

Learning Processes

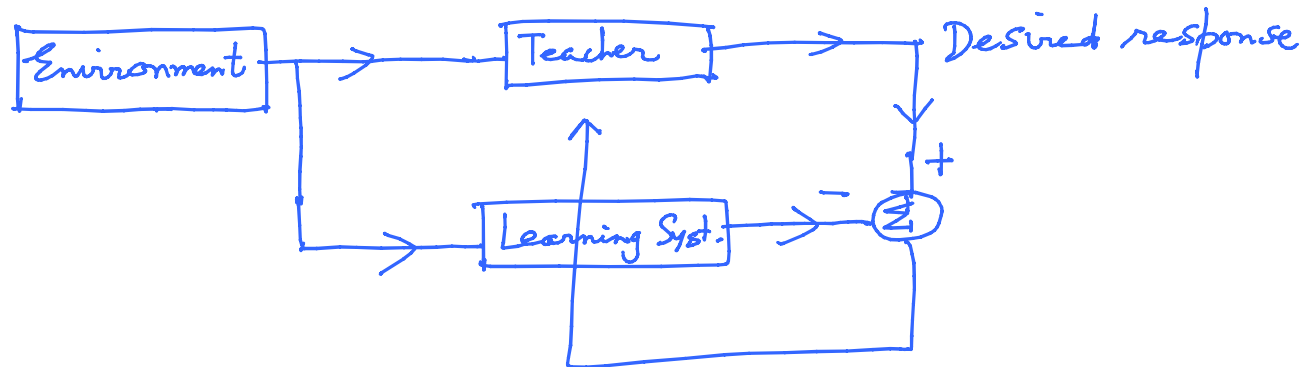
(a) Learning with a teacher

(b) Learning without a teacher

(i) Unsupervised learning

(ii) Reinforcement learning

(a)

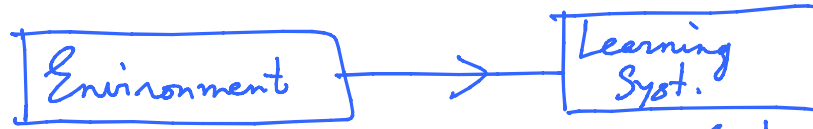


Knowledge of the expert / teacher about the environment is directly transferred to the neural n/w via training. This is also a basis for 'error correction learning'

(b) Learning w/o a teacher

Ex: Social etiquettes.

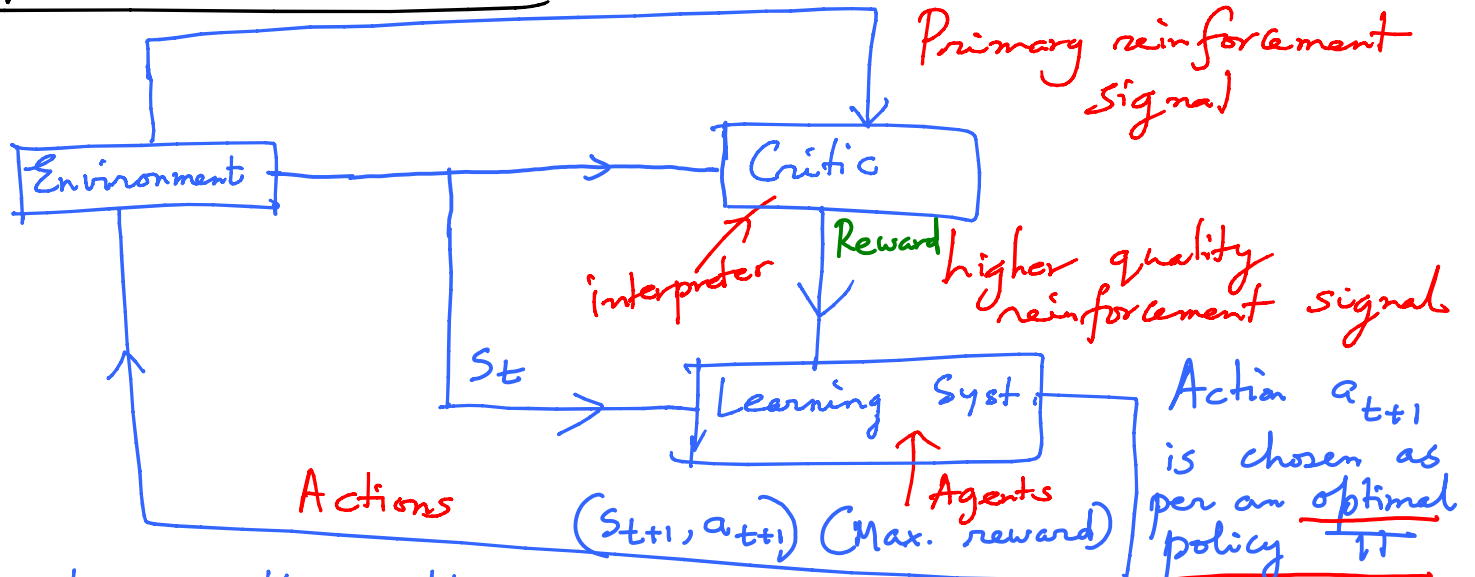
(i) Unsupervised learning



- (a) Clustering algos (K-means)
- (b) Self Organizing Map (competitive learning)

(ii)

Reinforcement learning



Learn the I/O mapping through continued interactions with the environment.

Learning Tasks

- 1) Pattern association
 - 2) Pattern recognition
 - 3) Functional Approximation
 - 4) Control
- ⋮

Pattern Association

An associative memory is a brain like distributed memory
Learn through associations

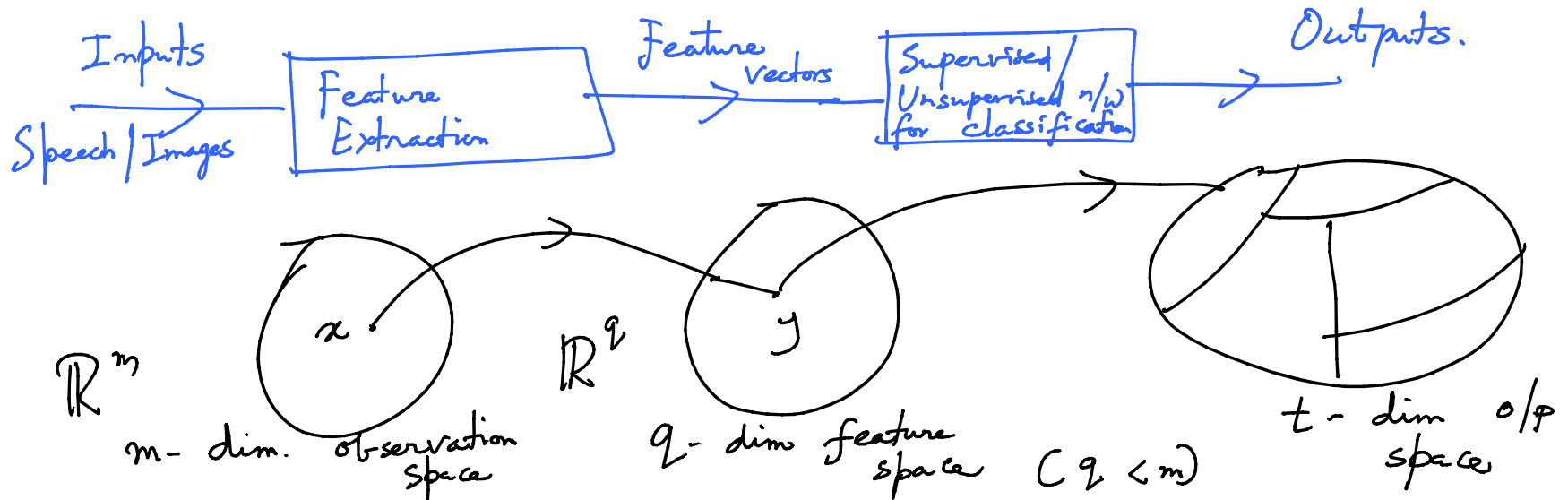
- | | | | |
|-----|--------------------|---|--------------|
| (a) | Auto association | : | Unsupervised |
| (b) | Hetero association | : | Supervised |

Storage Phase: $f : \underline{x}_k$ (Stimulus) \longrightarrow \underline{y}_k (memorized pattern)

Recall Phase: $\arg \max_{\underline{y}_k} \phi(\underline{x}, \mathcal{N})$
 (noisy stimulus) \uparrow network $\left(\mathcal{N} \text{ has } \underline{x}_k \text{ s embedded within} \right)$

Pattern Recognition

It is a process where by a received pattern / signal is assigned to one of the prescribed # of classes.



Functional Approximation

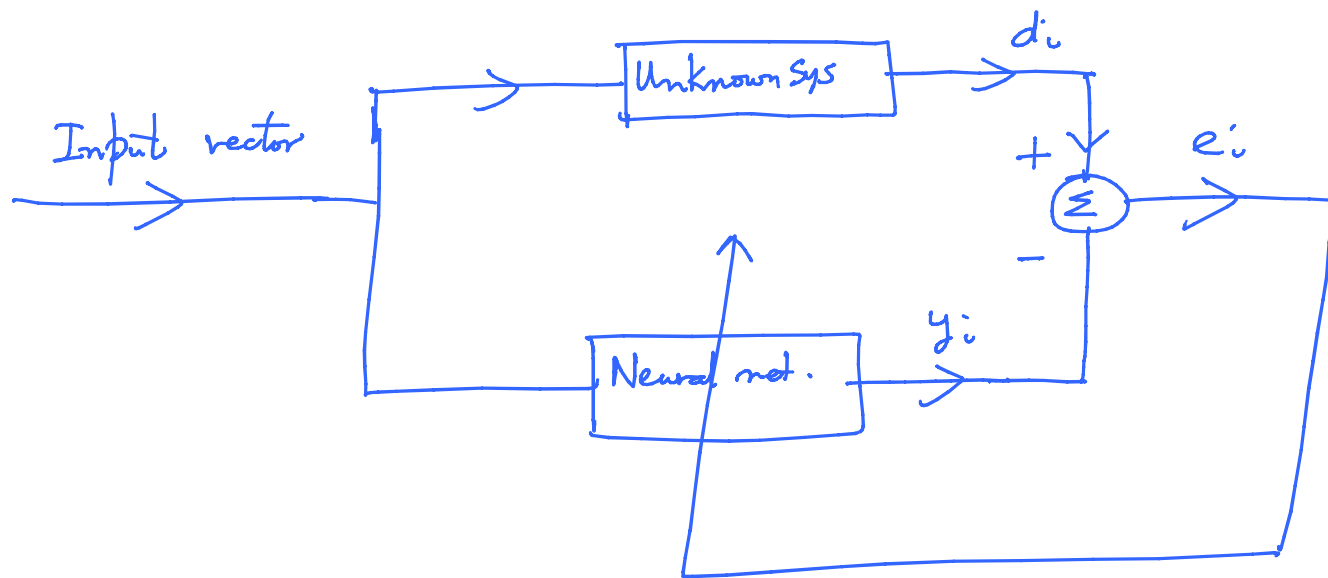
$$\underline{d} = f(\underline{x})$$

↖ ↙
o/p input

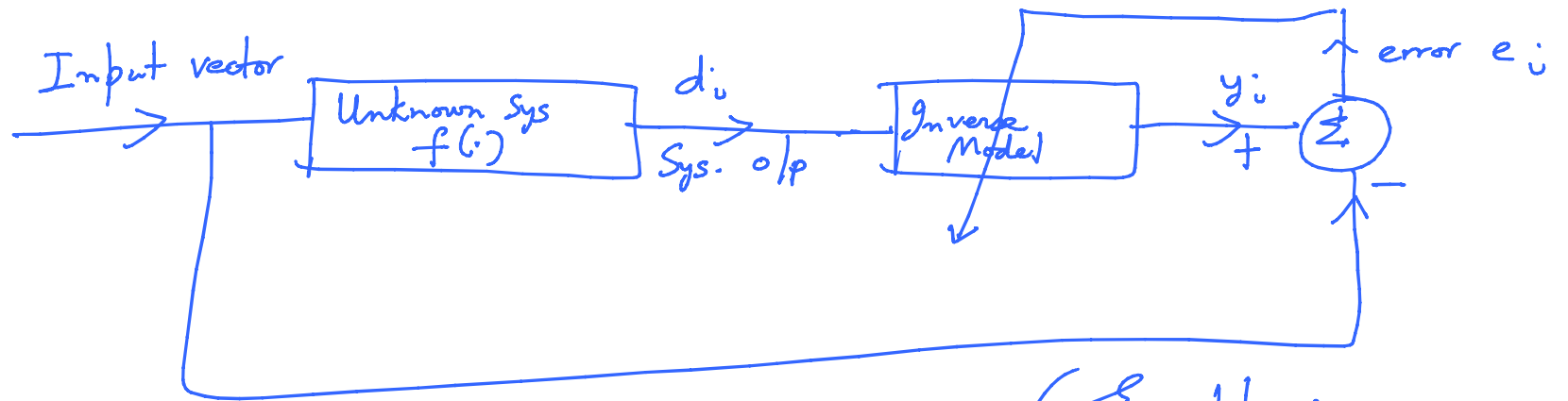
$f(\cdot)$ is unknown

Goal: Design a neural net. / it creates a mapping
 $\|F(\underline{x}) - f(\underline{x})\| < \epsilon \quad \forall \underline{x}$

Examples: (a) System Identification
(b) Inverse Modeling



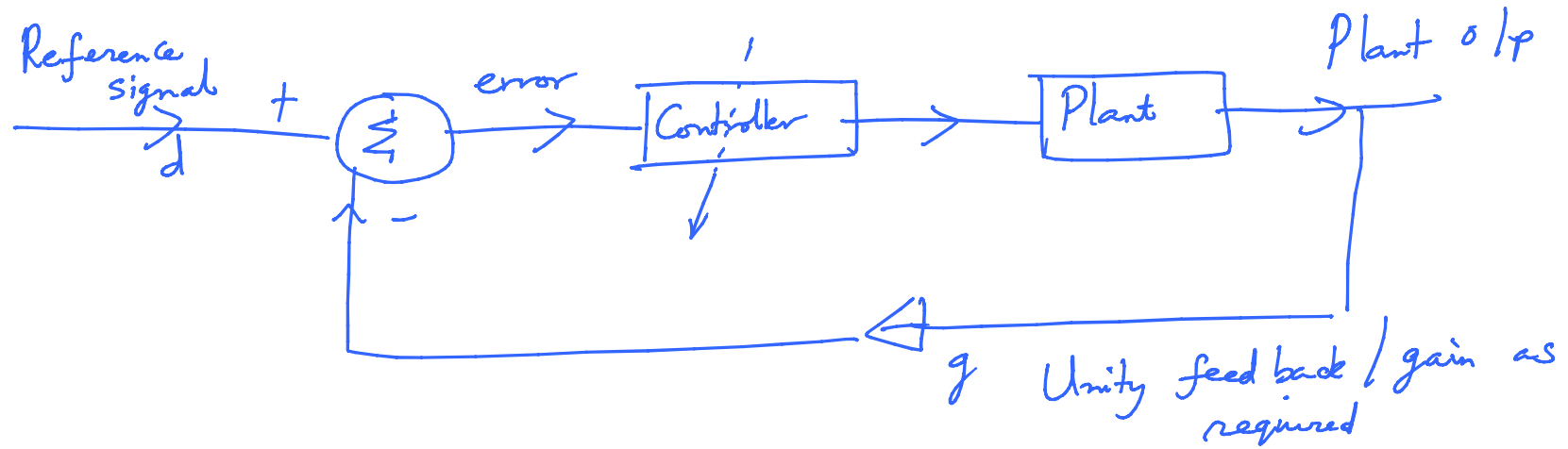
System Identification Block Diagram Schematic



Inverse Modeling

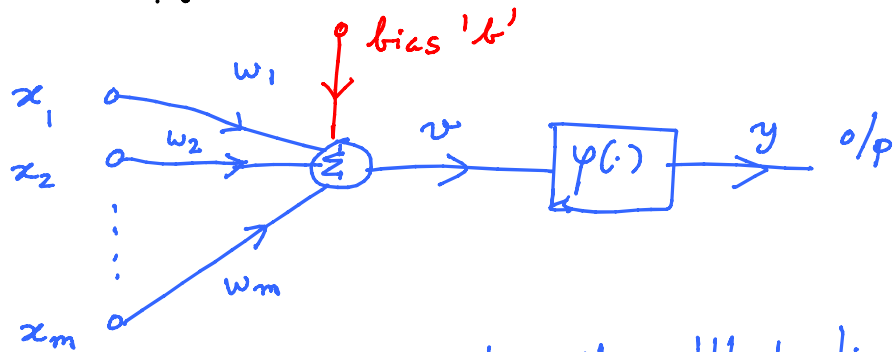
(Examples:
Channel equalization)

Control of a plant



PERCEPTRON

History: It is the first 'algorithmically' described neural n/w.
Rosenblatt (1943)



$$v = \sum_{i=1}^m w_i x_i + b$$

Inputs

GOAL:

Classify the externally applied stimulus into 2 classes c_1 and c_2 (2 class problem)

Decision Rule:

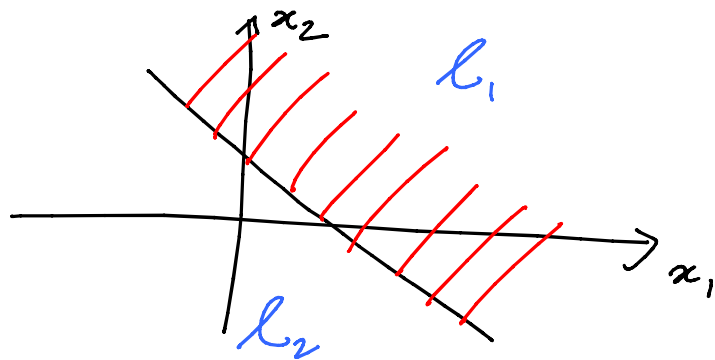
$$y = \begin{cases} +1 & \underline{x} \in \mathcal{L}_1 \\ -1 & \underline{x} \in \mathcal{L}_2 \end{cases}$$

Now, $\sum_{i=1}^m w_i x_i + b = 0$ is the equation of a hyperplane

We need to obtain the equation of the hyperplane

$$\underline{w}^T \underline{x} + b = 0$$

Illustration
in \mathbb{R}^2



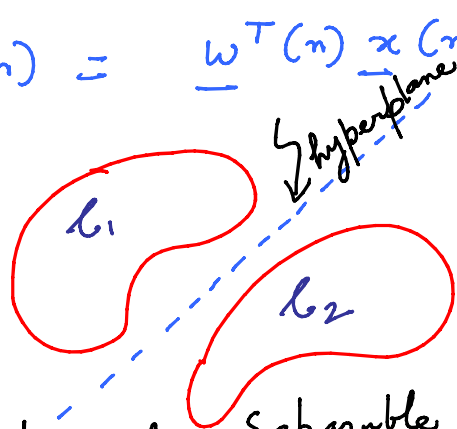
Perceptron

We form $\underline{x}(n) = [+1 \quad x_1(n) \quad x_2(n) \quad \dots \quad x_m(n)]^T$

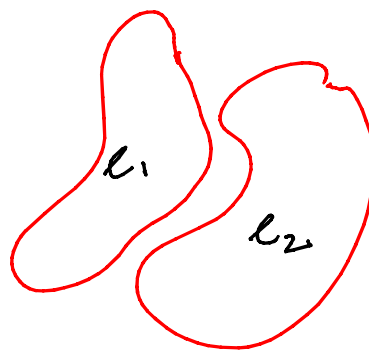
$\underline{w}(n) = [b \quad w_1(n) \quad w_2(n) \quad \dots \quad w_m(n)]^T$

These are $(m+1) \times 1$ vectors

$$v(n) = \underline{w}^T(n) \underline{x}(n)$$



(a) Linearly Separable



(b) Not linearly separable

(Classes do not overlap)

Suppose we consider inputs that come from a linearly separable class,

$$\begin{aligned} \underline{w}_{\text{opt}}^T \underline{x} > 0 &\implies \underline{x} \in \mathcal{C}_1 \quad \forall \underline{x} \\ \underline{w}_{\text{opt}}^T \underline{x} \leq 0 &\implies \underline{x} \in \mathcal{C}_2 \quad \forall \underline{x} \end{aligned}$$

Algorithm towards getting a solution to \underline{w} .

1)

If the n^{th} data point from the training set, i.e., $\underline{x}^{(n)}$ is correctly classified by the weight $\underline{w}^{(n)}$ then no correction is made to the w.t. vector in accordance to the following rule

$$\left\{ \begin{array}{l} \underline{w}^{(n+1)} = \underline{w}^{(n)} \\ \underline{w}^{(n+1)} = \underline{w}^{(n)} \end{array} \right. \quad \text{if} \quad \begin{array}{l} \underline{w}^{T(n)} \underline{x}^{(n)} > 0 \\ \text{and } \underline{x}^{(n)} \in \mathcal{L}_1 \\ \underline{w}^{T(n)} \underline{x}^{(n)} \leq 0 \\ \text{and } \underline{x}^{(n)} \in \mathcal{L}_2 \end{array}$$

ELSE

2)

Update the weight vector as

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \eta^{(n)} \underline{x}^{(n)}$$

$$\text{if } \begin{cases} \underline{w}^T(n) \underline{x}^{(n)} > 0 \\ \underline{x}^{(n)} \in \mathcal{L}_2 \end{cases}$$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \eta^{(n)} \underline{x}^{(n)}$$

$$\text{if } \begin{cases} \underline{w}^T(n) \underline{x}^{(n)} \leq 0 \\ \underline{x}^{(n)} \in \mathcal{L}_1 \end{cases}$$

Here $\eta^{(n)}$ is a learning parameter
Sometimes / Most times η is a fixed constant
typically small.

$\eta = 1$ also works

Perceptron Convergence Theorem

Let us start with the initial condition $\underline{w}(0) = \underline{0}$

Suppose $\underline{w}^T(n) \underline{x}(n) \leq 0$ for $n = 1, 2, \dots$
and $\underline{x}(n) \in \mathcal{L}_1$ i.e., it is incorrectly classified

With $\eta = 1$

$$\underline{w}(n+1) = \underline{w}(n) + \underline{x}(n)$$

Let us recursively unfold $\underline{w}(n+1)$

$$\underline{w}(n+1) = \underline{x}(n) + \underline{x}(n-1) + \dots + \underline{x}(1) + \underline{x}(0) \quad (1)$$

Since the classes are linearly separable, $\underline{w}_{opt}^T \underline{x}^{(n)} > 0$

\exists a solution \underline{w}_{opt}
for all inputs $\{\underline{x}\} \in \mathcal{L}_1$

For a fixed \underline{w}_0 , let us define

$$d = \min_{\underline{x}^{(n)} \in \mathcal{L}_1} \underline{w}_0^T \underline{x}^{(n)} \quad \text{--- (2)}$$

Multiply (1) b.s. by \underline{w}_0^T

$$\underline{w}_0^T \underline{w}^{(n+1)} = \underbrace{\underline{w}_0^T \underline{x}^{(0)} + \underline{w}_0^T \underline{x}^{(1)} + \dots + \underline{w}_0^T \underline{x}^{(n)}}_{\text{Totally } (n+1) \text{ terms}}$$

$$\underline{w}_0^T \underline{w}^{(n+1)} \geq (n+1)d \quad \text{--- (3)}$$

Let us apply Cauchy Schwartz inequality to the L.H.S. of (3)

$$\| \underline{w}_0 \|^2 \| \underline{w}^{(n+1)} \|^2 \geq \left(\underline{w}_0^T \underline{w}^{(n+1)} \right)^2 \quad (4)$$

From (4) and (3)

$$\| \underline{w}_0 \|^2 \| \underline{w}^{(n+1)} \|^2 \geq (n+1)^2 \alpha^2$$
$$\| \underline{w}^{(n+1)} \|^2 \geq \frac{(n+1)^2 \alpha^2}{\| \underline{w}_0 \|^2} \quad (5)$$

Let us deviate slightly,
 $\underline{w}(k+1) = \underline{w}(k) + \underline{x}(k)$; $k = 1, \dots, n$
 $\underline{x}(k) \in \mathcal{L}_1$

Taking the Euclidean norm on both sides of the above eqn

$$\|\underline{w}(k+1)\|^2 = \|\underline{w}(k)\|^2 + \|\underline{x}(k)\|^2 + 2 \underline{w}^T(k) \underline{x}(k)$$

But $\underline{w}^T(k) \underline{x}(k) \leq 0$ by assumption

$$\therefore \|\underline{w}(k+1)\|^2 \leq \|\underline{w}(k)\|^2 + \|\underline{x}(k)\|^2$$

$$\|\underline{w}(k+1)\|^2 - \|\underline{w}(k)\|^2 \leq \|\underline{x}(k)\|^2$$

$k = 1, \dots, n$
6

$$\| \underline{w}^{(n+1)} \|^2 \leq \sum_{k=0}^n \| \underline{x}^{(k)} \|^2 \quad \text{_____} \quad (7)$$

$$\text{Let } \beta = \max_{\underline{x}^{(k)} \in \mathcal{L}_1} \| \underline{x}^{(k)} \|^2$$

$$\| \underline{w}^{(n+1)} \|^2 \leq (n+1) \beta \quad \text{_____} \quad (8)$$

From eqns. (8) and (5), there must \exists a n_{\max} / it satisfies (8) and (5) with equality

$$\frac{(1 + n_{\max})^2 \alpha^2}{\|\underline{w}_0\|^2} = (1 + n_{\max})^\beta$$

$$n_{\max} = \left\lceil \frac{\beta}{\alpha^2} \|\underline{w}_0\|^2 \right\rceil - 1$$

\Rightarrow A solution \underline{w}_0 exists and the perceptron algo. converges in n_{\max} iterations.



Notes

- 1) One can start with any $\underline{w}(0)$ instead of $\underline{w}(0) = \underline{0}$
- 2) One can choose η to be a fixed small +ve quantity

Aside (Implementation)

You can also normalize the updates ξ the data
So that the update rule is bounded.

Batch Perceptron Algorithm

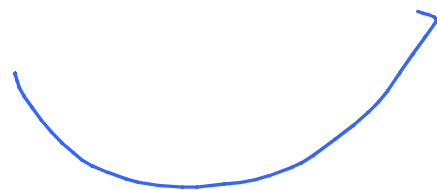
$$J(\underline{w}) = \sum_{\underline{x} \in \mathcal{H}} (-\underline{w}^T \underline{x}) d_{\underline{x}}$$

\mathcal{H} is the set of misclassified points
If all the points were correctly classified

$$\mathcal{H} = \emptyset; J(\underline{w}) = 0$$

$$\nabla_{\underline{w}} J = \sum_{\underline{x} \in \mathcal{H}} -\underline{x} d_{\underline{x}}$$

$$\begin{aligned} \underline{w}(n+1) &= \underline{w}(n) - \eta(n) \nabla_{\underline{w}} J(\underline{w}) \\ &= \underline{w}(n) + \eta(n) \sum_{\underline{x} \in \mathcal{H}} \underline{x} d_{\underline{x}} \end{aligned}$$



Proof of the batch perceptron algo

We assume that η remains the same for all 'n'.
Let $\eta = 1$. Let the initial weight vector be $\underline{w}(0) = \underline{0}$

Consider $\underline{w}^T \underline{x} \leq 0$ and $\underline{x} \in \mathcal{X}$

$$\underline{w}(n+1) = \underline{w}(n) + \sum_{\underline{x} \in \mathcal{X}_n} \underline{x}$$

$$= \sum_{\underline{x} \in \mathcal{X}_0} \underline{x} + \dots + \sum_{\underline{x} \in \mathcal{X}_m} \underline{x}$$

————— (A)

Let us consider $\underline{w}_0 / \underline{w}_0^T \underline{x} > 0 \quad \forall \underline{x} \in \mathcal{L}_1$

Let us premultiply (A) by \underline{w}_0^T

$$\underline{w}_0^T \underline{w}^{(n+1)} = \sum_{\underline{x} \in \mathcal{H}_0} \underline{w}_0^T \underline{x} + \dots + \underbrace{\sum_{\underline{x} \in \mathcal{H}_n} \underline{w}_0^T \underline{x}}_{(B)}$$

$$\text{Let } \alpha = \min_i \sum_{\underline{x} \in \mathcal{H}_i} \underline{w}_0^T \underline{x}$$

Now by Cauchy Schwarz inequality applied to LHS of (B)

$$\|\underline{w}_0\|^2 \|\underline{w}^{(n+1)}\|^2 \geq \left(\sum_{\underline{x} \in \mathcal{H}_n} \underline{w}_0^T \underline{x} \right)^2 \geq \frac{\alpha^2 \|\underline{w}_0\|^2 \|\underline{w}^{(n+1)}\|^2}{\|\underline{w}_0\|^2} \quad (C)$$

$$\underline{w}(n+1) = \underline{w}(n) + \sum_{\underline{x} \in \mathcal{X}_n} \underline{x}$$

$$\begin{aligned} \|\underline{w}(n+1)\|^2 &= \|\underline{w}(n)\|^2 + \left\| \sum_{\underline{x} \in \mathcal{X}_n} \underline{x} \right\|^2 \\ &\quad + 2 \underbrace{\underline{w}^T(n) \sum_{\underline{x} \in \mathcal{X}_n} \underline{x}}_{\leq 0} \end{aligned}$$

$$\|\underline{w}(n+1)\|^2 \leq \|\underline{w}(n)\|^2 + \left\| \sum_{\underline{x} \in \mathcal{X}_n} \underline{x} \right\|^2$$

Let us choose

$$\beta = \max_i \left\| \sum_{\underline{x} \in \mathcal{X}_i} \underline{x} \right\|^2$$

$$\|\underline{w}(n+1)\|^2 \leq \sum_{i=0}^n \left\| \sum_{\underline{x} \in \mathcal{X}_i} \underline{x} \right\|^2$$

$$\| \underline{w}^{(n+1)} \|^2 \leq (n+1)\beta \quad \text{--- (D)}$$

From (C) and (D) $\exists n_{\max}$ / the
 inequalities satisfy the equality condition

$$n_{\max} = \left\lceil \frac{\beta}{\alpha^2} \| \underline{w}_0 \|^2 \right\rceil - 1$$

Proves that the batch perceptron algorithm
 Converges after n_{\max} iterations

\Rightarrow

\square

Notes

In the batch perceptron algorithm 'n' is on an 'epoch' basis i.e., you group all the misclassified points ξ then do an update based on a gradient of the cost.

Whereas in the perceptron algo, 'n' is the online time step where data is presented to the N. Netw.

Relationship between perceptron and Bayes classifier

Consider a 2 class problem

Let p_i , $i = 1, 2$ be the a priori probability

C_{ij} be the cost of deciding l_i when l_j is true
 $i, j \in 1, 2$

$p(\underline{x} | l_i)$ be the conditional prob. density of a
data point \underline{x} in l_i

The risk

$$R = c_{11} p_1 \int_{\mathcal{H}_1 \text{ (Correct)}} p(\underline{x} | e_1) d\underline{x} + c_{21} p_1 \int_{\mathcal{H}_2 \text{ (incorrect)}} p(\underline{x} | e_1) d\underline{x} \\ + c_{12} p_2 \int_{\mathcal{H}_1 \text{ (incorrect)}} p(\underline{x} | e_2) d\underline{x} + c_{22} p_2 \int_{\mathcal{H}_2 \text{ (correct)}} p(\underline{x} | e_2) d\underline{x}$$

Any data point \in either \mathcal{H}_1 or \mathcal{H}_2

The following quantities are obvious:

$$\int_{\mathcal{X}} p(x|e_1) dx = \int_{\mathcal{X}} p(x|e_2) dx = 1$$

$$\left. \begin{array}{l} c_{11} < c_{21} \\ c_{22} < c_{12} \end{array} \right\}$$

$$R = c_{11} p_1 \int_{\mathcal{X}_1} p(\underline{x} | \ell_1) d\underline{x} + c_{22} p_2 \int_{\mathcal{X} - \mathcal{X}_1} p(\underline{x} | \ell_2) d\underline{x}$$

\mathcal{X}_1 ; corresponds to ℓ_1 $\mathcal{X} - \mathcal{X}_1$: corresponds to ℓ_2

$$+ c_{21} p_1 \int_{\mathcal{X} - \mathcal{X}_1} p(\underline{x} | \ell_1) d\underline{x} + c_{12} p_2 \int_{\mathcal{X}_1} p(\underline{x} | \ell_2) d\underline{x}$$

Idea: Keep $c_{21} p_1 + c_{22} p_2$ fixed

$$\begin{aligned}
 R = & c_{21} p_1 + c_{22} p_2 - \int_{\mathcal{X}_1} p_1 c_{21} P(x|e_1) dx \\
 & - \int_{\mathcal{X}_1} p_2 c_{22} P(x|e_2) dx \\
 & + \int_{\mathcal{X}_1} p_2 c_{12} P(x|e_2) dx \\
 & + \int_{\mathcal{X}_1} c_{11} p_1 P(x|e_1) dx
 \end{aligned}$$

Suppose $c_{21} p_1 + c_{22} p_2$ is a fixed cost.

$$R = c_{21} p_1 + c_{22} p_2 + \int_{\mathcal{L}_2} p_2 (c_{12} - c_{22}) \mathcal{P}(\underline{x} | \ell_2) d\underline{x} - \int_{\mathcal{L}_1} p_1 (c_{21} - c_{11}) \mathcal{P}(\underline{x} | \ell_1) d\underline{x}$$

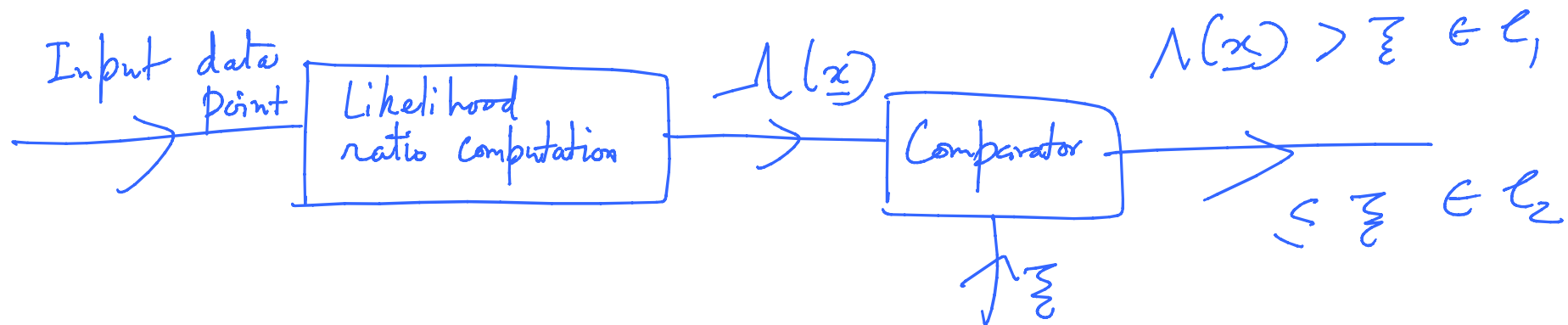
We want to minimize R

The integral must vanish for a threshold

$$P_1 (c_{21} - c_{11}) P(\underline{x} | \ell_1) \stackrel{\underline{x} \in \ell_1}{>} P_2 (c_{12} - c_{22}) P(\underline{x} | \ell_2)$$

Let $\lambda(\underline{x}) \triangleq \frac{P(\underline{x} | \ell_1)}{P(\underline{x} | \ell_2)}$ and

$$\underbrace{\lambda}_{\text{(threshold)}} = \frac{P_2 (c_{12} - c_{22})}{P_1 (c_{21} - c_{11})}$$



Example : Gaussian Distributions over 2 classes

Let us consider a multivariate Gaussian distribution

$$\text{Class } \ell_1 : \begin{aligned} E(\underline{x}) &= \underline{\mu}_1 \\ E\left((\underline{x} - \underline{\mu}_1)(\underline{x} - \underline{\mu}_1)^T\right) &= C \end{aligned}$$

$$\text{Class } \ell_2 : \begin{aligned} E(\underline{x}) &= \underline{\mu}_2 \\ E\left((\underline{x} - \underline{\mu}_2)(\underline{x} - \underline{\mu}_2)^T\right) &= C \end{aligned}$$

C is non-diagonal & we assume it is non-singular

$$P(\underline{x} | c_i) = \frac{1}{(2\pi)^{m/2} (\det(C))^{1/2}} e^{-\frac{1}{2} (\underline{x} - \underline{\mu}_i)^T C^{-1} (\underline{x} - \underline{\mu}_i)}$$

$i = 1, 2$

Let us suppose

(a) $P_1 = P_2$

(b) Misclassifications have the same cost

$C_{21} = C_{12}$

$\xi \quad C_{11} = C_{22}$

$$\begin{aligned}
 \log \Lambda(\underline{x}) &= \log \left[\frac{p(\underline{x} | \mu_1)}{p(\underline{x} | \mu_2)} \right] && \underbrace{(C^{-1})^T = C^{-1}} \\
 &= -\frac{1}{2} (\underline{x} - \underline{\mu}_1)^T C^{-1} (\underline{x} - \underline{\mu}_1) \\
 &\quad + \frac{1}{2} (\underline{x} - \underline{\mu}_2)^T C^{-1} (\underline{x} - \underline{\mu}_2) \\
 &= \underbrace{(\underline{\mu}_1 - \underline{\mu}_2)^T C^{-1}}_{\underline{w}^T} \underline{x} + \frac{1}{2} \underbrace{\left(\underline{\mu}_2^T C^{-1} \underline{\mu}_2 - \underline{\mu}_1^T C^{-1} \underline{\mu}_1 \right)}_{b}
 \end{aligned}$$

Let $\underline{w} = C^{-1} (\underline{\mu}_1 - \underline{\mu}_2)$

We have an equation of the form

$$y = \underline{w}^T \underline{x} + b$$

This is a linear classifier

Bayes classifier for

Gaussian distributions

≡
≡
≡
(Similar)

Perceptron

Subtle Differences b/w Perceptron & Bayes' Classifier

- 1) Perceptron assumes inherently that the patterns/data points are linearly separable.
This is not the case in Bayes' classifier.
- 2) Bayes' classifier minimizes the prob. of classification error
- 3) Perceptron is non-parametric \Rightarrow No assumptions on underlying distribution
- 4) Bayes' classifier is parametric
Perceptron update & convergence is simple & adaptive; Bayes' is not