Eigenvector (A) $\rightarrow [\vec{V_1} : \vec{V_2}]$

$\rightarrow \{\lambda_1, \lambda_2\}$

$$A\vec{V_1} = \lambda_1 \vec{V_1}$$

$$A\vec{V_2} = \lambda_2 \vec{V_2}$$

$$A \overbrace{[V_1, \vec{V_2}]}^{V} = \overbrace{[\vec{V_1} : \vec{V_2}]}^{V} \overbrace{\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}}^{D}$$

$$AV = VD \Rightarrow A = \cancel{V} D V^{-1}$$

Lecture cont~~i~~ (Recall)

$$y = \{0, 1\}$$

$x_1, \dots, x_p$ feature

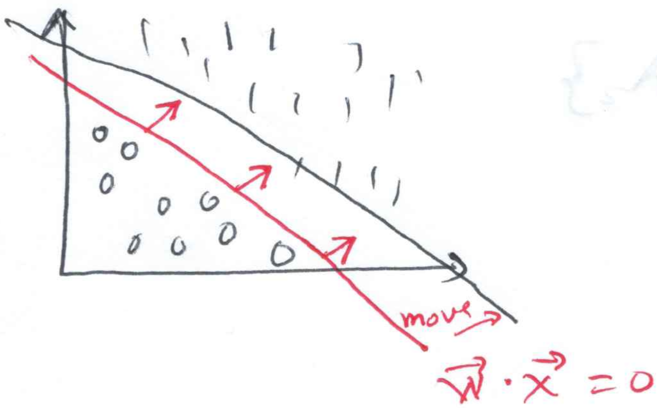
need $\hat{y} = g(\vec{x})$

$$\mathbb{D} = \left\{ \begin{bmatrix} \leftarrow \vec{x}_1 \rightarrow \\ \leftarrow \vec{x}_2 \rightarrow \\ \vdots \\ \leftarrow X_n \rightarrow \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\}$$

$$\mathcal{A}(H, \mathbb{D})$$

$$\mathcal{H} = \left\{ \mathbb{1}_{\vec{w} \cdot \vec{x} > 0}, \vec{w} \in \mathbb{R}^{p+1} \right\}$$

➢ Perceptron Learning Algorithm



① $\vec{w} = \vec{0} \longrightarrow \vec{w}^{t=0}$ or random.

② Calculate $\hat{y}_i = \mathbb{1}$

③ update all $j = 1 \ldots p+1$

$$W_1^{t=1} = W_1^{t=0} + (y_i - \hat{y}_i)$$

$$W_2^{t=1} = W_2^{t=1} + (y - \hat{y}_2)\, x_{i1}$$

$$\vdots$$

$$W_{p+1}^{t=1} = W_{p+1}^{t=0} + (y_i - \hat{y}_i)\, x_p$$

④ Repeat for $i = 1 \ldots$

⑤ Repeat steps $2 \to 4$ until error is reached.

or a max # of iterat°s.

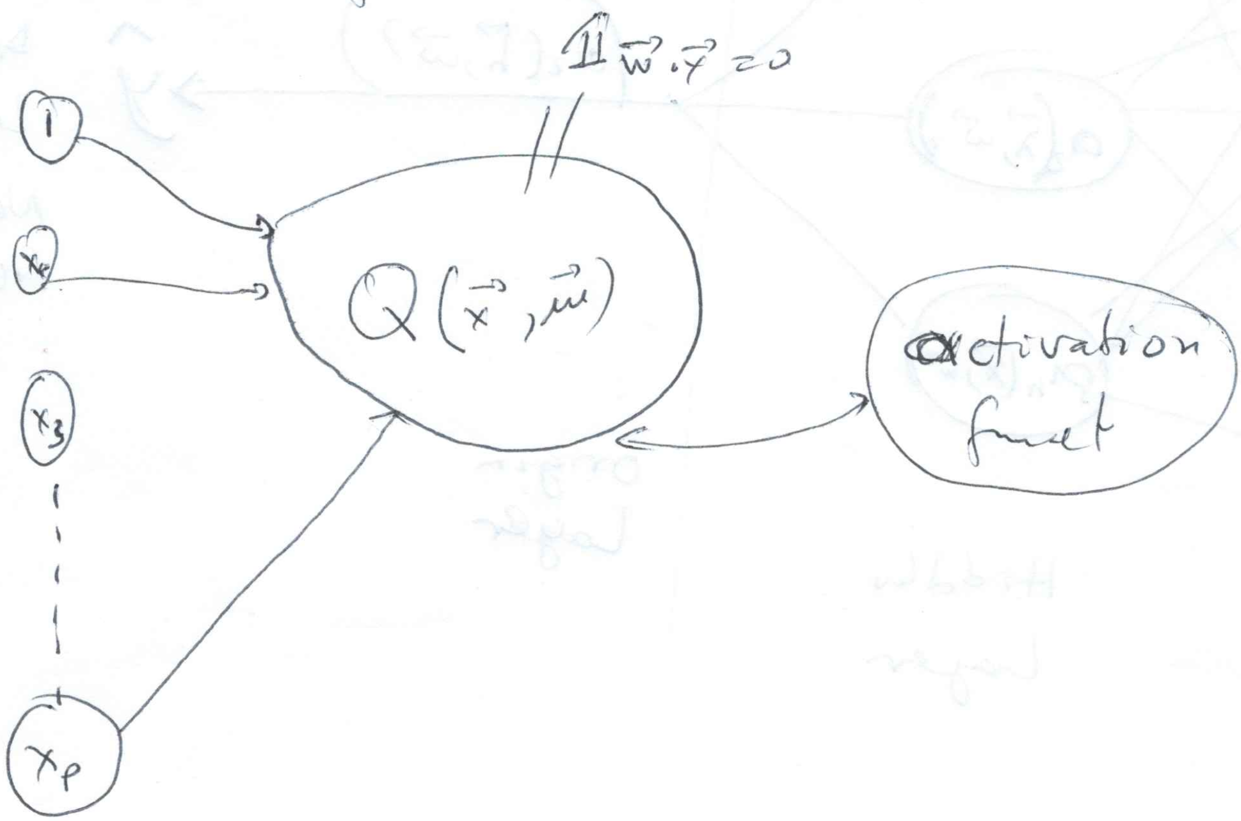
If $\mathbb{D}$ is "linearly separable" i.e $\exists \vec{u}$

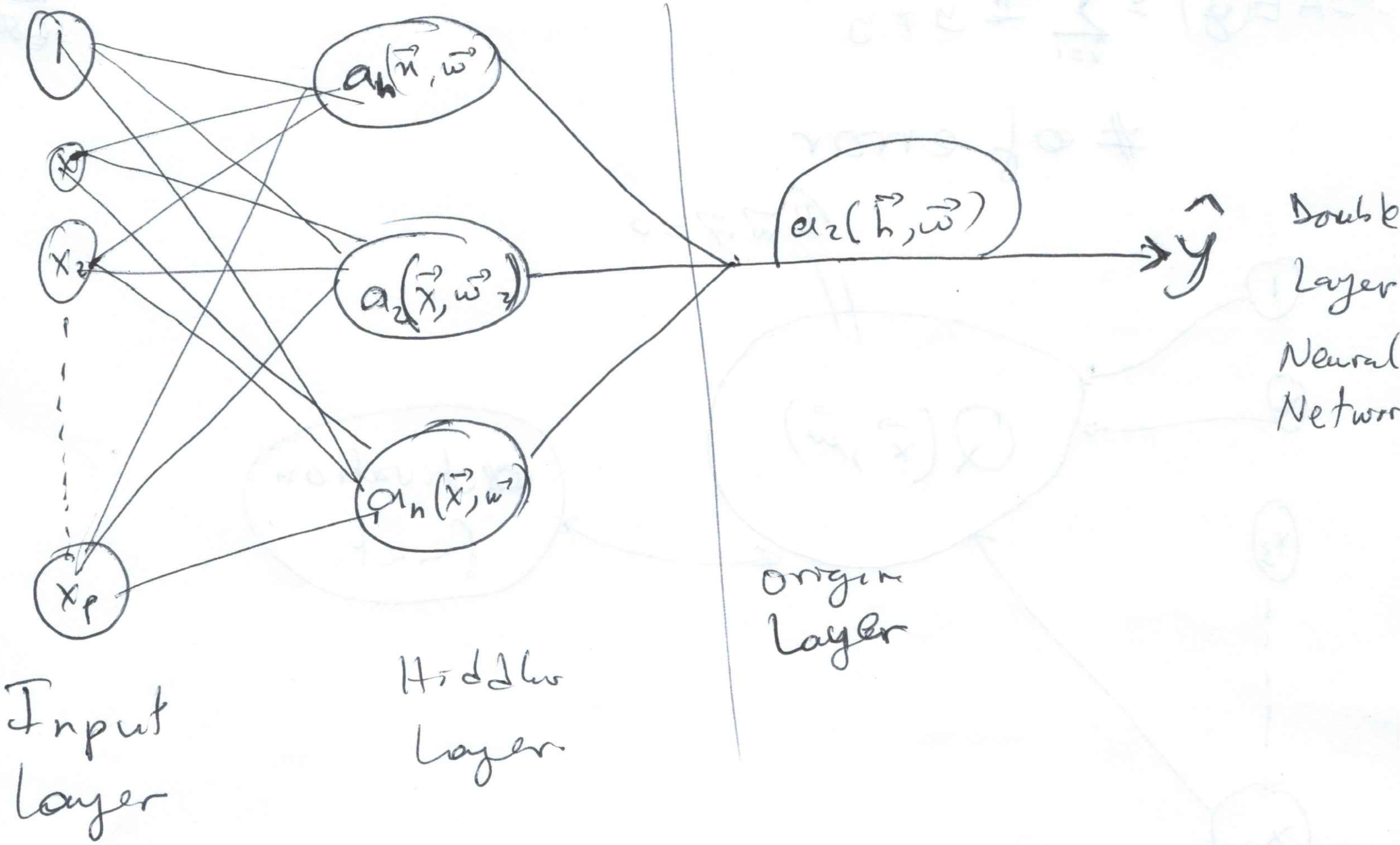
s.t $\mathbb{1}_{\vec{w}\cdot\vec{x} > 0}$ yelled an error in $\mathbb{D}$, then

th. alg will find it.

$$SAE(g) = \sum_{i=1}^{n} \mathbb{1}_{\hat{y} \neq y}$$

# of error




1
x_1
x_2
x_p
$a_1(\vec{x},\vec{w})$
$a_2(\vec{x},\vec{w}_2)$
$a_n(\vec{x},\vec{w})$
$a_2(\vec{h},\vec{w})$
$\hat{y}$
Input Layer
Hidden Layer
Origin Layer
Double Layer Neural Networ