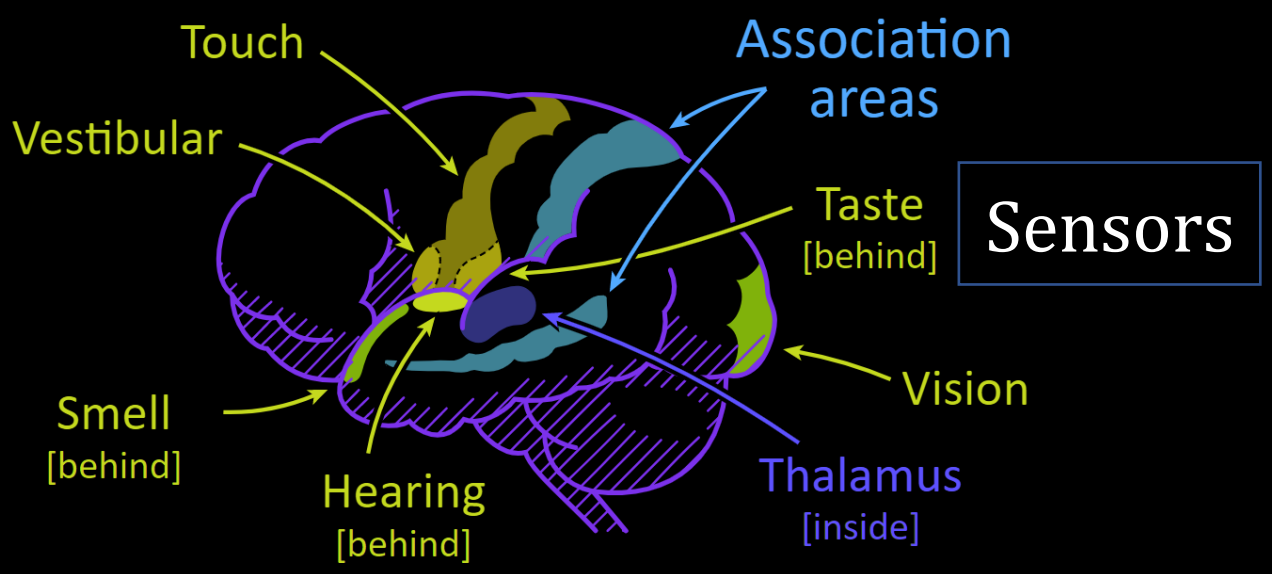


Fig. 2.12 Block diagram of inverse system modeling.

2.10 Learning Tasks: Control System

- Defining a plant is a process or critical part of a system that is to be maintained in a controlled condition.
- The control of a plant is a learning task that can be done by a neural network
- The **brain** is living proof that it is possible to build **a generalized controller** that takes full advantage of parallel distributed hardware,
  - Receives signals from many sensors (eyes, ears, skin…)
  - controls many thousands of actuators (muscle fibers) in parallel,
  - can handle nonlinearity and noise, and can optimize over a long-range planning horizon



Actuators

https://learn-genetics.b-cdn.net/senses/worktogether/brainroute.svg

2.10 Learning Tasks: Control System

- Consider the feedback control system shown in Fig. 2.13.
- The plant output **y** is fed back directly to the input and subtracted from a reference signal **d** supplied from an external source.
- The error signal **e** so produced is applied to a neural controller for the purpose of adjusting its free parameters.

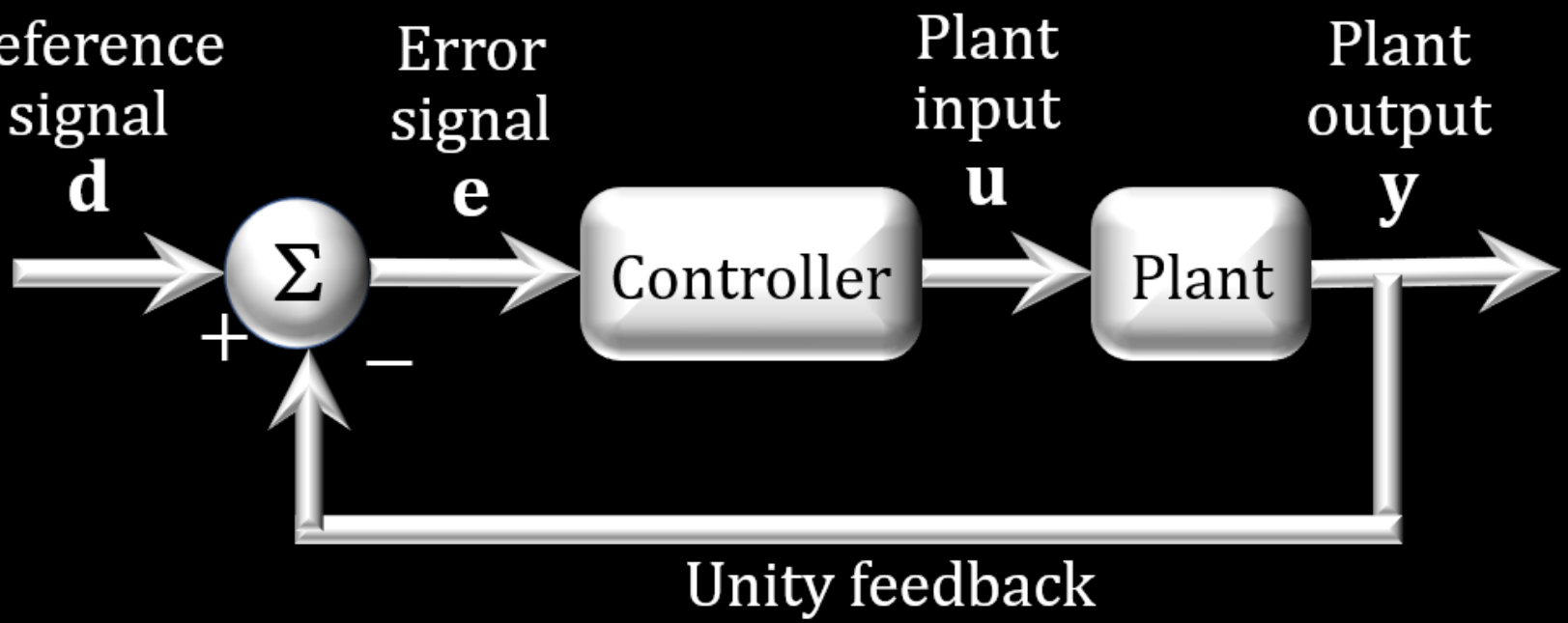


Fig. 2.13 Block diagram of feedback control system.

2.10 Learning Tasks: Control System

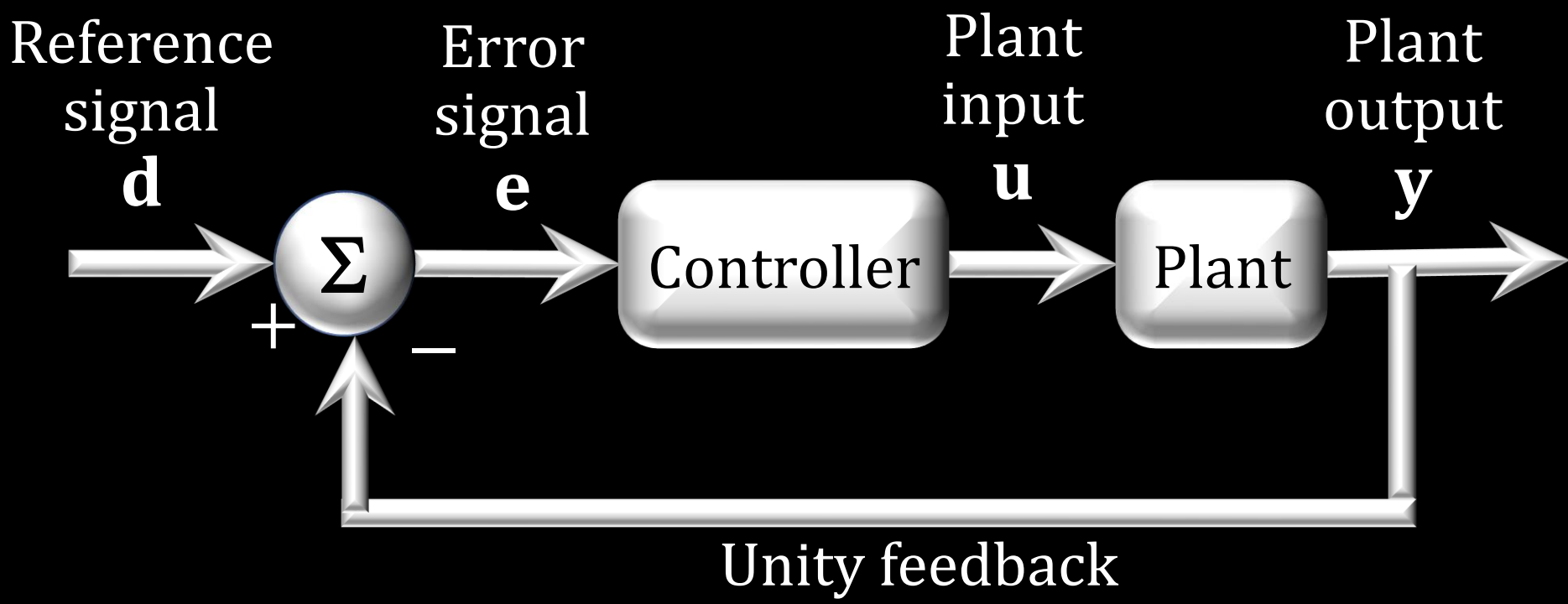


Fig. 2.13 Block diagram of feedback control system.

2.10 Learning Tasks: Control System

- To perform adjustments on the free parameters of the plant in accordance with an *error-correction learning* algorithm we need to know the Jacobian matrix

$$\mathbf{J} = \left\{\frac{\partial y_k}{\partial u_j}\right\} \tag{2.23}$$

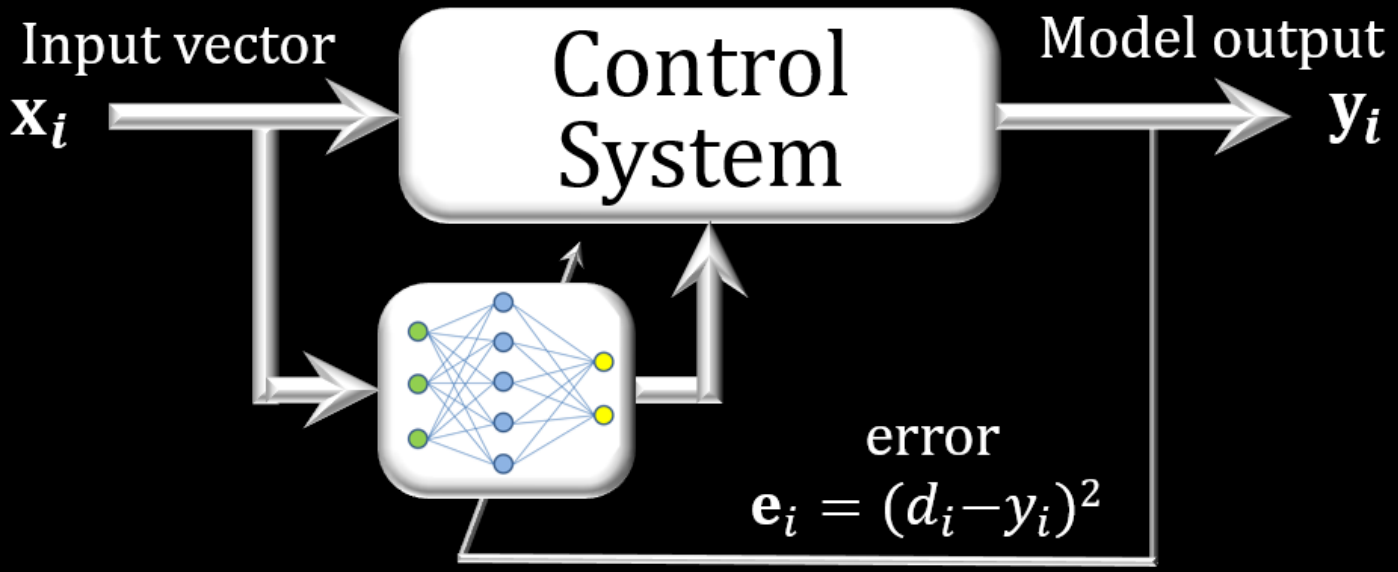
where y_k is an element of plant output $\mathbf{y}$ and u_j is an element of plant input $\mathbf{u}$.

- Two approaches may be used to account the partial derivatives $\partial y_k / \partial u_j$ which depends on the operating point of the plant

2.10 Learning Tasks: Control System

1. Indirect learning

- Using actual input-output measurements on the plant, a neural network model is first constructed to produce a copy of it.
- This model provides an estimate of the Jacobian matrix **J**.
- The partial derivatives are subsequently used in the error-correction learning algorithm for computing the adjustments to the free parameters of the neural controller.



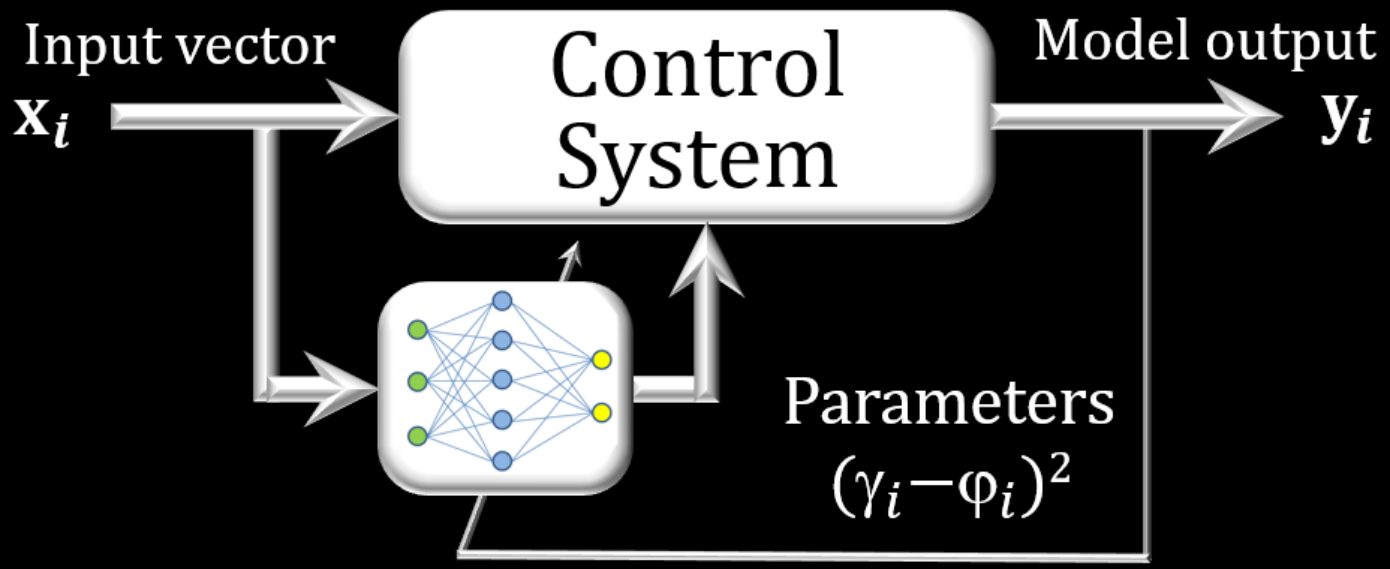
Block diagram of **indirect learning** method.

https://www.researchgate.net/publication/228962378_Development_of_a_Matlab_Toolbox_for_the_Design_of_Grey-Box_Neural_Models/figures?lo=1

2.10 Learning Tasks: Control System

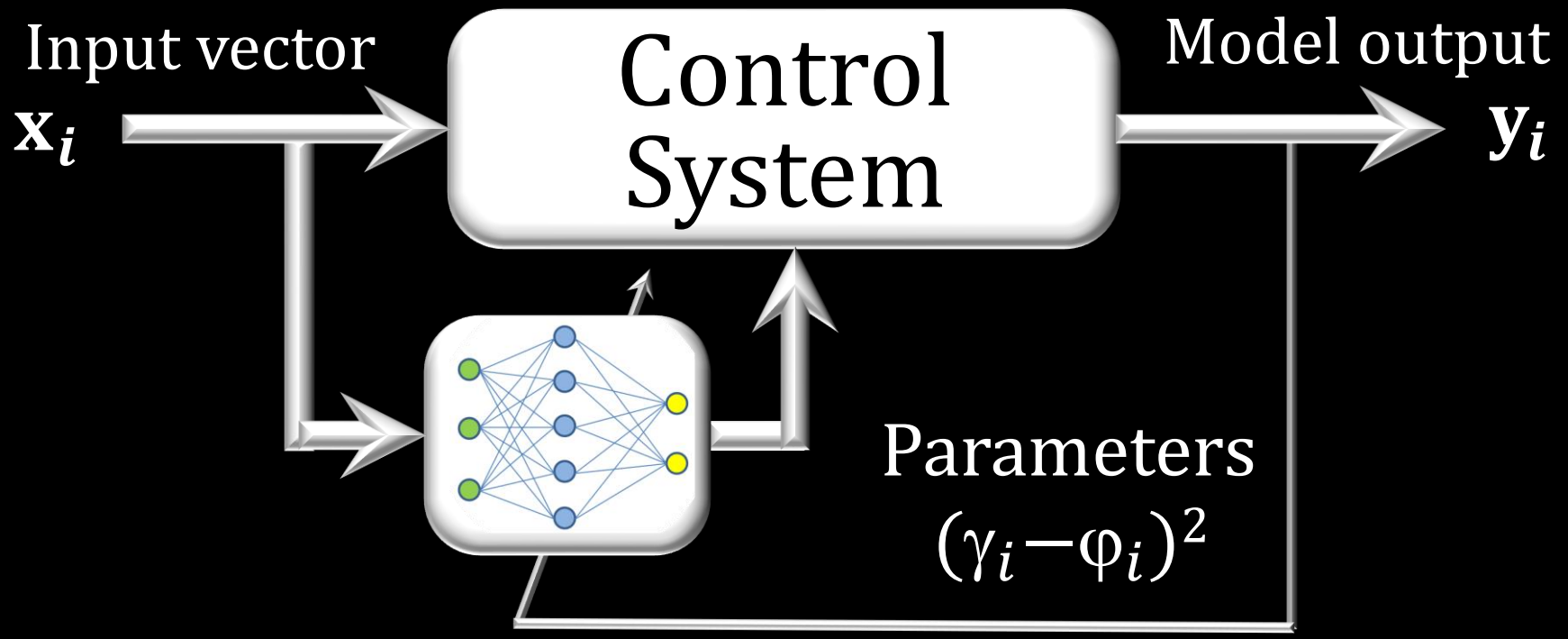
2. Direct learning

- The signs of the partial derivatives $\partial y_k / \partial u_j$ are generally known and remain constant over the dynamic range of the plant.
- We may approximate these partial derivatives by their individual signs.
- The neural controller is thereby enabled to learn the adjustments to its free parameters directly from the plant.



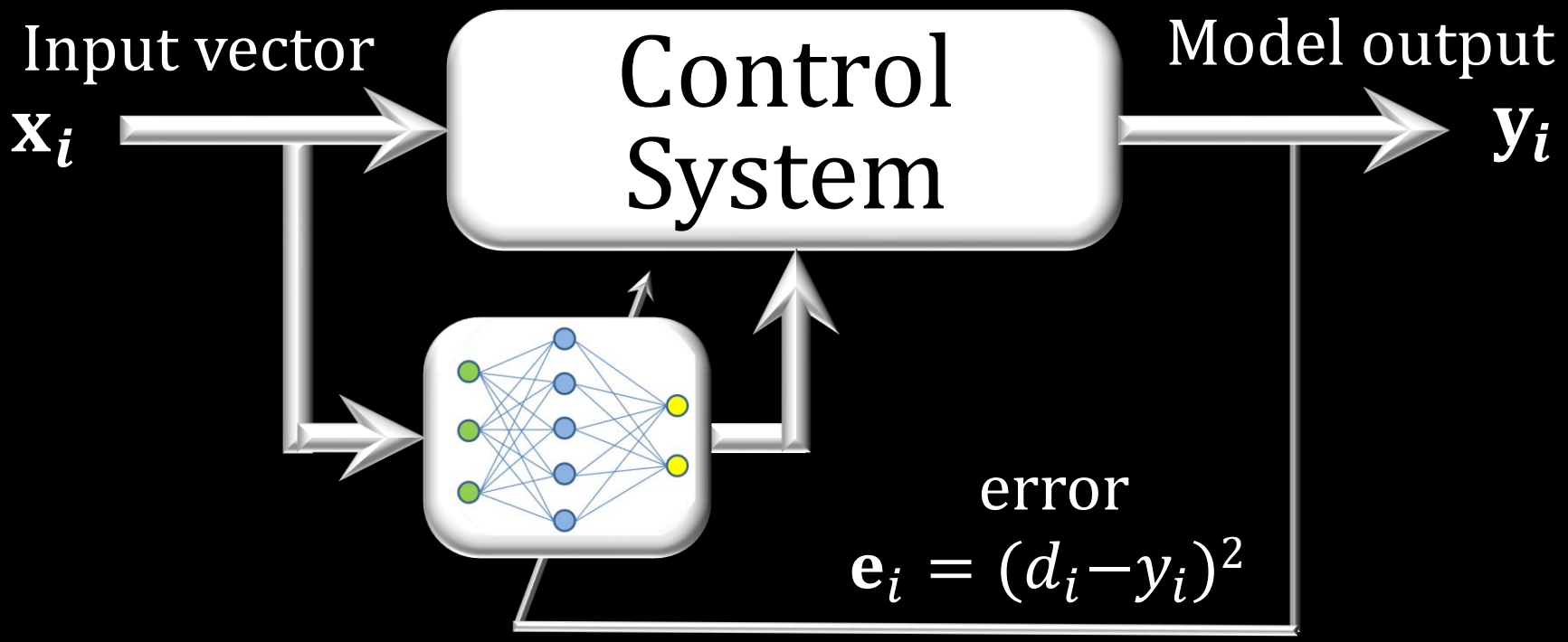
Block diagram of **direct learning** method.

2.10 Learning Tasks: Control System



Block diagram of **direct learning** method.

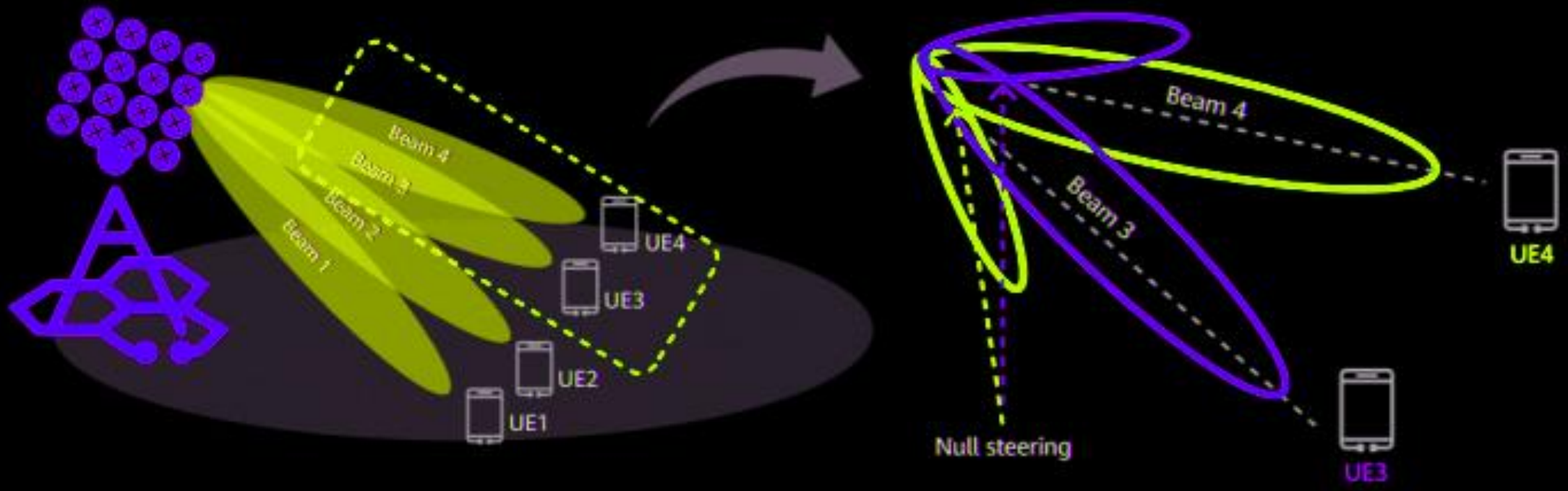
2.10 Learning Tasks: Control System



Block diagram of **indirect learning** method.

2.10 Learning Tasks: Beamforming

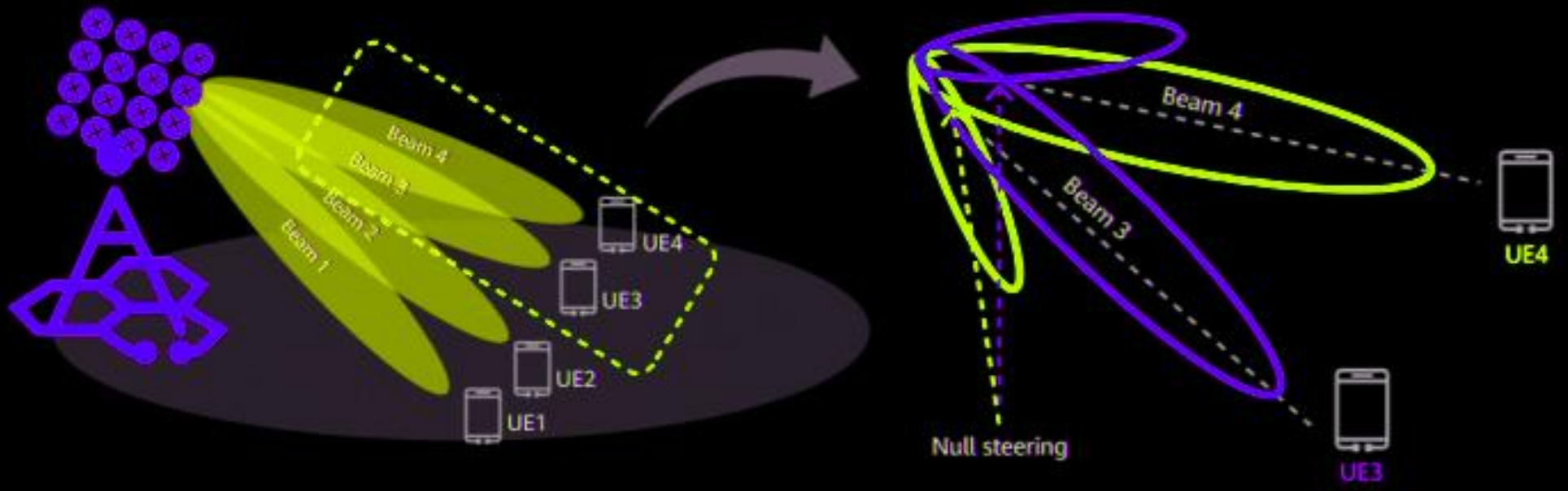
- Beamforming is a spatial form of filtering and is used to distinguish between the spatial properties of a target signal and background noise
- Beamforming is compatible with the use of a neural network, for which the cues from psychoacoustic studies of human auditory responses and the cortical layers of auditory systems of echo-locating bats



https://blog.huawei.com/2020/08/17/the-wonders-of-5g-beamforming/

2.10 Learning Tasks: Beamforming

- Beamforming is commonly used in radar and sonar systems where the primary task is to detect and track a target of interest in the combined presence of receiver noise and interfering signals (e.g., jammers).
- This task is complicated by two factors.
  1. The target signal originates from an unknown direction.
  2. There is no a priori information available on the interfering signals.



2.10 Learning Tasks: Beamforming

- To eliminate these complications, we use a Generalized Sidelobe Canceller (GSLC), the block diagram of which is shown in Fig. 2.16.
- The system consists of the following components:
  - An array of antenna elements, which provides a means of sampling the observed signal at discrete points in space.
  - A linear combiner defined by a set of fixed weights $\{w_i\}_{i=1}^{m}$ the output of which is a desired response.
  - A signal-blocking matrix $\mathbf{C}_a$, the function of which is to cancel interference that leaks through the sidelobes of the radiation pattern of the spatial filter representing the linear combiner.

2.10 Learning Tasks: Beamforming

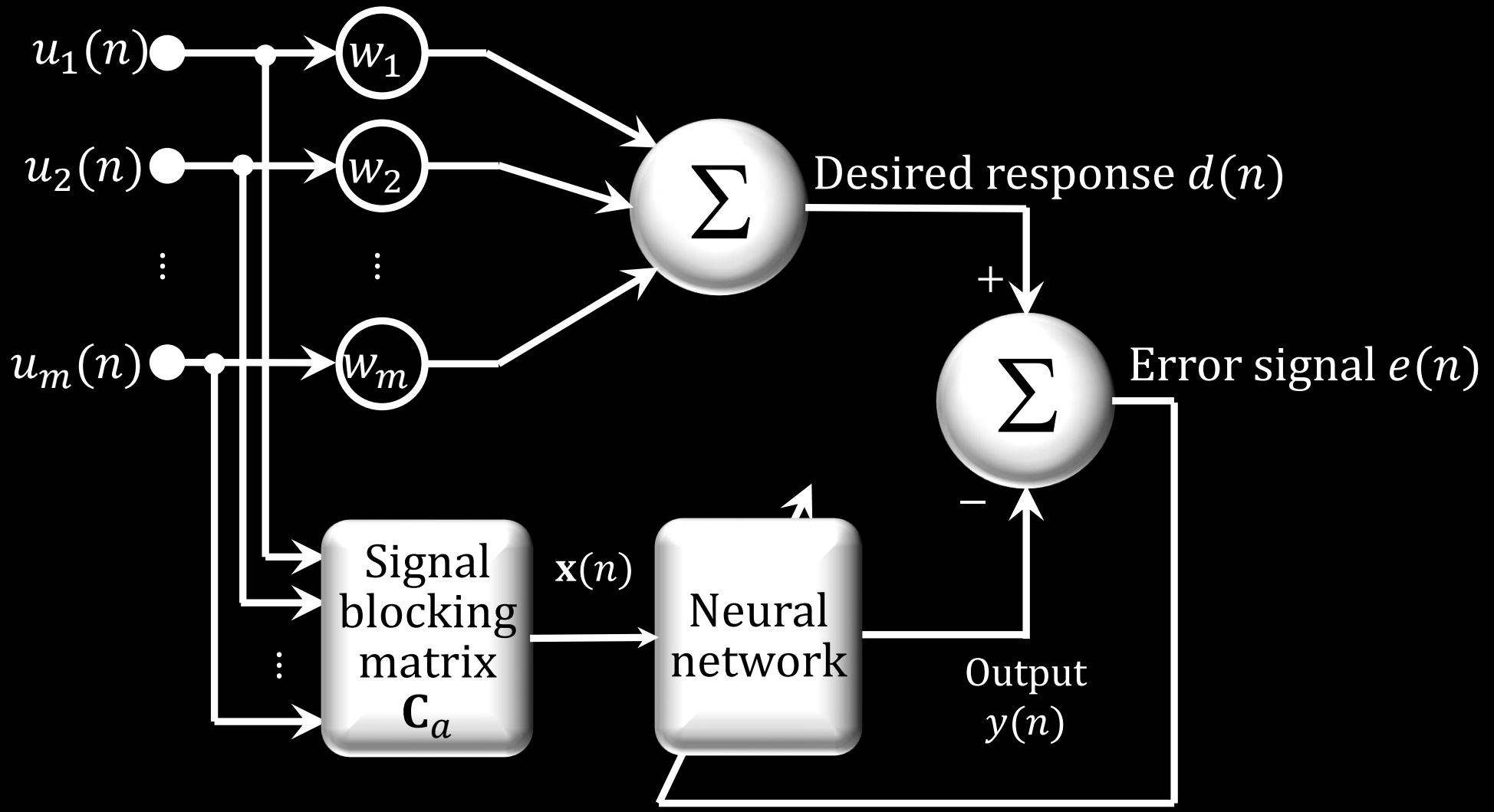
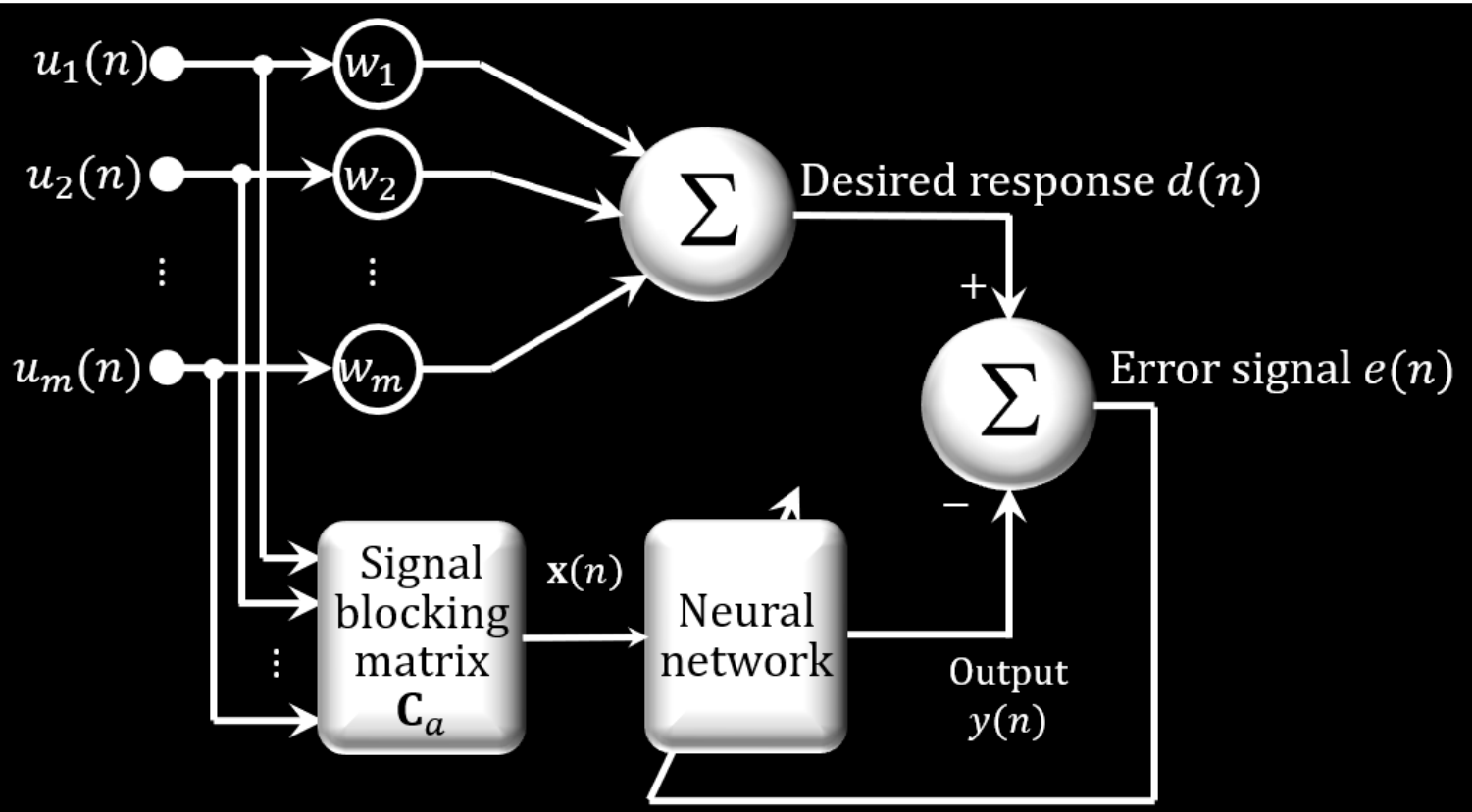


FIGURE 2.16 Block diagram of generalized sidelobe canceller.

2.10 Learning Tasks: Beamforming

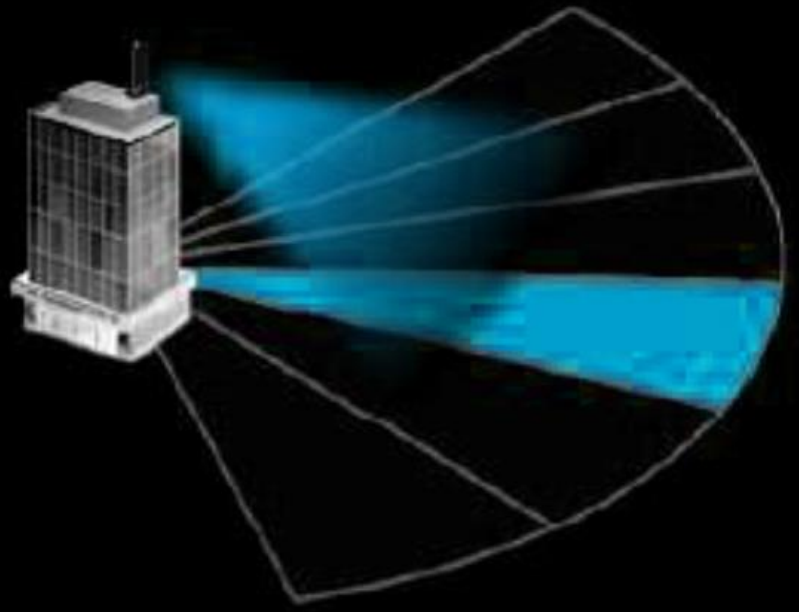
- Adjustments to the free parameters of the neural network are performed by an error-correcting learning algorithm that operates on the error signal $e(n)$.



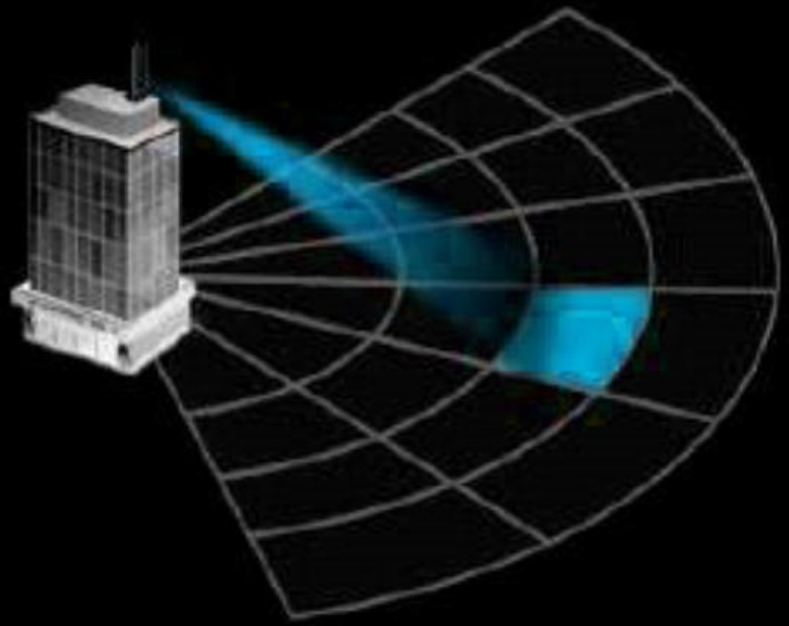
- Thus, the GSLC operates under the supervision of the linear combiner that assumes the role of a "teacher."
- A beamformer that uses a neural network for learning is called a neural beamformer or neuro-beamformer.

2.10 Learning Tasks: Beamforming

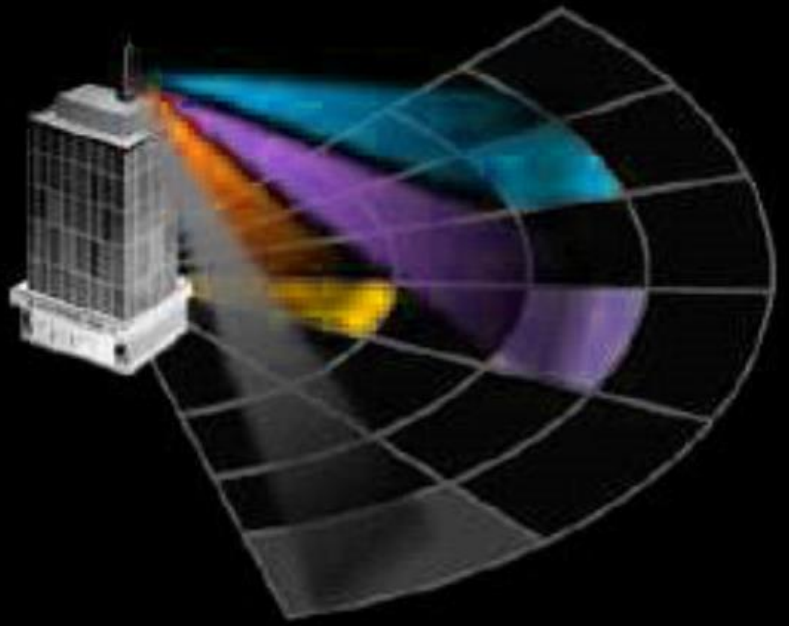
- Thus, the GSLC operates under the supervision of the linear combiner that assumes the role of a "teacher."
- A beamformer that uses a neural network for learning is called a neural beamformer or neuro-beamformer.



Conventional beamforming in horizontal direction



3D beamforming for a single user equipment

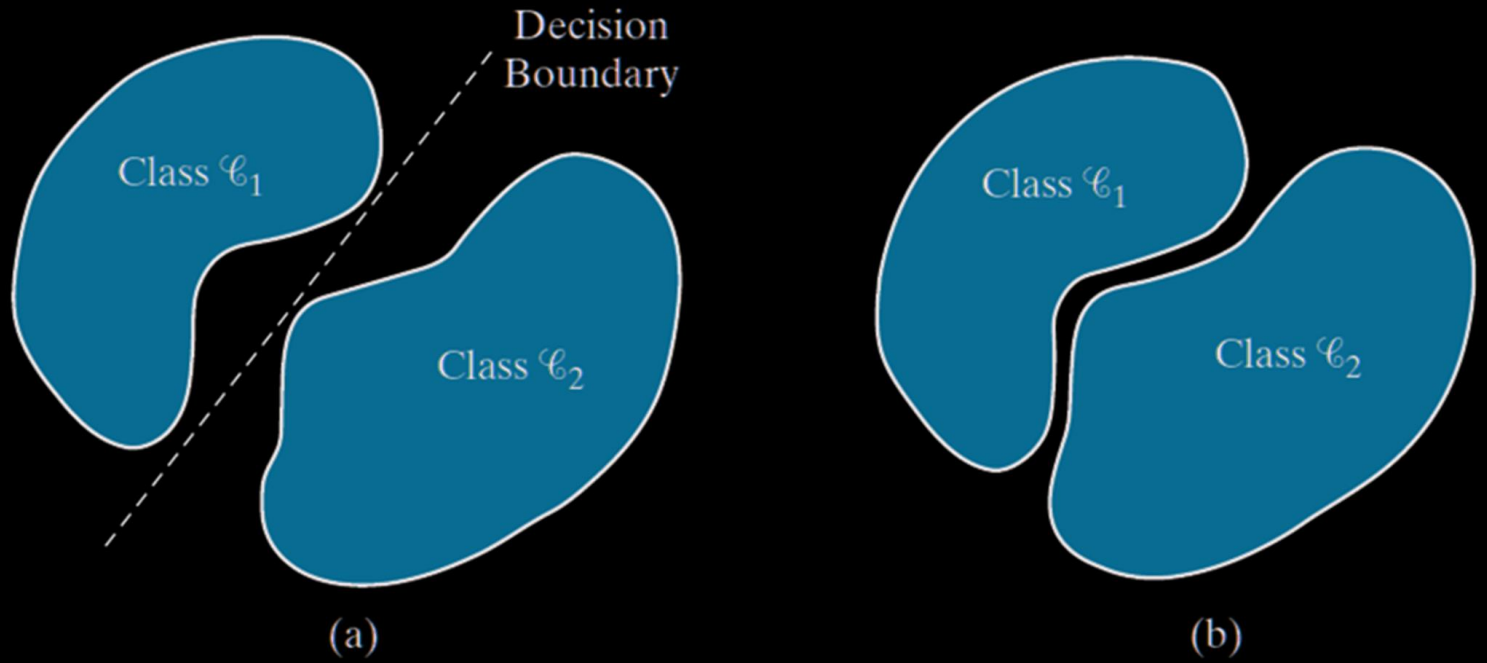


3D beamforming for multiple user equipments

https://www.mdpi.com/2073-8994/11/10/1260

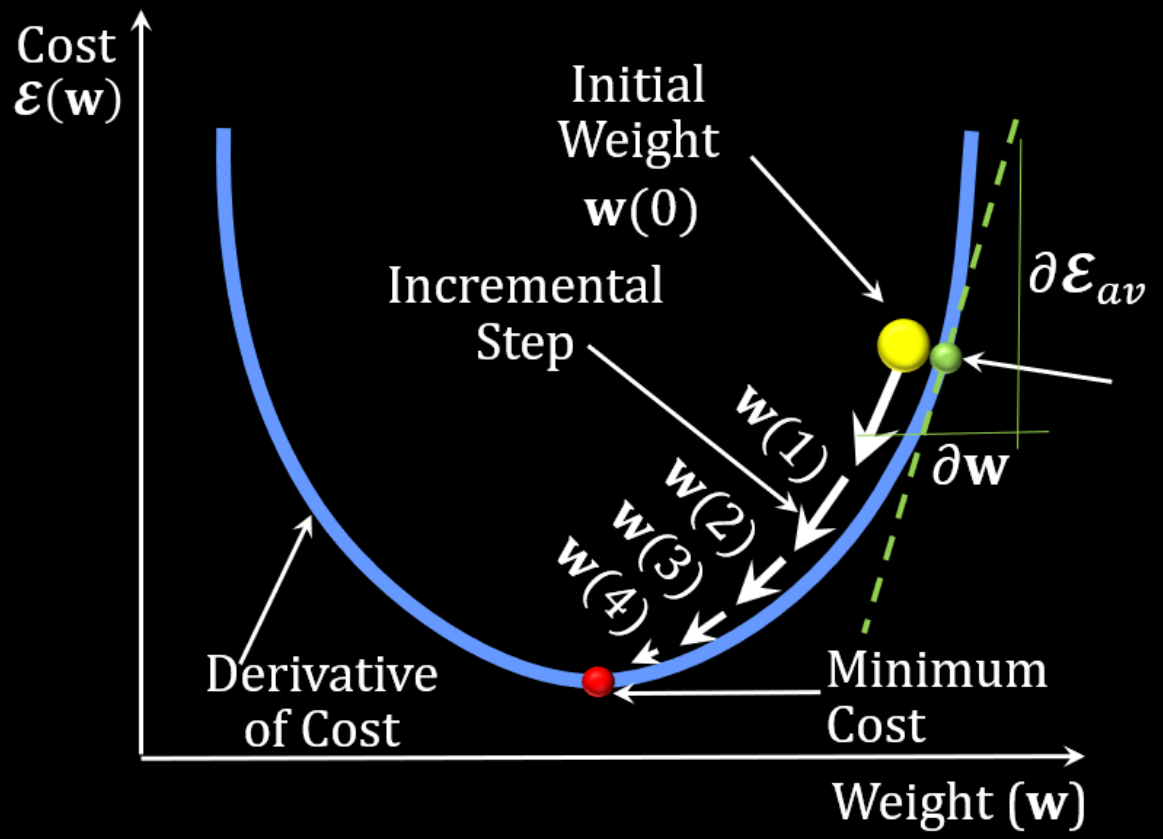
2.10 Learning Tasks: Classification

- Classification: This type of network was trained to differentiate between male and female faces.
- It is actually very difficult to create an algorithm to do so yet an ANN has been shown to have near-human capacity to do so.
- Thus, the GSLC operates under the supervision of the linear combiner that assumes the role of a "teacher."



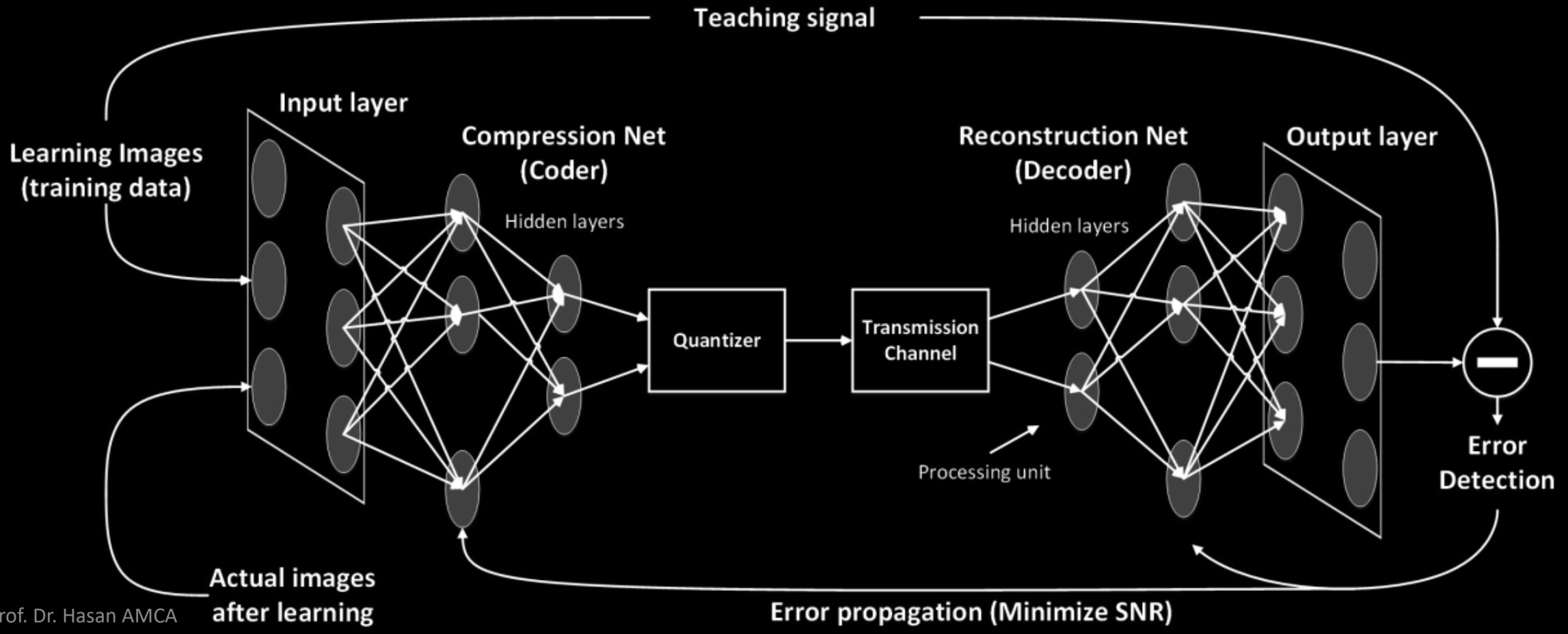
2.10 Learning Tasks: Optimization

- It is extremely difficult to find algorithms for solving optimization problems.
- A famous optimization problem is the Travelling Salesman Problem in which a salesman must travel to each of a number of cities, visiting each one once and only once in an optimal (i.e. least distance or least cost) route.
- Several types of neural networks converges to a 'goad-enough' solutions to this problem
- i.e. solutions which may not be globally optimal but can be shown to be close to the global optimum for any given set of parameters



2.10 Learning Tasks: Data Compression

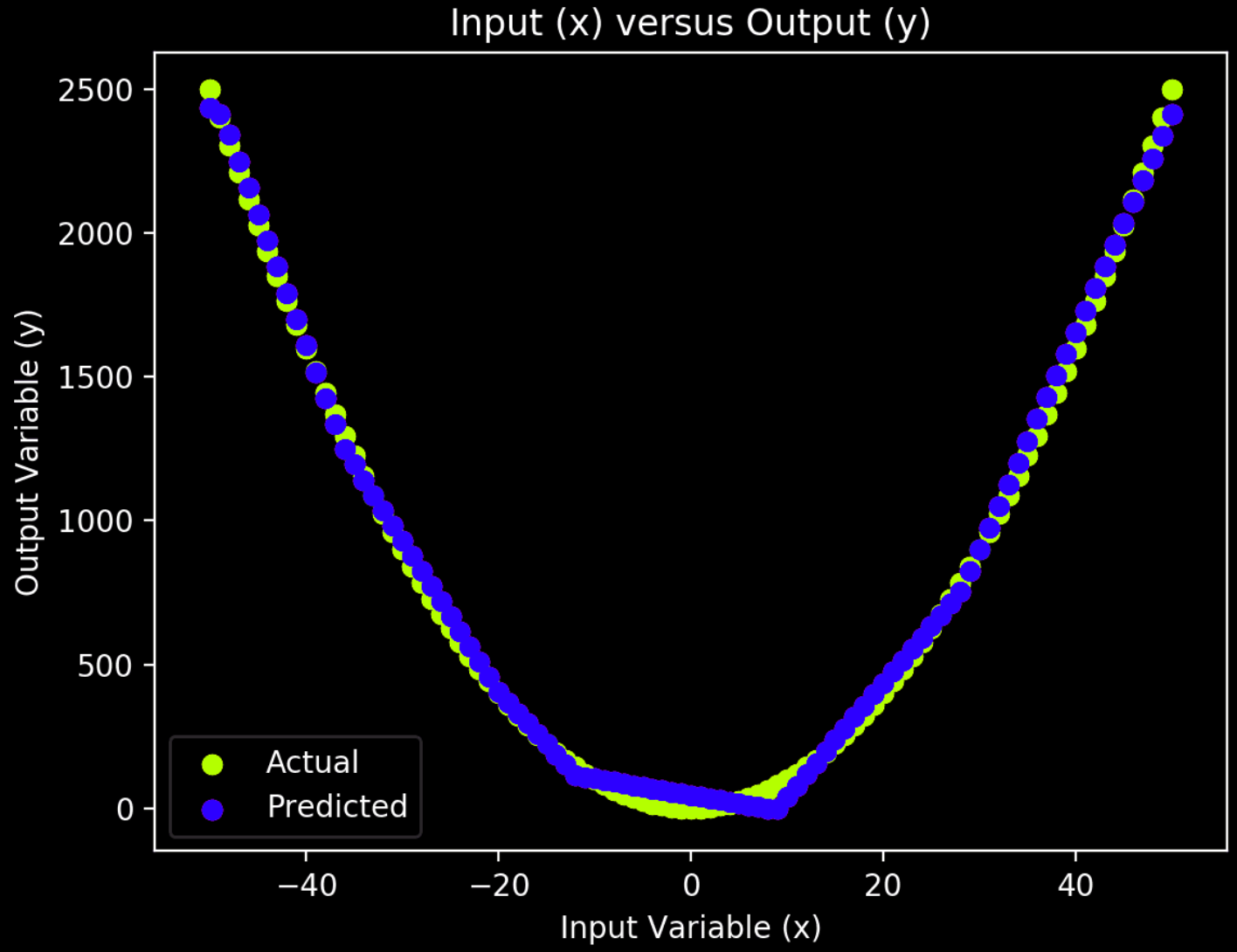
- There are many ANNs which have been shown to be able to represent input data in a compressed format losing as little of the information as possible
- e.g. in image compression, a great deal of information given by a pixel to pixel-representation of data is redundant, and a more compact depiction of image can be found



https://arxiv.org/pdf/1904.03567.pdf

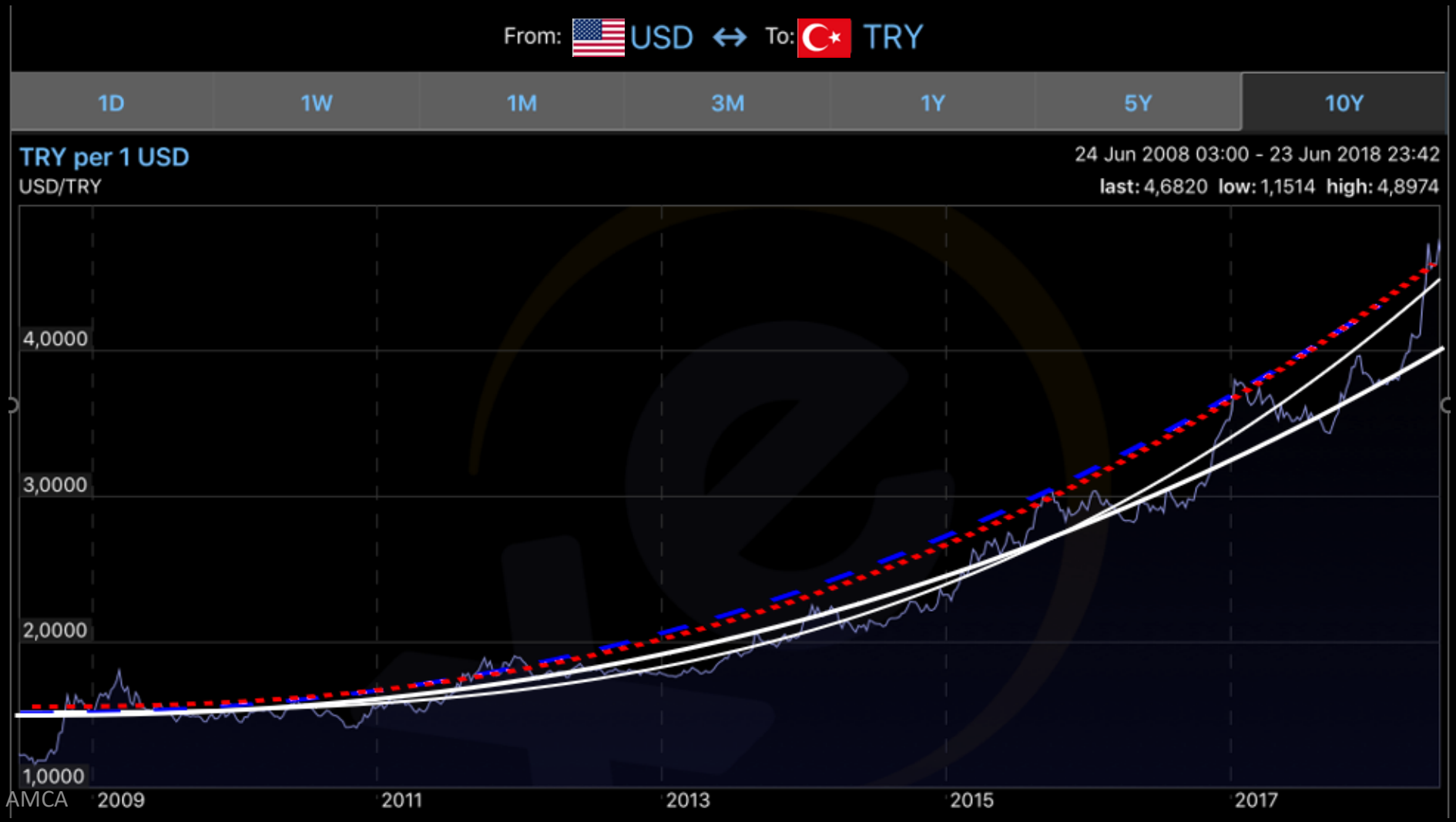
2.10 Learning Tasks: Approximation

- A neural network can be trained to approximate the mapping so that a future input will give approximately the correct answer
- i.e. The answer which the mapping should give.



2.10 Learning Tasks: Prediction

- Prediction task may start with a given set of previous examples from a time series, such as a set of closing prices for FTSE, to predict next (future) sample



2.10 Learning Tasks: Filtering

- The term *filter* is a device or algorithm used to extract desired information of interest from a set of noisy data which may arise from variety of sources, e.g.
  - noisy sensors or transmission through a noisy communication channel.
- We may use a filter to perform three basic information processing tasks:

1. **Filtering**. This task refers to the extraction of information about a quantity of interest at discrete time n by using data measured up to and including time n.
2. **Smoothing**. Differs from filtering in that information about the quantity of interest need not be available at time n, and data measured later than time n can be used in obtaining this information.
3. **Prediction**. The aim of prediction (forecasting) is to derive information about what the quantity of interest will be like at some time $n + n_0$ in the future, for some $n_0 > 0$, by using data measured up to and including time n.

2.10 Learning Tasks: Filtering

- *Example*: Humans have a remarkable ability to focus on a speaker in the noisy environment of a cocktail party.
- In the context of (artificial) neural networks, a similar filtering problem arises under the umbrella of blind signal separation.
- To formulate the blind signal separation, consider a set of unknown source signals $\{s_i(n)\}_{i=1}^{m}$ that are mutually independent of each other.
- These signals are linearly mixed by an unknown *sensor* to produce the *m*-by-1 observation vector as shown in Fig. 2.14.

$$\mathbf{x}(n) = \mathbf{A}\mathbf{u}(n) \tag{2.24}$$

where

$$\mathbf{u}(n) = [u_1(n), u_2(n), \ldots u_m(n)]^T \tag{2.25}$$

$$\mathbf{x}(n) = [x_1(n), x_2(n), \ldots x_m(n)]^T \tag{2.26}$$

- and $\mathbf{A}$ is an unknown nonsingular mixing matrix of dimensions *m*-by-*m*.

2.10 Learning Tasks: Filtering

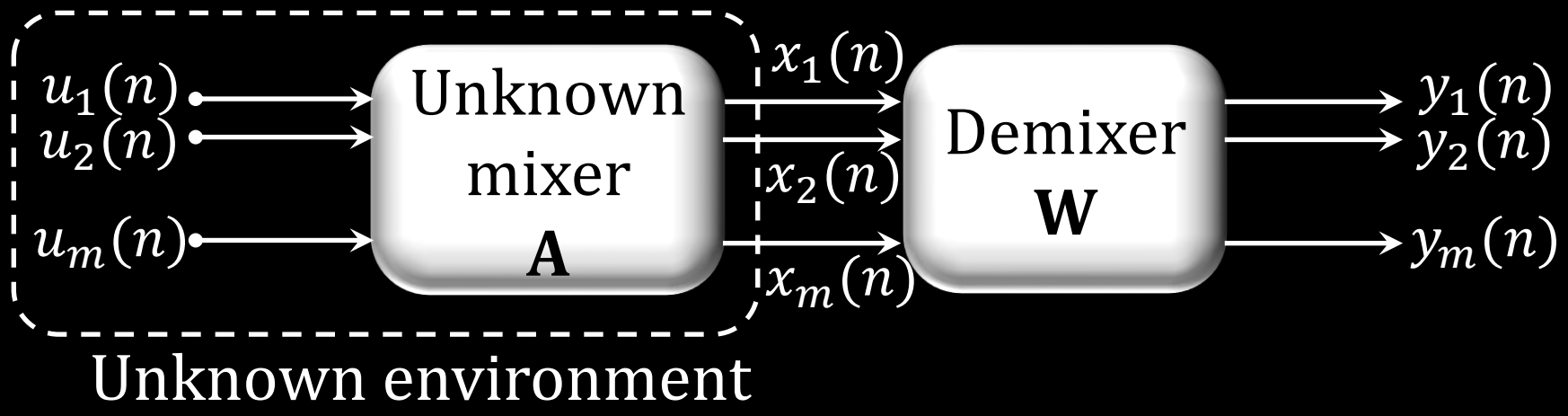


Fig. 2.14 Block diagram of blind source separation.

Goal: Given the observation vector $x(n)$, the requirement is to recover the original signals $u_1(n), u_2(n), \ldots u_m(n)$ in an unsupervised manner.

2.10 Learning Tasks: Filtering

- The requirement is to predict the present value $x(n)$ of a process, given past values of the process that are uniformly spaced in time as shown by $x(n - T)$, $x(n - 2T)$, .. . , $x(n - mT)$, where T is the sampling period and m is the prediction order.
- Prediction may be solved by using error-correction learning in an unsupervised manner since the training examples are drawn directly from the process itself, as depicted in Fig. 2.15, where $x(n)$ serves the purpose of desired response.

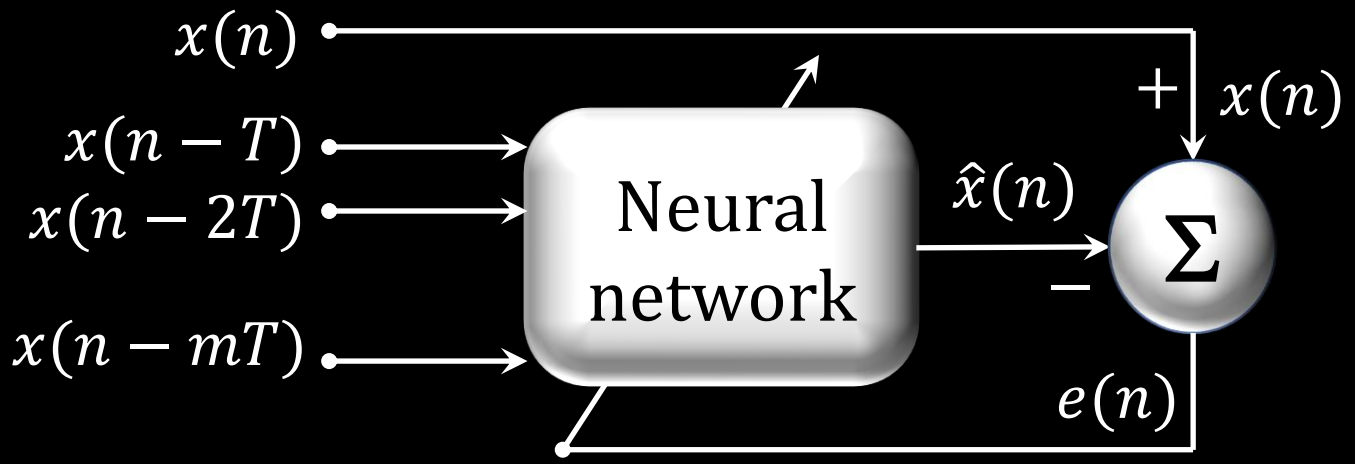


Fig. 2.15 Block diagram of nonlinear prediction.

2.10 Learning Tasks: Filtering

- The error signal $e(n)$ defined as the difference between $x(n)$ and $\hat{x}(n)$ is used to adjust the free parameters of the neural network.
- Prediction may be viewed as a form of model building in the sense that
  - the smaller the prediction error, the better the network serves as a model of the underlying physical process responsible for generating the data.

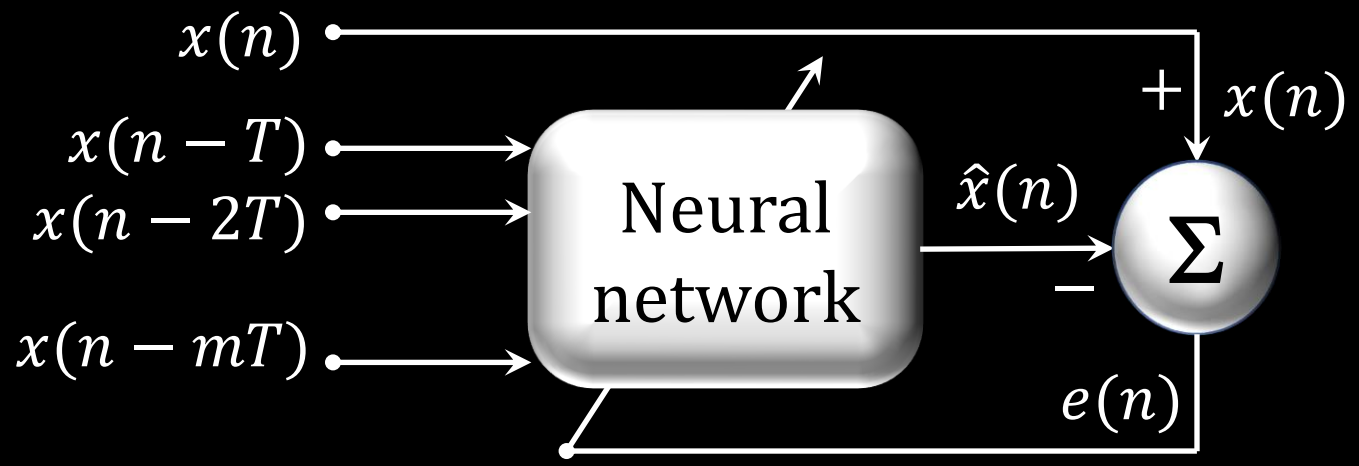


Fig. 2.15 Block diagram of nonlinear prediction.