4/1/15

(FINISH) SVMs + NEURAL NETWORKS

① Multi-class SVM

→ So far: Binary SVM $y \in \{+1, -1\}$

→ For multiclass $y \in \{1, 2, \ldots K\}$; your options:

→ One-vs-Rest (All) K-binary SVMs

→ One-vs-One $\binom{K}{2}$-binary SVMs

→ [Gammer & Singer JMLR '01]

1-natively multi-class SVM

K weight vectors $\{\vec{w}^{(1)}, \ldots, \vec{w}^{(K)}\}$ $\vec{w}^{(i)} \in \mathbb{R}^d$

K "scores" for each input $\vec{x}$

$$S^{(1)}(\vec{x}) = \vec{w}^{(1)} \cdot \vec{x}$$

$$\vdots$$

$$S^{(K)}(\vec{x}) = \vec{w}^{(K)} \cdot \vec{x}$$

prediction $\Rightarrow$ $\hat{y} = \underset{c \in \{1, \ldots K\}}{\operatorname{argmax}} S^{(c)}(\vec{x})$

How to learn parameters?

[Similar structure as binary SVM OP]

$$\min \quad \text{Norm}^2(\vec{w}) + C \sum_i \text{slack}_i$$

$$\text{s.t} \quad \text{Correct classification} \geq \text{Margin} - \text{slack}_i$$

$$\Leftrightarrow \min_{\substack{\vec{w}^{(1)} \dots \vec{w}^{(K)} \\ b^{(1)} \dots b^{(K)}}} \frac{1}{2}\sum_{c=1}^{K} \|w^{(c)}\|^2 + C\sum_{i=1}^{N} \xi_i$$

s.t $\underbrace{\left[\vec{w}^{(y_i)} \cdot \vec{x}_i + b^{(y_i)}\right]}_{\text{Score of correct class } y_i} - \underbrace{\left[\vec{w}^{(y)} \cdot \vec{x}_i + b^{(y)}\right]}_{\text{Score of incorrect class } y} \geq 1 - \xi_i \quad \underbrace{\forall i, y \neq y_i}_{\text{for all incorrect classes}}$

$\xi_i \geq 0$

Still a QP! → $K(d+1)$ vars

→ $N \cdot (K-1) + N$ constraints

Exercise: For $K=2$, can we reduce this to a "regular" binary SVM?
[Yes, with an assumption]

$$\min_{\substack{\vec{w}^{(1)}\ \vec{w}^{(2)} \\ b^{(1)}\ b^{(2)}}} \frac{1}{2}\left[\vec{w}^{(1)} \cdot \vec{w}^{(1)} + \vec{w}^{(2)} \cdot \vec{w}^{(2)}\right] + C\sum_{i=1} \xi_i$$

s.t
$$\begin{cases} w^{(1)} \cdot \vec{x}_i + b^{(1)} - \vec{w}^{(2)} \cdot \vec{x}_i - b^{(2)} \geq 1 - \xi_i & \forall i\ y_i = 1 \\ \vec{w}^{(2)} \cdot \vec{x}_i + b^{(2)} - \vec{w}^{(1)} \cdot \vec{x}_i - b^{(1)} \geq 1 - \xi_i & \forall i\ y_i = 2 \end{cases}$$

→ assume/constrain $\vec{w}^{(1)} = -\vec{w}^{(2)}$, $b^{(1)} = -b^{(2)}$

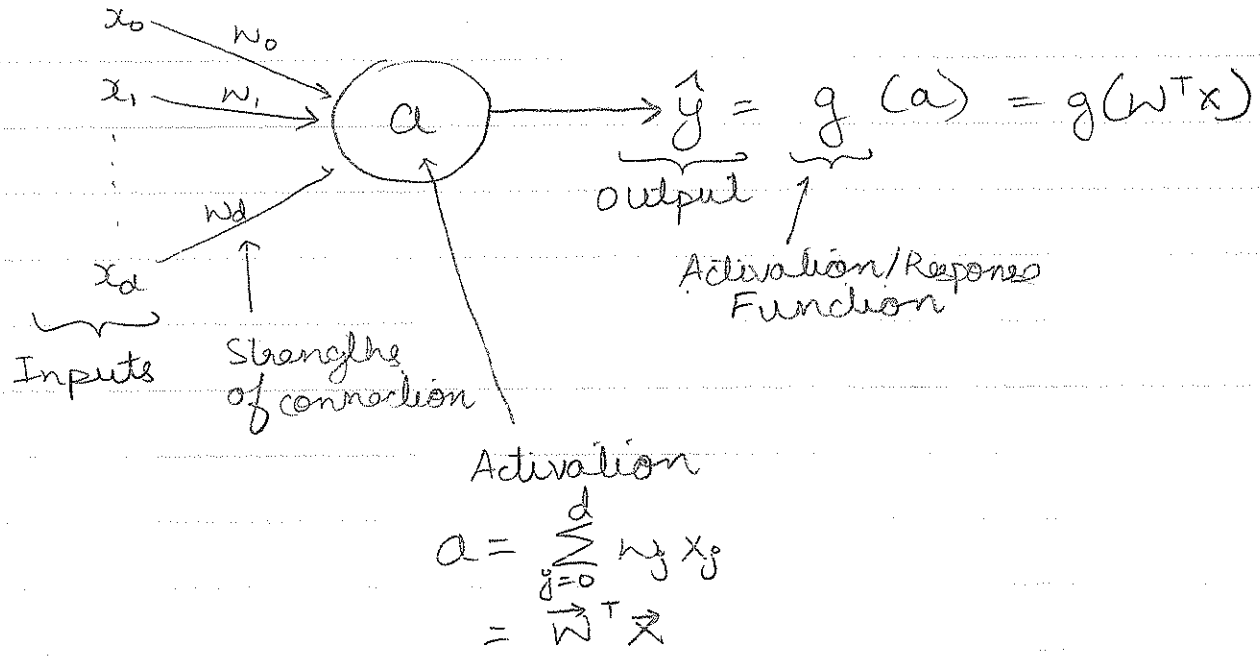
→ get back binary SVM QP (up to constants)

(2) Where we are going: with these classifiers:

Purely Linear Classifiers	Kernel Classifiers	"Naturely" Non-linear Classifiers
[Linear in $\vec{x}$ & $\vec{w}$]	[Linear in $\Phi(\vec{x})$ non-linear in $\vec{x}$, but linear in $\vec{w}$]	[Non-linear in $\vec{x}$ & $\vec{w}$]
GNB, LR (linear) SVM	Kernel SVMs	Neural Nets, Decision Trees

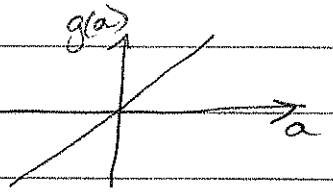
→ Flow of class

(3) Artificial "Neuron"



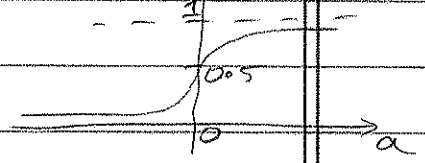
Many different activation functions possible

(a) Linear $g(a) = a$



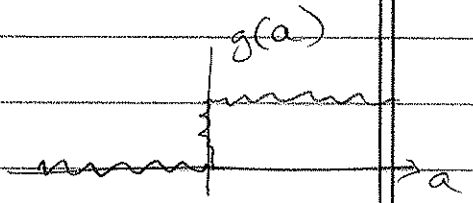
$\Rightarrow \hat{y} = g(\vec{w}^T \vec{x}) = \vec{w}^T \vec{x} \Leftrightarrow$ Linear Regression

(b) Sigmoid / Logistic $g(a) = \frac{1}{1+e^{-a}}$



$\Rightarrow \hat{y} = \frac{1}{1+e^{-w^T x}}$ | Logistic Regression $P(Y=1 \mid \vec{x}, \vec{w})$

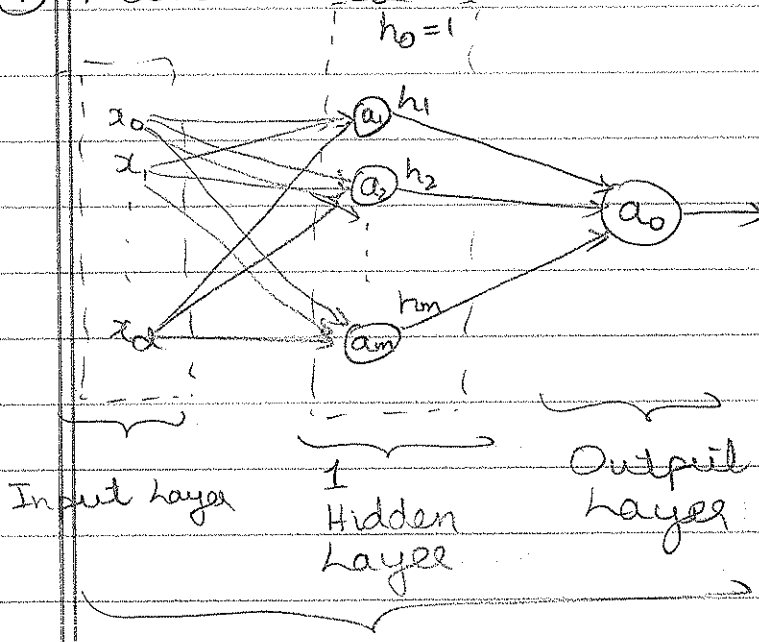
(c) Step $g(a) = \begin{cases} 1 & a \geq 0 \\ 0 & \text{else} \end{cases}$



$\hat{y} = \begin{cases} 1 & \text{if } w^T x \geq 0 \\ 0 & \text{if } w^T x < 0 \end{cases}$ Binary Classifier

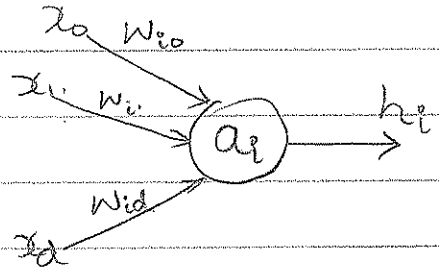
called "Perceptron" in this context.

④ Neural Network : Multi-layer Perceptrons



3-Layer NN

"Zoom-in"



→ Hidden activation $a_i = w_{i0} \cdot x_0 + w_{i1} x_1 + \cdots + w_{id} x_d$
$= \vec{w}_i \vec{x}$

→ Hidden response $h_i = g(a_i) = g(\vec{w}_i^T x)$

→ Stack all hidden responses into a vector

$$\vec{h} = \begin{bmatrix} h_0 = 1 \\ h_1 \\ \vdots \\ h_m \end{bmatrix}$$

→ Output Activation $a_o = w_{o0}h_0 + w_{o1}h_1 + \cdots + w_{om}h_m$
$= \vec{w}_o^T \vec{h}$

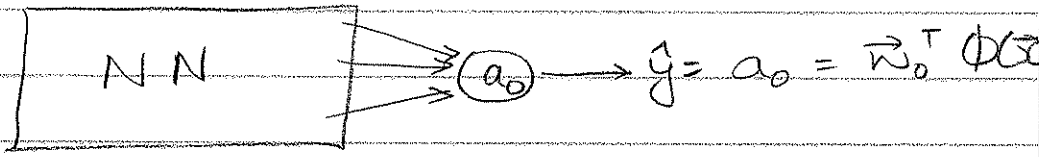
→ Output $\hat{y} = g(\vec{w}_o^T \vec{h})$

⑤ Output Layer & Response Functions

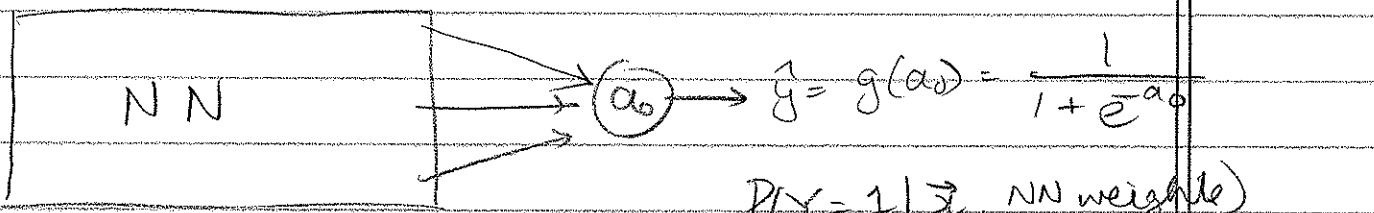
Think about a NN (till the penultimate layer) as a "feature extractor" converting $\vec{x} \to \phi(\vec{x})$
but unlike Kernel SVMs $\phi(\vec{x})$ is really $\phi(\vec{x}; \vec{w})$
↖ we will learn $\vec{w}$ to extract different features

The last layer in a NN will depend on the task

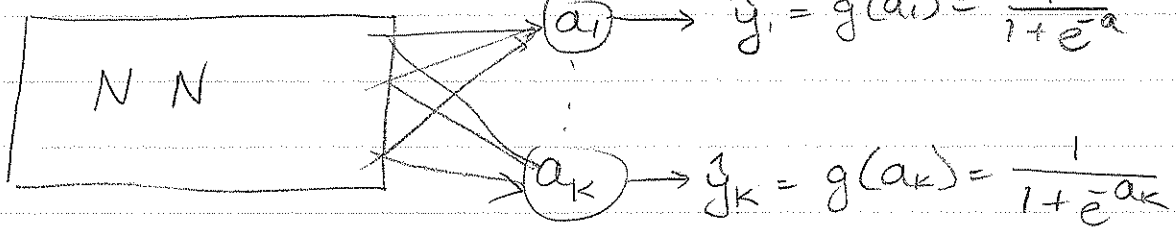
(a) Regression : 1 Output Unit; Linear Output Response



(b) Binary Classification: 1 Output Unit: Sigmoid Output Response



(c) Multi-label classification [Note: not multi-class]
K-Output Units ; each Sigmoid



→ An input can be assigned to multiple classes!

(d) Multi-class classification
K-Output Units : "soft-max" response