2/14/18

Lecture 5

$\text{eign}(A) \to [\vec{V}_1 \vdots \vec{V}_2]$, $\{\lambda_1, \lambda_2\}$ $\quad A\vec{v}_1 = \lambda_1 \vec{v}_1$, $A\vec{v}_2 = \lambda_2 \vec{v}_2$

$$A[\vec{V}_1 \vdots \vec{V}_2] = [\vec{V}_1 \vdots \vec{V}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad AV = VD \to A = VDV^{-1}$$

$\mathcal{Y} = \{0, 1\}$

$x_1, \dots, x_p$ features

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

$$\mathbb{D} = \left\{ \begin{bmatrix} \leftarrow x_1 \rightarrow \\ \leftarrow x_2 \rightarrow \\ \vdots \\ \leftarrow x_n \rightarrow \end{bmatrix}, \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\}$$

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

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

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

objective function, core function, fitness function
(# of errors in $\mathbb{D}$)

<u>Perceptron Learning Algorithm</u>

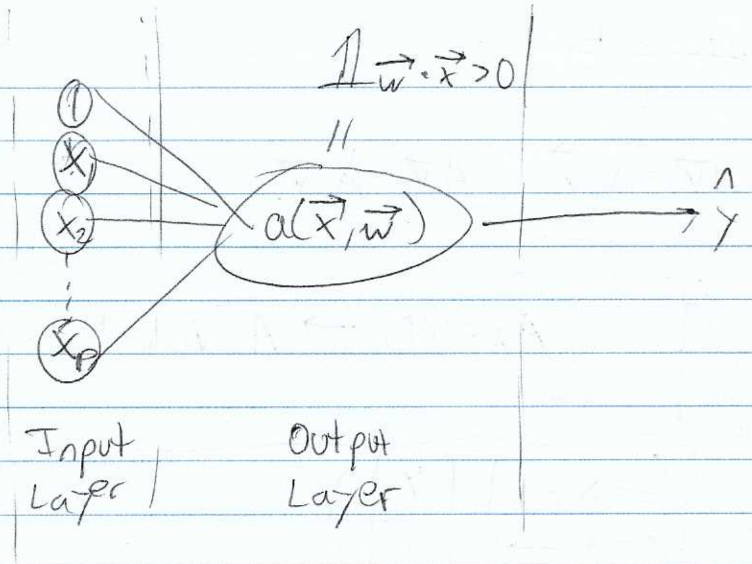
① Initialize $\vec{w} = \vec{0} = \vec{w}^{t=0}$ or random
② Calculate $\hat{y}_i = \mathbb{1}_{\vec{w}^{t=0} \cdot \vec{x}}$
③ Update all weights $j = 1 \dots p+1$

$$w_1^{t=1} = w_1^{t=0} + (y_i - \hat{y}_i)1$$
$$w_2^{t=1} = w_2^{t=0} + (y_i - \hat{y}_i)x_{i1}$$
$$\vdots$$
$$w_{p+1}^{t=1} = w_{p+1}^{t=0} + (y_i - \hat{y}_i)x_{p1}$$

④ Repeat for $i = 1 \dots n$
⑤ Repeat steps 2-4 until a threshold error is reached or max. # of iterations.

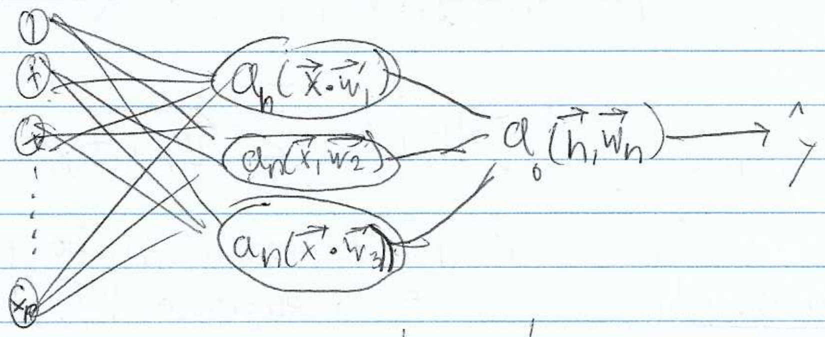
If $\mathbb{D}$ is "linearly seperable" i.e. $\exists \vec{w}$ s.t. $\mathbb{1}_{\vec{w} \cdot \vec{x} > 0}$ yields no errors in $\mathbb{D}$, then it has no errors

(But in practice its rare because few things are seperable)

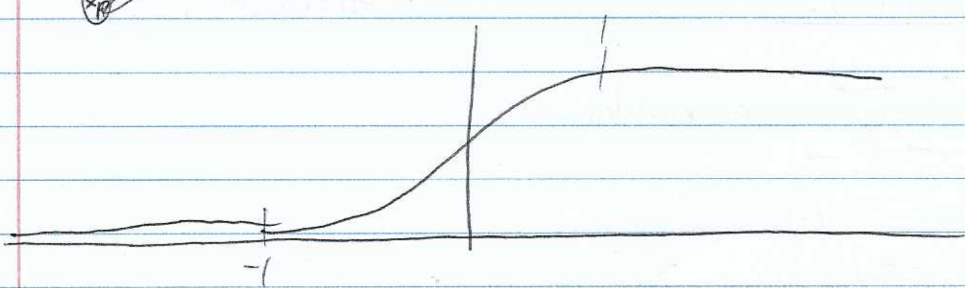


Single Layer Neural Network

Depth = 1



Depth = 2



Sigmoid, logistic