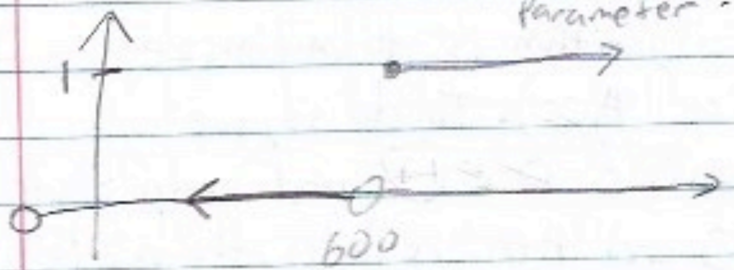
* starts the class with some code until 35 min.

Lec 3

$$\mathcal{H} = \{ \mathbb{1}_{x \geq \theta} : \underset{\substack{\uparrow \\ \text{model} \\ \text{parameter.}}}{\theta} \in \underset{\substack{\uparrow \\ \text{parameter space}}}{\Theta} \}$$



prediction

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

$$y = g(\vec{x}) + e = \hat{y} + e = \hat{y} + \overset{e}{\overset{\parallel}{(y - \hat{y})}}$$

The algorithm $\mathcal{A}$ produces g. Since g is fully specified by θ, the algorithm selects/estimates/optimizes/fits a θ. Let's create an algorithm. A bad algorithm will have high estimation error.

$$\begin{array}{cc|cc|c} & & \hat{y} & & \\ & & 0 & 1 & \\ \hline y & 0 & 0 & -1 & e \\ & 1 & +1 & 0 & \end{array}$$

Let's define an overall error function/objective function called "misclassification error" (ME) $ME = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{g(\vec{x}_i) \neq y_i} = \frac{1}{n} \sum_{i=1}^{n} |e_i|$

⟶

Or accuracy (ACC) as

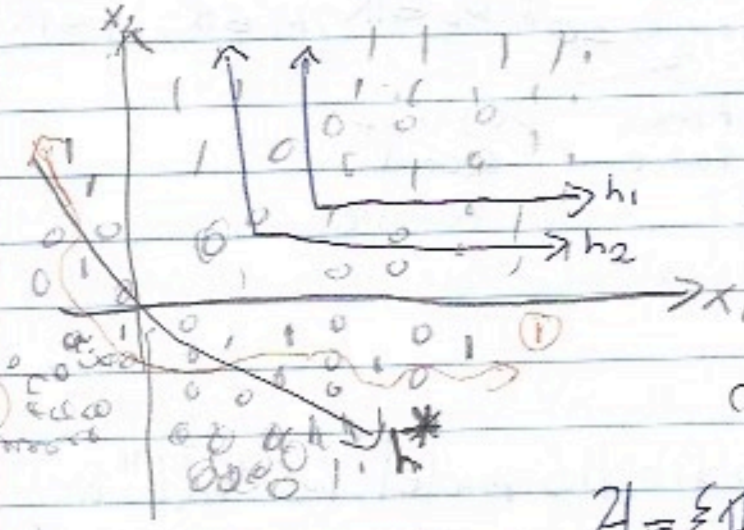
$$ACC = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{g(\vec{x}_i) = y_i} = 1 - ME$$

Goal of the algorithm is to minimize ME (or maximize ACC). To do so, we check every possible $\theta \in \Theta$ and keep track of the $ME(\theta)$ and then return the model w/ the lowest ME.

How to define parameter space? It must be finite b/c we need to check (ie compute ME) each element. Gabriel says grid up $[300, 850]$ e.g. $\{351, 352, \ldots, 849, 850\}$. That's fine, but it's more convenient to only check the unique values of x.

$\mathcal{A}$ produces $g(x) = \mathbb{1}_{x \geq \text{argmin}\left\{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\mathbb{1}_{x_i \geq 0} \neq y_i}\right\}}$

Let's make a loan model w/ two continuous x's i.e. x_1, x_2 $(p=2)$



$$\dim[\Theta] = 2 = p$$

A 2-dimensional threshold model extending what we have before has candidate set:

$$\mathcal{H} = \left\{ \mathbb{1}_{x_1 \geq \theta_1} \mathbb{1}_{x_2 \geq \theta_2} : \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \in \Theta \right\}$$

This candidate set of "angle bracket" looking things is very restrictive! Which means we will probably have high misspecification error. Let's use another hypothesis-set: all lines.

$$\mathcal{H} = \left\{ \mathbb{1}_{x_2 \geq \underset{\text{intercept}}{\underset{\uparrow}{a}} + \underset{\text{slope}}{\underset{\uparrow}{b}} x_1} : a \in \mathbb{R}, b \in \mathbb{R} \right\}$$

The slope and intercept provide you w/ enough "degree of freedom" to specify any seperating line. We need an algorithm to find g i.e. to specify a and b. This is a hard problem so we will study it w/ different conditions.

→

We will reparameterize the hypothesis space to be:

$$\mathcal{H} = \left\{ \mathbb{1}_{\overset{\overset{\mathbb{1}_{\vec{w}\cdot\vec{x} \geq 0}}{\parallel}}{w_0 + w_1 x_1 + w_2 x_2 \geq 0}} : w_0 \in \mathbb{R}, w_1 \in \mathbb{R}, w_2 \in \mathbb{R} \right\}$$

weight of the first feature

weight of second feature

intercept term
or "bias"

In order to fit this model, we "add" a dummy value of 1 to each data record:

$$\vec{x} = [750 \ \$58000] \longrightarrow \vec{x} = [1 \ 750 \ \$58000]$$

So we append the $\vec{1}$, the n-dim column vector to x, the matrix of features in $\mathbb{D}$.

We only need 2 parameters (a,b) but here we have three (w_0, w_1, w_2) and hence we are "over-parameterized" meaning we have infinite solutions seen here:

$x_1 + x_2 = 7$

$$\mathbb{1}_{\vec{w}\cdot\vec{x} \geq 0} = \mathbb{1}_{c\vec{w}\cdot\vec{x} \geq 0} \quad \forall c \neq 0$$

pin

⟶

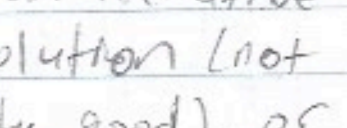
A: find w_0, w_1, w_2 to minimize ME i.e.

$$\vec{w}_* := \underset{\vec{w}\in\mathbb{R}^3}{\arg\min}\left\{\sum_{i=1}^{n} \mathbb{1}_{\mathbb{1}_{\vec{w}\cdot\vec{x}_i \geq 0} = y_i}\right\} = \arg\min\{ME$$

We have a problem here. There is no analytic solution since the indicator function is non-differenti

We need a way to search over all possible lines. So (1) we need to reduce the # of lines like before, (2) use an iterative algorithm to find a local solution (not the best but hopefully pretty good), or (3) change our objective function.

In the setting of perfect linear seperability e.g. where ME of that linear discrimination model is zero (i.e no errors). Consider the 1957 Pereception iterative algorithm for p features:



Step 1: Initialize $\vec{w}^{t=0} = \vec{0}_{p+1}$ or to a random vector value.

Step 2: Compute $\hat{y}_i = \mathbb{1}_{\vec{w}^{t=0} \cdot \vec{x}_i \geq 0}$

Step 3: For $j = 0, 1, \cdots, p$ set

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

$$w_1^{t=1} = w_1^{t=0} + (y_i - \hat{y}_i)(x_{i,1})$$

$$\vdots$$

$$w_p^{t=1} = w_p^{t=0} + (y_i - \hat{y}_i)(x_{i,p})$$

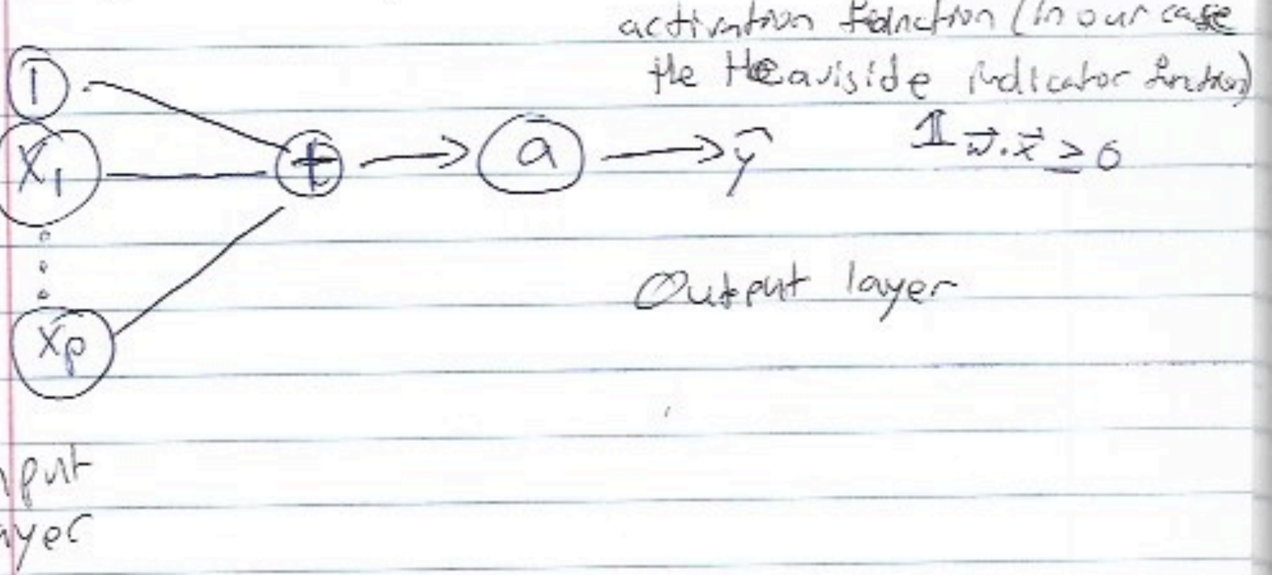
Step 4: Repeat steps 2 and 3 for $i = 1, \cdots, n$ (all the obs.).

Step 5: Repeat steps 2, 3 and 4 until $ME = 0$ i.e. all e_i's $= 0$ or until a prespecified (large) # of iterations.

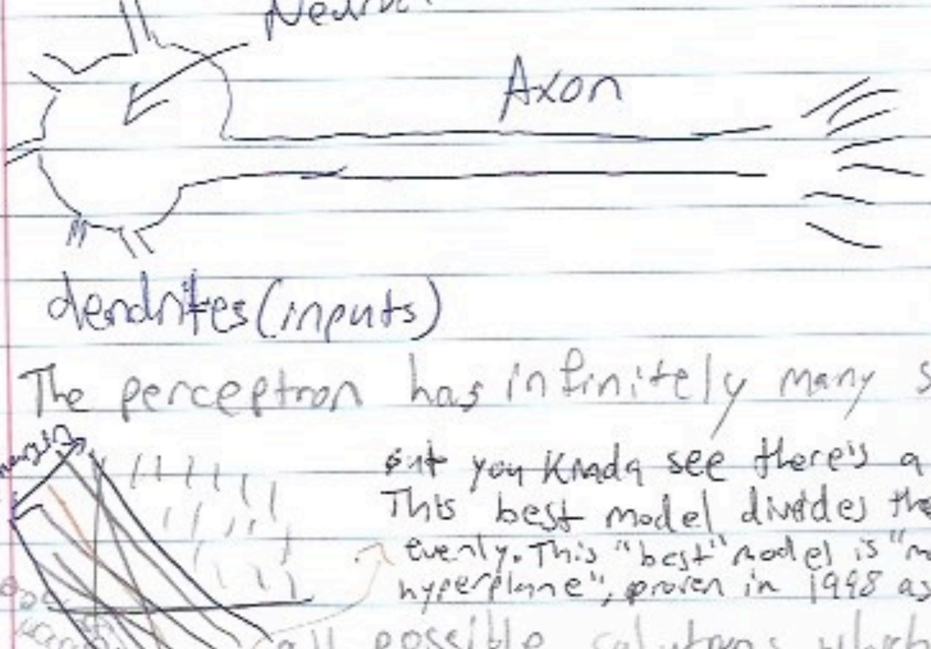
The perception is proved to converged for linearly separable datasets but for non-linearly seperable datasets, anything can happen so it may fail.

→

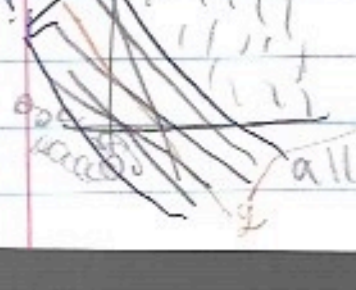
Diagram of perception:


activation function (in our case
the Heaviside indicator function)
$\mathbb{1}_{\vec{w}\cdot\vec{x}\geq 0}$
1
X_1
X_p
+
a
$\hat{y}$
Output layer
input
layer

The perception is a type of "neural network" model. So are deep learning models. They're called neurons since they kind of act like neurons.



The perceptron has infinitely many solutions



but you kinda see there's a best model. This best model divides the margin (wedge) evenly. This "best" model is "maximum margin hyperplane", proven in 1998 as optimal linear classifier

all possible solutions which vary based on starting values