Math 390.4 Lec 3 2/5/19

$$y = g(\vec{x}) + \underbrace{h^*(\vec{x}) - g(\vec{x})}_{\text{estimation error}} + \underbrace{f(\vec{x}) - g(\vec{x})}_{\text{misspecification error}} + \underbrace{\epsilon(\vec{x}) - f(\vec{x})}_{\text{error due to ignorance}}$$

ϵ ("the residual") $\qquad \hat{y} = g(\vec{x}_*)$

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

- $\mathcal{D}$: known examples
- $\mathcal{L}$: set of candidate models / allowable model space

Let's see a full example ...

Back to the story about loans $y = \{0, 1\}$ (binary response)

- 1 will pay back loan (creditworthy)
- 0 will not pay back loan (not ...)

Want to build model g.

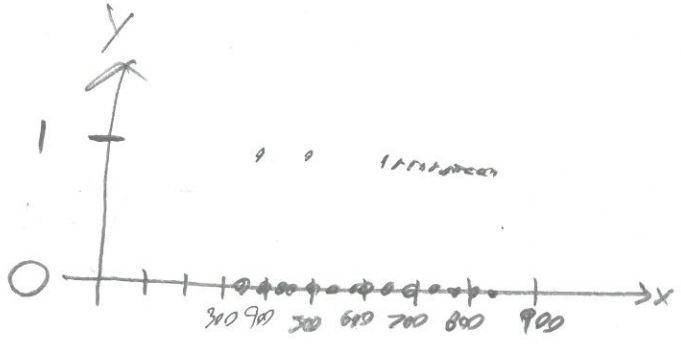
"Binary classification model", "classification model"

Imagine the one x: credit score. $\mathcal{X} = [300, 850]$

$\mathcal{D} = \langle X, y \rangle = \left\{ \begin{bmatrix} 810 \\ 390 \\ 750 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\}$ ← historical data for 4 different people.

Conventionally ... also ...

$= [X, y] = \begin{bmatrix} 810 & 1 \\ 390 & 0 \\ 750 & 1 \\ \vdots & \vdots \end{bmatrix}$



Null Model. No features! $g = \text{Mode}[y]$, $\mathcal{H} = Y = \{0,1\}$

Let us pick the simplest model space

$\mathbb{1}_{x \geq \theta} = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{else} \end{cases}$ indicator function

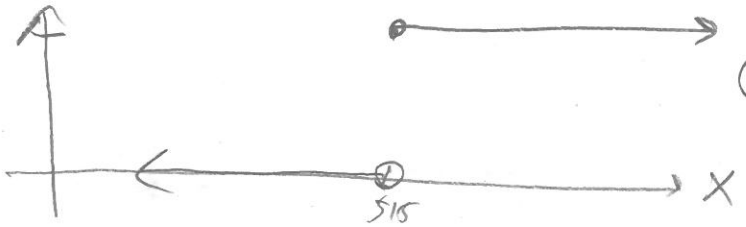
$$\mathcal{H} = \left\{ \mathbb{1}_{x \geq \theta} : \theta \in \mathbb{R} \right\}$$

"threshold model"

θ: parameter (greek variables $\theta, \beta, w, \ldots$)

$\mathbb{R}$: parameter space

e.g. $g(x) = \mathbb{1}_{x \geq 515.1} \in \mathcal{H}$, $g(x) = \mathbb{1}_{x \geq 611.3} \in \mathcal{H}$, etc...



these are the predictions that the model produces

the algorithm A produces g. Since g is fully specified by θ, A just needs to select a θ. How should selecting the θ that gives the least prediction errors on D?

error is when $y = 0$ but $\hat{y} = 1$ or when $y = 1$ but $\hat{y} = 0$

misclassification error

$$ME := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{g(\vec{x}_i) \neq y_i}$$

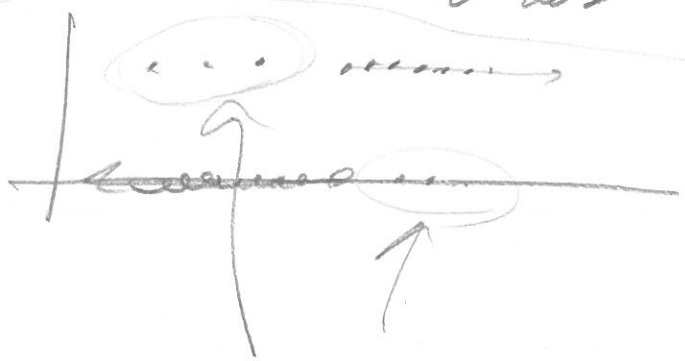
(y_i: true label; $\hat{y}_i = g(\vec{x}_i)$: predicted label)

accuracy

$$ACC := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{g(\vec{x}_i) = y_i} = 1 - ME$$

Minimizing error $\equiv$ (same) maximizing accuracy

A "fit" θ by considering all $\theta \in \mathbb{R}$ and finding the one that minimizes ME. ME is called an objective function, target function, fitness function, or thing to be optimized.



A does $\theta_g = \underset{\theta \in \mathcal{H}}{\text{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\mathbb{1}_{x_i \geq \theta} \neq y_i} \right\}$ correct but ugly! 3

Note the common $\frac{1}{n}$ won't affect the argmin given

errors. Are they residuals? The $y - \hat{y}$ would be the residuals

$$y = g(x) + e$$

$$y = \hat{y} + (y - \hat{y})$$

Can we rewrite A's objective function to be a function of $e_1, \ldots, e_n$?

$$ME = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n} \sum_{i=1}^{n} |e_i| = \frac{1}{n} \sum_{i=1}^{n} e_i^2$$

$|e_i|$: SAE; mean: MAE

e_i^2: SSE; mean: MSE

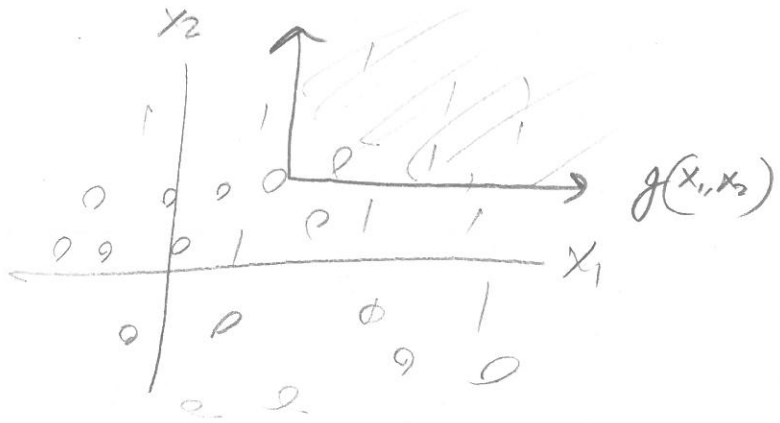
Minimizing any of these is equivalent in classification.

A: $g = \underset{h \in \mathcal{H}}{\text{argmin}} \{ SSE(h) \} \Leftrightarrow \theta_g = \underset{\theta \in \mathcal{H}}{\text{argmin}} \{ SSE(\theta) \}$ okay to rely on computer?

We have to specify $\mathcal{H}$. Ideas? $\mathcal{H} = \mathcal{X}$. All available values in D.

Let's now consider X_1, X_2 two parameters $\dim[\mathcal{H}] = 2$

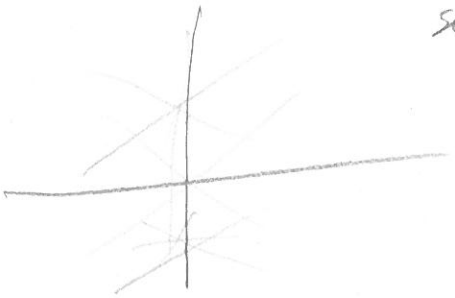
$$\mathcal{H} = \left\{ \mathbb{1}_{x_1 \geq \theta_1, x_2 \geq \theta_2} \; ; \; \theta_1, \theta_2 \in \mathcal{H} \right\}$$



$\mathcal{H}$ is very restrictive! Still could work well... but how about something less restrictive but still very simple? Linear?

$$\mathcal{H} = \{ \mathbb{1}_{x_2 \geq a + b x_1} : a, b \in \mathbb{R} \}$$

↑ ↑
slope and intercept give enough "degrees of freedom" for any separating line



we need an A to find a & b. We first reparameterize...

"intercept" or "bias" ↓ ↓ weight on x_1 ↓ weight on x_2

$$x_2 \geq a + b x_1 \Rightarrow -a - b x_1 + x_2 \geq 0 \Rightarrow w_0 + w_1 x_1 + w_2 x_2 \geq 0$$

Now we have 3 parameters, but we only needed 2. This is an "overparameterized" model. For each h, there are ∞ equivalent models... all its comb of $\vec{w}$ i.e. $c\vec{w}$ s.t. $c \in \mathbb{R}$

If we also set $\vec{x} = [1 \quad x_1 \quad x_2]$ ← add a 1 to each observation $\Rightarrow \vec{w} \cdot \vec{x} > 0$

$$\mathcal{H} = \{ \mathbb{1}_{\vec{w} \cdot \vec{x} \geq 0} : \vec{w} \in \mathbb{R}^3 \}$$

$p + 1 = 3$
↑ "for bias or intercept" term

Need A to find j. $j = \underset{k \in \mathcal{H}}{\operatorname{argmin}} \{SSE(k)\}$ equivalent to

$$\vec{w} = \underset{\vec{w} \in \mathbb{R}^3}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \mathbb{1}_{\vec{w} \cdot x_i \neq y_i} \right\}$$

Last time we checked only all thresholds in x. We can check all $\vec{w} \in \mathbb{R}^3$!

Can you solve it analytically? No... indicator function isn't differentiable.

Consider the following algorithm called the perceptron learning algorithm (1957): For the general case of p features:

1. Initialize $\vec{w}^{t=0} = \vec{0}$ or random
2. Compute $\hat{y}_i = \mathbb{1}_{\vec{w}^{t=0} \cdot \vec{x}}$
3. For $j = 0, 1, \ldots, p$ $\quad e_i$ $\quad$ if $e_i = 1$ need to make $\vec{w} \cdot \vec{x}$ larger

$$w_0^{t=1} = w_0^{t=0} + (y_i - \hat{y}_i)(1)$$

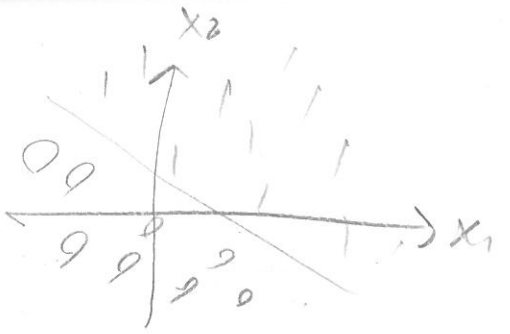
$$w_1^{t=1} = w_1^{t=1} + (y_i - \hat{y}_i)(x_{i,1})$$

$$\vdots$$

$$w_p^{t=1} = w_p^{t=1} + (y_i - \hat{y}_i)(x_{p,1})$$

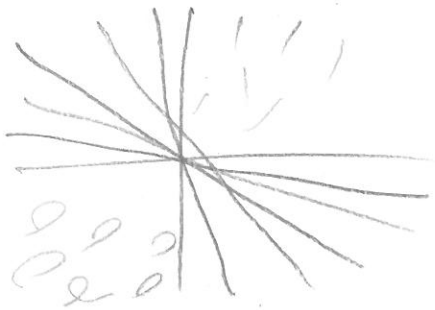
if $e_i = 0$ .... smaller

4. Repeat steps 2 & 3 for $i = 1, \ldots, n$
5. / / / / 2, 3, 4 until no errors or until a prespecified max # of iterations



The perceptron is proven to converge if the y's are linearly separable. If not, it will fail!!

Besides solutions being equal to a multiplicative constant, there are infinitely many solutions that give linear separability e.g.



Which one is best? The one that divides the margin evenly.

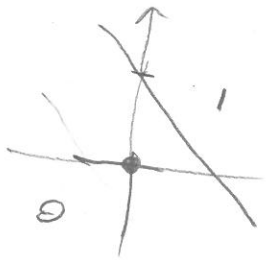
Weaknesses

① Needs lin. sep.

② Cannot find "best" lin sep. model

Simple

i	x_1	x_2	y
1	-1	-1	0
2	1	1	1



$\vec{w}_{t=0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$\hat{y}_1 = \mathbb{1}_{\vec{w}\cdot\vec{x}\geq 0} = 1$

$t=1, i=1$

$w_0 = (0) + (-1)(1) = -1$

$w_1 = (0) + (-1)(-1) = 1$

$w_2 = (0) + (-1)(1) = -1$

$t=1, i=2$

$w_0 = (1) + (0)$

$w_1 = (1) + (0)$

$w_2 = (-1) + (0)$

$\hat{y}_2 = \mathbb{1}_{1+(1)\cdot(1)\geq 0} = 1$

$t=2, i=1$

$w_0 = (-1) + (0)$

$w_1 = (1) + (0)$

$w_2 = (-1) + (0)$

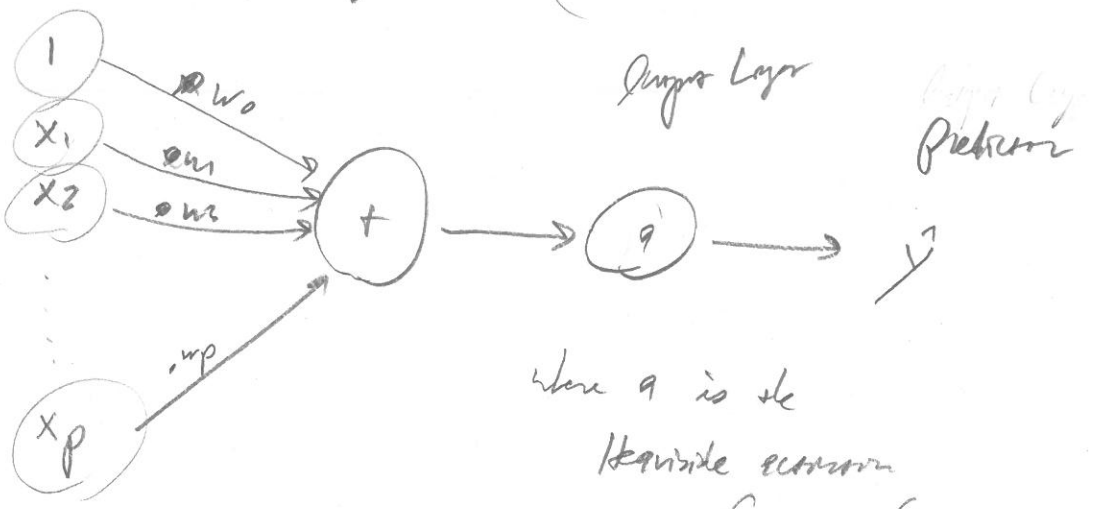
$\hat{y}_1 = \mathbb{1}_{-1+(0)(-1)+(-1)(1)\geq 0} = 0$

✓ converged

$\hat{y} = \mathbb{1}_{-1 + x_1 - x_2 \geq 0}$

$\Rightarrow x_2 \geq +1 - x_1$

Diagram of perceptron (a single layer neural network)



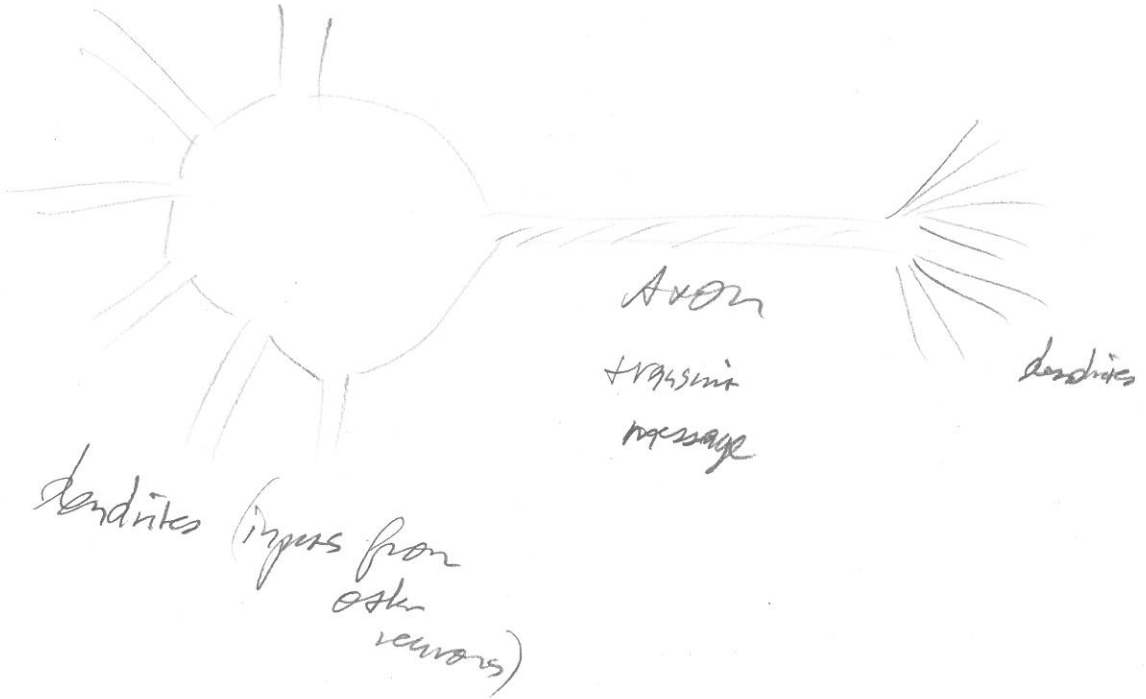
Output Layer

Prediction

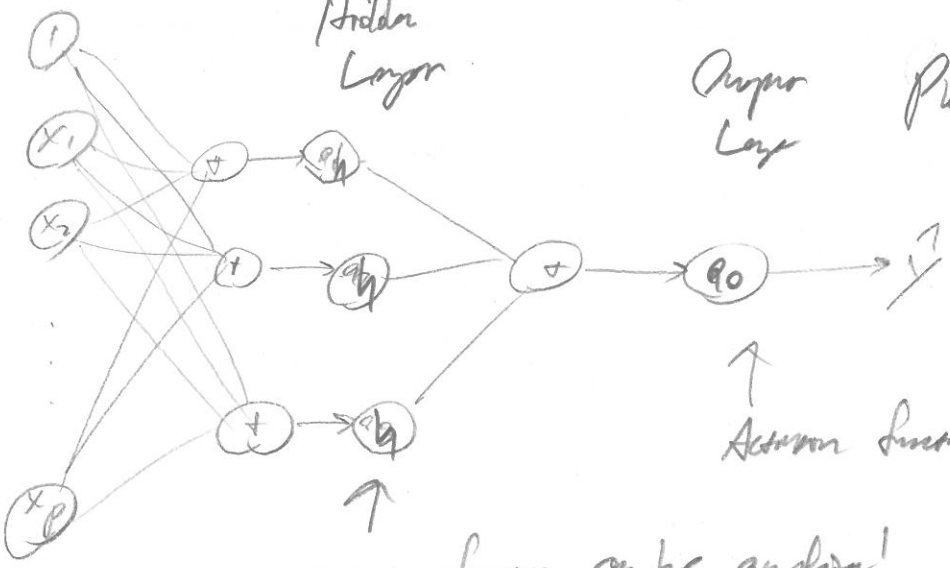
where g is the Heaviside activation function (the indicator)

Input Layer

Why called neural network? Inspired by neuron..



Deep Learning



Let's attempt to fix the problems with the perceptron

(1) Finds "best" model out of all models if lin. sep

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