Lecture #25: The Perceptron Algorithm

First, some hype about what we will cover:

Electronic 'Brain' Teaches Itself

The Navy last week demonstrated the embryo of an electronic computer named the Perceptron which, when completed in about a year, is expected to be the first non-living mechanism able to "perceive, recognize and identify its surroundings without human training or control." Navy officers demonstrating a preliminary form of the device in Washington said they hesitated to call it a machine because it is so much like a "human being without life."

Dr. Frank Rosenblatt, research psychologist at the Cornell Aeronautical Laboratory, Inc., Buffalo, N. Y., designer of the Perceptron, conducted the demonstration. The machine, he said, would be the first electronic device to think as the human brain. Like humans, Perceptron will make mistakes at first, "but it will grow wiser as it gains experience," he said.

The first Perceptron, to cost about $100,000, will have about 1,000 electronic "association cells" receiving electrical impulses from an eyelike scanning device with 400 photocells. The human brain has ten billion responsive cells, including 100,000,000 connections with the eye.

Difference Recognized

recognize the difference between right and left, almost the way a child learns.

When fully developed, the Perceptron will be designed to remember images and information it has perceived itself, whereas ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons, Dr. Rosenblatt said, will be able to recognize people and call out their names. Printed pages, longhand letters and even speech commands are within its reach. Only one more step of development, a difficult step, he said, is needed for the device to hear speech in one language and instantly translate it to speech or writing in another language.

Self-Reproduction

In principle, Dr. Rosenblatt said, it would be possible to build Perceptrons that could reproduce themselves on an assembly line and which would be "conscious" of their existence.

Perceptron, it was pointed out, needs no "priming." It is not necessary to introduce it to surroundings and circumstances, record the data involved and then store them for future comparison as is the case

Main Question: How do we find a linear separator?

i.e. it should have all +s on one side and all −s on the other

To keep it simple, will ignore offset: $w^T x \cancel{+b} \geq 0$

In 1958 Rosenblatt discovered a simple but powerful algorithm

<u>Setting</u>: Suppose we are given a sequence of examples $(x_1, y_1), (x_2, y_2), \ldots$

vector in $\mathbb{R}^d$ label, either $+1$ or -1

Perceptron Algorithm:

Initialize $w_0 = 0$

vector in $\mathbb{R}^d$

For each training example

Predict: $y_i' = \operatorname{sgn}(w_t^T x_i)$; $\operatorname{sgn}(c) = \begin{cases} +1 & c > 0 \\ -1 & \text{else} \end{cases}$

If $y_i' \neq y_i$

then **Update** $w_{t+1} \leftarrow w_t + \eta y_i x_i$

Return final weight vector

Here η is called the <u>learning rate</u>, and $0 < \eta \leq 1$

Note: we only update the weight when we make a mistake ($y_i' \neq y_i$)

Let's get some intuition for what this is doing

Thought Experiment: Suppose we make a mistake on (x_i, y_i)

assume, for simplicity, $y_i = +1$

Then the new weight vector is

$$w_{t+1} = w_t + \eta x_i$$

Thus the new inner product is

$$w_{t+1}^T x_i = (w_t + \eta x_i)^T x_i = \underbrace{w_t^T x_i}_{\text{old inner product}} + \underbrace{\eta \|x_i\|^2}_{>0}$$

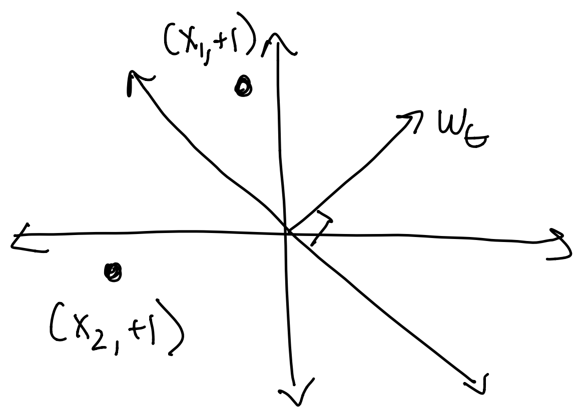
Since $y_i = +1$, we want $w_{t+1}^T x_i > 0$. Before it was negative, so we push it closer to positive

Or, intuitively:

Perceptron Algorithm = If we make a mistake, try to do better on that example

Let's visualize it in 2-d

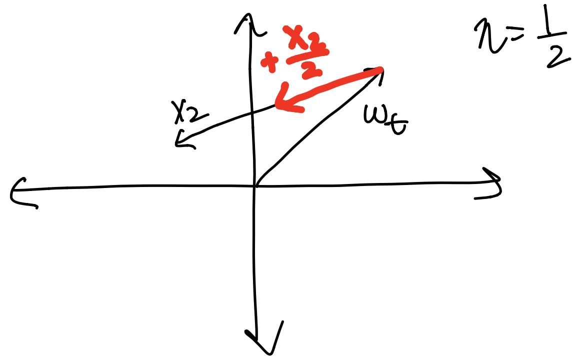
<u>Decision rule after t mistakes:</u>



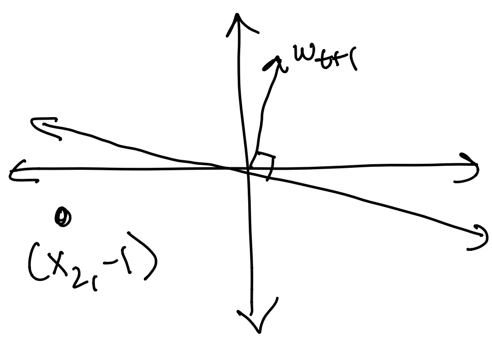
<u>Q</u>: On which of these examples does our current rule make a mistake?

x_2, because it's on the wrong side given that its label +1

So how do we update the weight vector?



Thus our new rule is

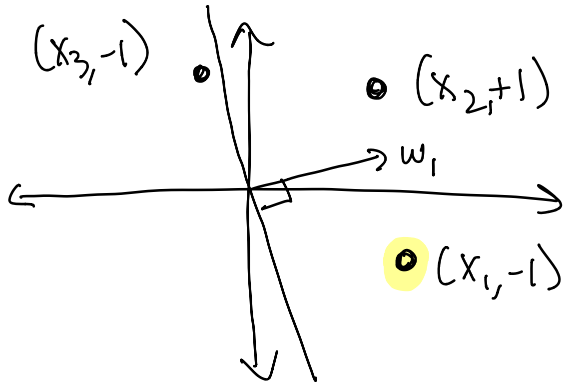


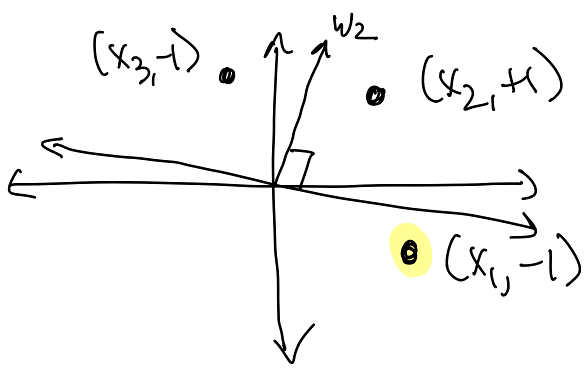
So now $(x_2, -1)$ is closer to being on the correct side

It is not clear how to analyze the algorithm

Q2: How do we know we're not doing worse on the examples we previously got right?

Concrete Example:





Uh-oh! We might just keep cycling

Q3: What went wrong?

There is no line thru the origin that gets everything right

Observation: The perceptron algorithm can't work if there is no linear separator in the first place!

But if there is a linear separator with margin then it will!

Margin Assumption: Suppose there is a unit vector u that satisfies

(1) For all x_i labelled $+1$ we have

$$u^T x_i \geq \gamma$$

(2) For all x_i labelled -1 we have

$$u^T x_i \leq -\gamma$$

Or to put it more succinctly, for any example

$$y_i (u^T x_i) \geq \gamma$$

Under the margin assumption, we make few mistakes:

<u>Theorem</u>: Suppose that the margin assumption holds and furthermore each example x_i satisfies

$$\|x_i\| \le R$$

Set $\eta = 1$. Then the <u># of mistakes</u> is at most

$$\left(\frac{R}{\gamma}\right)^2$$

we just need two simple ingredients:

<u>Claim 1</u>: After t mistakes

$$u^T w_t \ge t\gamma$$

Consider what happens on our first mistake

$$u^T w_{t+1} = u^T w_t + y_i u^T x_i$$

$$\ge u^T w_t + \gamma \qquad \text{use margin condition}$$

$w_0 = 0$ so $u^T w_0 = 0$. Thus every time we make a mistake $u^T w_t$ increases by $\ge \gamma$

Claim 2: After t mistakes
$$\|w_t\|^2 \le tR^2$$

Again, let's see what happens when we make a mistake:

$$\|w_{t+1}\|^2 = \|w_t + y_i x_i\|^2$$
$$= (w_t + y_i x_i)^T (w_t + y_i x_i)$$
$$= \underbrace{w_t^T w_t}_{\|w_t\|^2} + 2y_i w_t^T x_i + \underbrace{x_i^T x_i}_{\|x_i\|^2 \le R^2}$$
$$\le \|w_t\|^2 + \underbrace{2y_i w_t^T x_i}_{\text{I claim } < 0.\text{ why?}} + R^2$$

B/c it's a mistake!

Thus $\|w_{t+1}\|^2 \le \|w_t\|^2 + R^2$

And so everytime we make a mistake the length squared increases by at most R^2

Now we can put it all together:

$R\sqrt{t} \geq \|w_t\|$ from Claim 2

$\|w_t\| \geq u^T w_t$ b/c u is a unit vector, $u^T w_t = \|w_t\| \cos\theta$

Finally $u^T w_t \geq t\gamma$ from Claim 1

Thus we have:

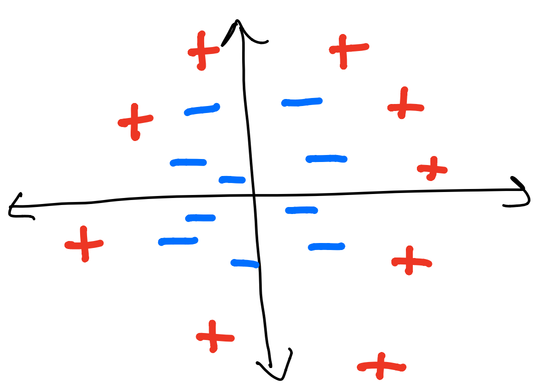
$$R\sqrt{t} \geq t\gamma \Rightarrow \frac{R}{\gamma} \geq \sqrt{t} \Rightarrow \left(\frac{R}{\gamma}\right)^2 \geq t$$

# of mistakes

Last topic for today:

Q4: What can we do if there is no linear separator?

For example:



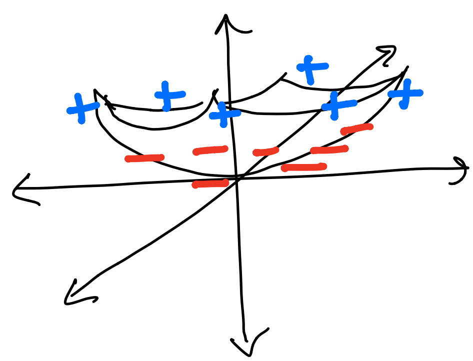
Idea: Map to a higher dimensional space

Imagine we take a 2-d example

$$\begin{bmatrix} x \\ y \end{bmatrix} \longmapsto \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

But how do we choose z?

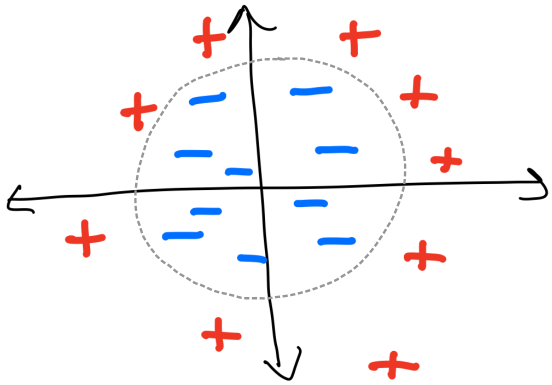
Let's set $z = x^2 + y^2$. Then we get



Q5: Is there a linear separator now?

Yes, there is one using the z coordinate

What does it look like for the original data?



This is called a **kernel**: There is a lot of linear algebra to finding/using them, but beyond the scope of this class

Main Point Linear algebra is not always limited to doing linear things