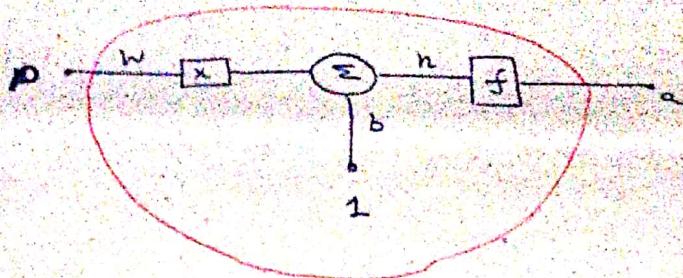


Neural Networks

p.w

Single input neuron

$$x = p \cdot w$$

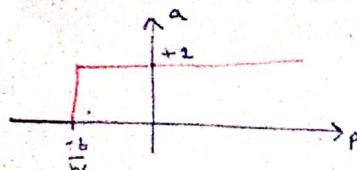
$$n = x + b \cdot 1 = p \cdot w + b$$

$$a = f(n) = f(p \cdot w + b)$$

Transfer / Activation functions

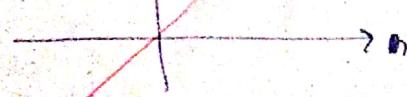
* Hardlimit = step function

$$* a = \text{hardlim}(w \cdot p + b) \rightarrow p \geq -\frac{b}{w} \rightarrow +1, \quad p < -\frac{b}{w} \rightarrow 0$$



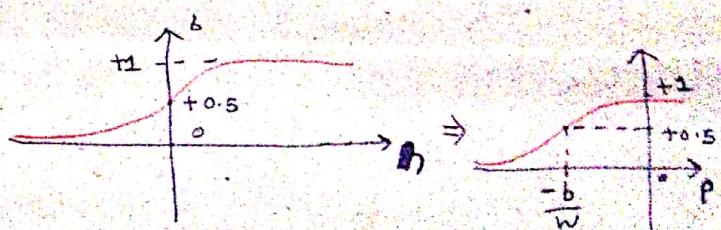
* Linear = purelin

$$* a = w \cdot p + b$$



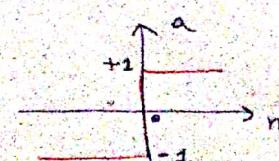
* Log-sigmoid

$$* a = \frac{1}{1+e^{-n}}; n = w \cdot p + b$$



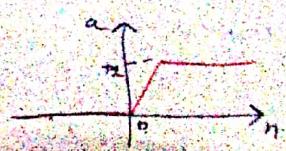
* Symmetric-Hard limit

$$* a = \begin{cases} +1, & n \geq 0 \\ -1, & n < 0 \end{cases}$$



* Saturating linear

$$* a = \begin{cases} 0; & n < 0 \\ n; & 0 \leq n \leq 1 \\ 1; & n > 1 \end{cases}$$



* Sigmoidic (Inverting) function

$$a = \begin{cases} 1 & n = 0 \\ \frac{1}{1 + e^{-n}} & n \neq 0 \\ 0 & n = 1 \end{cases}$$

* Hyperbolic tangent sigmoid

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

* Positive linear

$$a = \begin{cases} n & n > 0 \\ 0 & n = 0 \\ 1 & n < 0 \end{cases}$$

* Competitive

$$a = \begin{cases} 1 & i \text{ is neuron with maximum off value } n \\ 0 & \text{all other neurons} \end{cases}$$

$$\Theta \rightarrow n = w \cdot p + b$$

$$w = [w_{11} \ w_{12} \ w_{13} \dots \ w_{1r}]^T \in \mathbb{R}^{r \times 1}$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_r \end{bmatrix} \in \mathbb{R}^{r \times 1} \Rightarrow a = f(w \cdot p + b)$$

1) Graphical method

Class - 4

$$w^T p + b = 0$$

Boundary line

→ boundary line $\Rightarrow w^T p = \text{some } (-b)$ in boundary line

→ if boundary line passes through origin then $b = 0$

p_1

p_2

p_3

p_4

p_5

p_6

p_7

p_8

p_9

p_{10}

p_{11}

p_{12}

p_{13}

p_{14}

p_{15}

p_{16}

p_{17}

p_{18}

p_{19}

p_{20}

p_{21}

p_{22}

p_{23}

p_{24}

p_{25}

p_{26}

p_{27}

p_{28}

p_{29}

p_{30}

p_{31}

p_{32}

p_{33}

p_{34}

p_{35}

p_{36}

p_{37}

p_{38}

p_{39}

p_{40}

p_{41}

p_{42}

p_{43}

p_{44}

p_{45}

p_{46}

p_{47}

p_{48}

p_{49}

p_{50}

p_{51}

p_{52}

p_{53}

p_{54}

p_{55}

p_{56}

p_{57}

p_{58}

p_{59}

p_{60}

p_{61}

p_{62}

p_{63}

p_{64}

p_{65}

p_{66}

p_{67}

p_{68}

p_{69}

p_{70}

p_{71}

p_{72}

p_{73}

p_{74}

p_{75}

p_{76}

p_{77}

p_{78}

p_{79}

p_{80}

p_{81}

p_{82}

p_{83}

p_{84}

p_{85}

p_{86}

p_{87}

p_{88}

p_{89}

p_{90}

p_{91}

p_{92}

p_{93}

p_{94}

p_{95}

p_{96}

p_{97}

p_{98}

p_{99}

p_{100}

p_{101}

p_{102}

p_{103}

p_{104}

p_{105}

p_{106}

p_{107}

p_{108}

p_{109}

p_{110}

p_{111}

p_{112}

p_{113}

p_{114}

p_{115}

p_{116}

p_{117}

p_{118}

p_{119}

p_{120}

p_{121}

p_{122}

p_{123}

p_{124}

p_{125}

p_{126}

p_{127}

p_{128}

p_{129}

p_{130}

p_{131}

p_{132}

p_{133}

p_{134}

p_{135}

p_{136}

p_{137}

p_{138}

p_{139}

p_{140}

p_{141}

p_{142}

p_{143}

p_{144}

p_{145}

p_{146}

p_{147}

p_{148}

p_{149}

p_{150}

p_{151}

p_{152}

p_{153}

p_{154}

p_{155}

p_{156}

p_{157}

p_{158}

p_{159}

p_{160}

p_{161}

p_{162}

p_{163}

p_{164}

p_{165}

p_{166}

p_{167}

p_{168}

p_{169}

p_{170}

p_{171}

p_{172}

p_{173}

p_{174}

p_{175}

p_{176}

p_{177}

p_{178}

p_{179}

p_{180}

p_{181}

p_{182}

p_{183}

p_{184}

p_{185}

p_{186}

p_{187}

p_{188}

p_{189}

p_{190}

p_{191}

p_{192}

p_{193}

p_{194}

p_{195}

p_{196}

p_{197}

p_{198}

p_{199}

p_{200}

p_{201}

p_{202}

p_{203}

p_{204}

p_{205}

p_{206}

p_{207}

p_{208}

p_{209}

p_{210}

p_{211}

p_{212}

p_{213}

p_{214}

p_{215}

p_{216}

p_{217}

p_{218}

p_{219}

p_{220}

p_{221}

p_{222}

p_{223}

p_{224}

p_{225}

p_{226}

p_{227}

p_{228}

p_{229}

p_{230}

p_{231}

p_{232}

p_{233}

p_{234}

p_{235}

p_{236}

p_{237}

p_{238}

p_{239}

p_{240}

p_{241}

Hardlim/

2) General Method \rightarrow Perceptron Rule

$$* w_{\text{new}} = w_{\text{old}} + e \cdot p$$

$$* b_{\text{new}} = b_{\text{old}} + e$$

purelin/.

3) Hebbian Learning Rule (Hebb Rule)

* Training set = $\{p_1, t_1\}, \{p_2, t_2\}, \{p_3, t_3\}, \dots, \{p_n, t_n\}$

$$w_{\text{new}} = w_{\text{old}} + t_e \cdot p_2$$

$$b_{\text{new}} = b_{\text{old}} + t_2$$

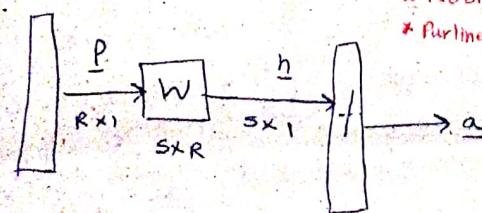
* Initial weight matrix, bias = 0 :-

$$b = \sum_{i=1}^n t_i$$

$$T = [t_1 \ t_2 \ \dots \ t_n],$$

$$P = [p_1 \ p_2 \ \dots \ p_n].$$

Linear Associator



$$a = \text{purelin}(s_x_1) = h = W \cdot p + b$$

$$a = W \cdot p - ①$$

$$a_i = \sum_j W_{ij} \cdot p_j$$

$$\text{But, } W = t_1 \cdot p_1^T + t_2 \cdot p_2^T + \dots + t_n \cdot p_n^T$$

$$W = \sum_{i=1}^n t_i \cdot p_i^T - ②$$

For an input p_k ;

$$a_k = \left(\sum_{i=1}^n t_i \cdot p_i^T \right) \cdot p_k$$

$$a_k = \sum_{i=1}^n t_i \cdot (p_i^T \cdot p_k)$$

If p_i 's are orthonormal Then, $p_i^T \cdot p_k = \begin{cases} 1 & i=k \\ 0 & i \neq k \end{cases}$

$$a_k = t_k$$

∴ Inputs vectors are orthonormal output. then Hebb rule produces correct

Page 20 in Hasintha's notes/.

Purelin(1), Pseudo inverse Rule

- * If the input vectors are not orthonormal then hebb rule gives an error.

$$* \underline{W} \cdot \underline{P}_Q = \underline{t}_Q \quad \forall Q = 1, 2, 3, \dots, Q$$

\hookrightarrow performance index / error; $F(W) = \sum_{Q=1}^Q \| \underline{t}_Q - \underline{W} \cdot \underline{P}_Q \|^2$

$$F(W) = \| E \|^2 \Rightarrow E = T - W \cdot P$$

* $F(W)$ can be zero if, $\boxed{W = T P^{-1}}$ P is not always a

$$W = T P^+$$

\hookrightarrow Pseudoinverse matrix

Not always possible to take
 P^{-1}

$$P \cdot P^+ \cdot P = P$$

$$P^+ \cdot P \cdot P^+ = P^+$$

$$P^+ P = (P^+ P)^T$$

$$P P^+ = (P P^+)^T$$

Note :- Number of rows in $P >$

Number of columns in P ,

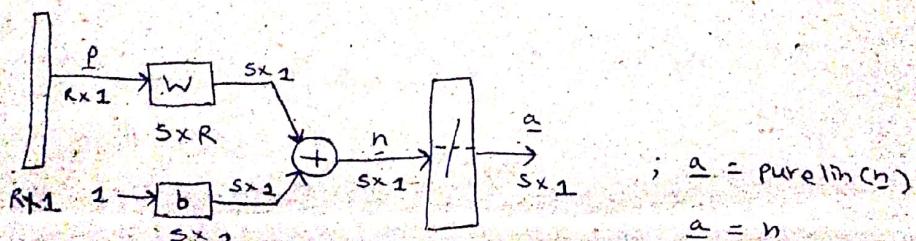
Columns of P are independent

$$\rightarrow P^+ = (P^T P)^{-1} \cdot P^T$$

Purelin(1), Widrow - Hoff Learning Rule

ADALINE Network

\uparrow
ADaptive LInear NEuron \rightarrow A perceptron network with a linear transfer function



LMS [Least Mean Squared] Algorithm

$$* \underline{x} = \begin{bmatrix} \underline{w} \\ b \end{bmatrix}, \underline{z} = \begin{bmatrix} \underline{p} \\ \underline{t} \end{bmatrix} \quad ; \quad a = \text{purelin}(n) = n$$

But, MSE = $F(\underline{x}) \Rightarrow$

$$F(\underline{x}) = E(e^a) = E[(t-a)^2] = E[(t - \underline{x}^T \underline{z})^2]$$

$$\therefore F(\underline{x}) = E[t^2 - 2t \cdot \underline{x}^T \underline{z} + (\underline{x}^T \underline{z})^2]$$

$$= E[t^2] - 2 \underline{x}^T E[t \cdot \underline{z}] + \underline{x}^T E[\underline{z} \cdot \underline{z}^T] \cdot \underline{x}$$

$$= (1 - \min(0, t))^2 + \min(1, t) \cdot \min(0, \underline{x}^T \underline{z}) = \sigma$$

$$F(\underline{x}) = C - \frac{1}{2} \underline{x}^T \underline{h} + \frac{1}{2} \underline{x}^T R \underline{x}$$

(5)

General form of quadratic function:-

$$F(\underline{x}) = C + \underline{d}^T \underline{x} + \frac{1}{2} \underline{x}^T A \underline{x}$$

$$\therefore \underline{d} = -2\underline{h}, \quad A = 2R$$

Locating the minimum \Rightarrow

$$\begin{aligned}\nabla F(\underline{x}) &= \nabla(C + \underline{d}^T \underline{x} + \frac{1}{2} \underline{x}^T A \underline{x}) \\ &= \underline{d}^T \cdot \frac{1}{2} \cdot 2A \cdot \underline{x}\end{aligned}$$

$$\therefore \nabla F(\underline{x}) = -2\underline{h} + 2R \cdot \underline{x}$$

* @ stationary points,

$$\nabla F(\underline{x}) = 0$$

$$R \cdot \underline{x} = \underline{h}$$

$$\underline{x}^* = R^{-1} \cdot \underline{h}$$

Positive definite:
* square matrix = A;
 $\forall \underline{x} \neq 0 \Rightarrow \underline{x}^T A \underline{x} > 0$

Semi-positive definite:
* square matrix = A;
 $\forall \underline{x} \neq 0 \Rightarrow \underline{x}^T A \underline{x} \geq 0$

$$\begin{aligned}w_{k+1} &= w_k + 2\alpha \cdot e_k \cdot p_k^T \\ b_{k+1} &= b_k + 2\alpha \cdot e_k\end{aligned}$$

where,

$$0 < \alpha < \frac{1}{\lambda_{\max}}$$

λ_{\max} = Maximum eigen value of R;

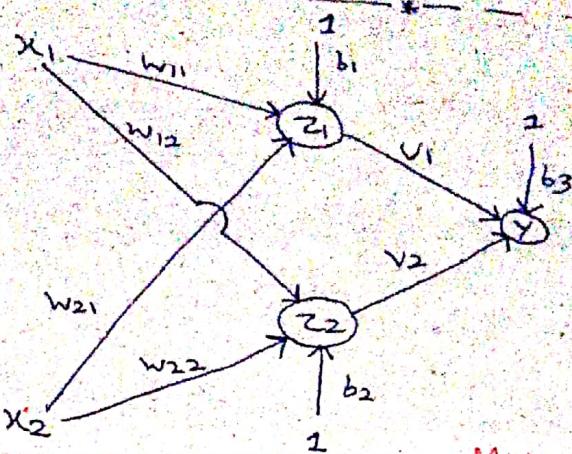
$$R = E[\underline{\zeta} \cdot \underline{\zeta}^T]$$

$$\underline{\zeta} = \left[\begin{array}{c} p \\ \vdots \\ p \end{array} \right]$$

$$R = \frac{1}{n} [P_1 P_1^T + P_2 P_2^T]$$

hardlim/1.

MADALINE [Many ADALINE]



Training

→ MR1:

Changes $w_{11}, w_{12}, w_{21}, w_{22}$,
 b_1, b_2

→ MR2:

Changes all parameters

Motivation for updates:

1) If we perform the updates only if there is an error

2) If $t = 1 \neq y = -1$

* $z_1 = z_2 = -1$

* Need to make atleast one of them to 1

3) If $t = -1 \neq y = 1$

* Atleast one z has 1

* Need to make all to -1

$\rightarrow t = 1, y = -1 \rightarrow$

$$\omega_{11}(\text{new}) = \omega_{11}(\text{old}) + \alpha \cdot (1 + z_{in1}),$$

$$\omega_{21}(\text{new}) = \omega_{21}(\text{old}) + \alpha \cdot (1 + z_{in1}),$$

$$\omega_{12}(\text{new}) = \omega_{12}(\text{old}) + \alpha \cdot (1 + z_{in2}),$$

$$\omega_{22}(\text{new}) = \omega_{22}(\text{old}) + \alpha \cdot (1 + z_{in2}).$$

$\rightarrow t = -1, y = 1 \rightarrow$

$$\omega_{11}(\text{new}) = \omega_{11}(\text{old}) + \alpha \cdot (-1 - z_{in1}),$$

$$\omega_{21}(\text{new}) = \omega_{21}(\text{old}) + \alpha \cdot (-1 - z_{in1}),$$

$$\omega_{12}(\text{new}) = \omega_{12}(\text{old}) + \alpha \cdot (1 + z_{in2}),$$

$$\omega_{22}(\text{new}) = \omega_{22}(\text{old}) + \alpha \cdot (1 + z_{in2}).$$