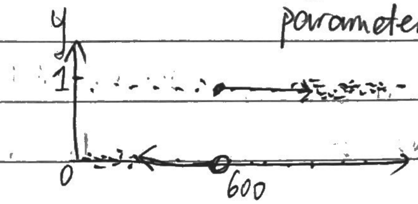
lec #3

$$\mathcal{H} = \{ \mathbb{1}_{x \geq \theta} : \theta \in \Theta \}$$

θ ↑ model parameter

Θ ↖ parameter space



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

$$y = g(\vec{x}) + e = \hat{y} + e = \hat{y} + \underbrace{(y - \hat{y})}_{e}$$

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

$y \backslash \hat{y}$	0	1
0	0	-1
1	+1	0

e

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

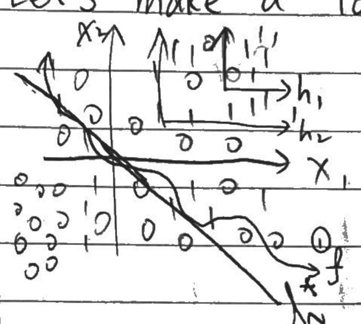
goal of the algorithm is to minimize ME (or maximize Acc).

To do so, we check every possible $\theta \in \Theta$ and keep track of the ME(theta) and then return the model with the lowest ME.

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

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

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



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

A two 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 brace"-looking things is very restrictive! which means we'll probably have high misspecification error. Let's use another hypothesis set: all lines.

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

The slope and intercept provide you with enough "degree of freedom" to specify any separating 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 with different conditions.

We'll first reparameterize the hypothesis space to be:

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

intercept term or "bias"   weight of the 1st feature, weight of the 2nd feature

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

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

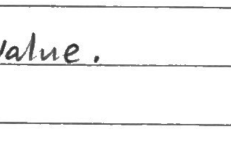
So we append the $\vec{1}$, the n-dimension column vector to X, the matrix of features in $\mathbb{D}$.

We only need 2 parameters (a, b) but here we are "over-parameterized" meaning we have infinite solutions seen here: $\mathbb{1}_{\vec{w}\cdot\vec{x} \geq 0} = \mathbb{1}_{c\vec{w}\vec{x} \geq 0} \quad \forall c \neq 0$

$\mathcal{A}$: find w_0, w_1, w_2 to minimize ME i.e.

$$\vec{w}_* = \text{argmin} \left\{ \sum_{i=1}^{n} \mathbb{1}_{\mathbb{1}_{\vec{w}\cdot\vec{x}_i \geq 0} = y_i} \right\} = \text{argmin} \{ME\}$$

We have a problem here. There is no analytic solution. We need a way to search over all possible lines. So (1) we need to reduce the # of lines like before, (2) Use an iteriative 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:

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

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

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

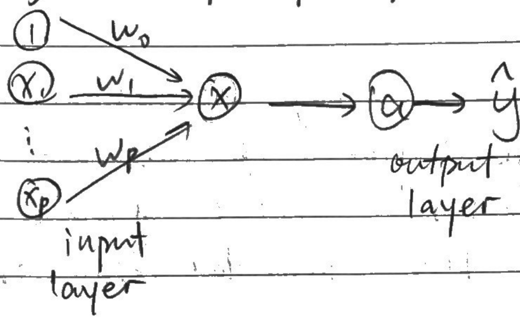
$$w_0^{t=1} = w_0^{t=0} + (y_i - \hat{y}_i)(1)$$
$$w_1^{t=1} = w_1^{t=0} + (y_i - \hat{y}_i)(x_{i,1})$$
$$\vdots$$
$$w_p^{t=1} = w_p^{t=0} + (y_i - \hat{y}_i)(x_{i,p})$$

4: Repeat steps 2, 3 for $i = 1, \cdots, n$ (all the observations)

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

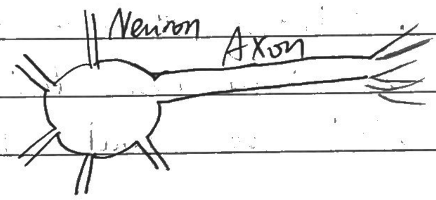
The perception is proved to converged for linearly separable datasets but for non-linearly separable datasets, anything can happen so it may fail

Diagram of perception:

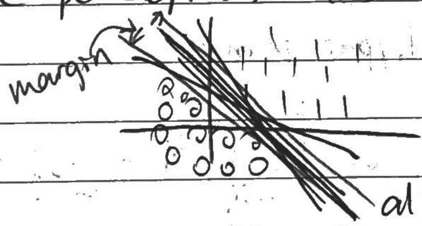


activation function (in our case the heaviside indicator function)

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



The perceptron has infinitely many solutions

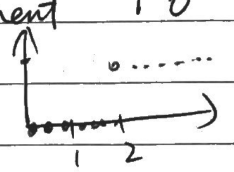


all possible solutions which vary based on starting values.

But you kinda see there's a "**best**" model. This "best" model divides the margin (AKA wedge) evenly. This "best" model is called the "maximum margin hyperplane" and it was proven in 1998 to be the optimal linear classifier

$$\mathbb{1}_{\text{statement}} := \begin{cases} 1 & \text{if statement is true} \\ 0 & \text{if statement is false} \end{cases}$$

$$g(x) = \mathbb{1}_{x>2}$$



not differentiable