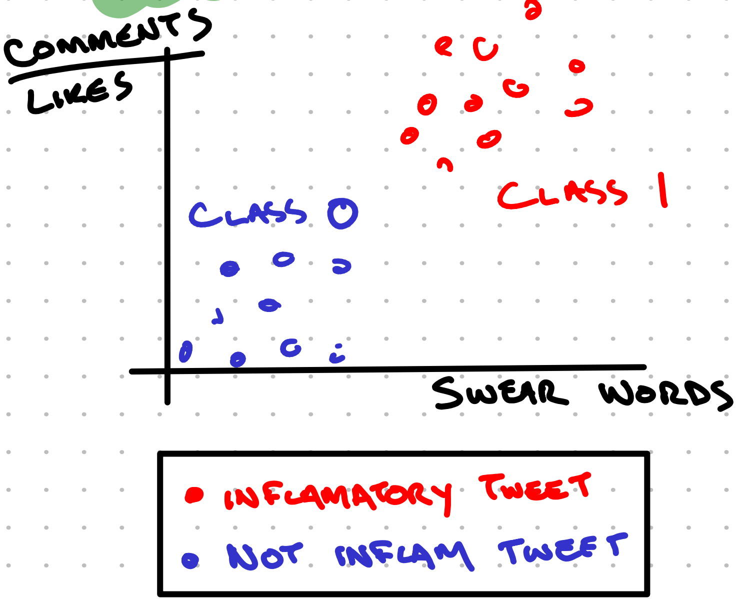
CS 2810 DAY 5 FEB 1

→ LINEAR PERCEPTRON

LINEAR PERCEPTRON



A FUNCTION WHICH ESTIMATES ONE OF TWO CLASSES, DEFINED BY VECTOR $\vec{w}$

VECTOR

ESTIMATED CLASS

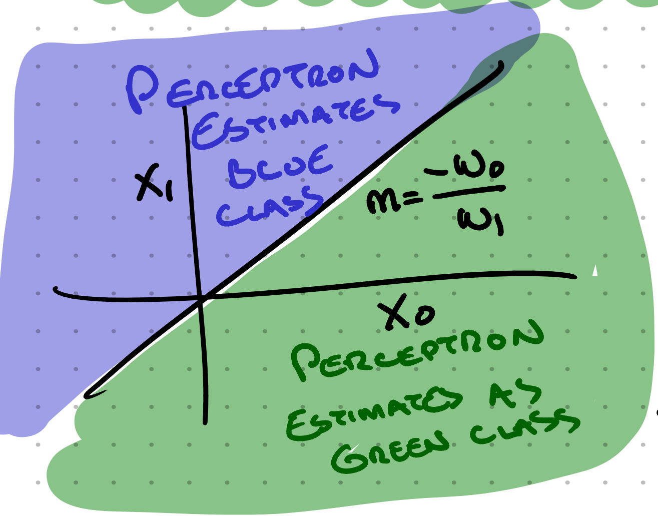
$$f(\vec{x}) = \begin{cases} 1 & \text{IF } x \cdot w \geq 0 \\ 0 & \text{OTHERWISE} \end{cases}$$

ICA 1

WHAT IS PERCEPTRON'S CLASS ESTIMATE IF PERCEPTRON HAS $w = \begin{bmatrix} -1 \\ 4 \end{bmatrix}$?

$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

LINEAR PERCEPTRON: DECISION BOUNDARY



GOAL: VISUALIZE ALL POINTS WHICH ARE "CLOSE" TO BEING CLASSIFIED AS EITHER CLASS 0 OR CLASS 1

$$x \cdot w = 0 \iff \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 0$$

$$\iff x_0 w_0 + x_1 w_1 = 0$$

$$\iff x_1 = -\frac{w_0}{w_1} x_0$$

ICA 2

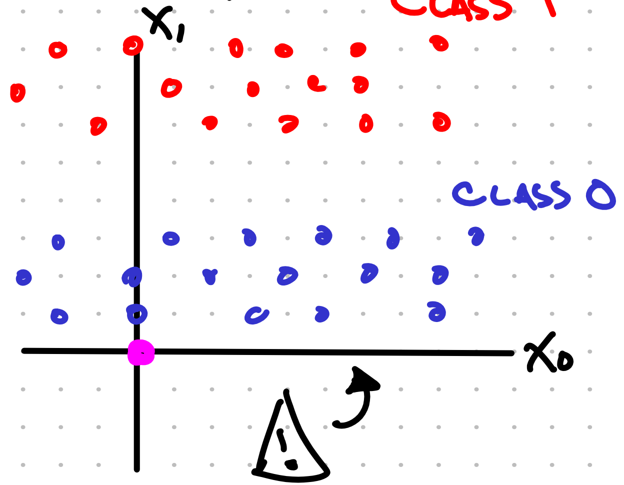
Find a perceptron weight $\vec{w}$ which distinguishes all samples below (scatters drawn to scale)

Big problem:

All boundaries pass through origin

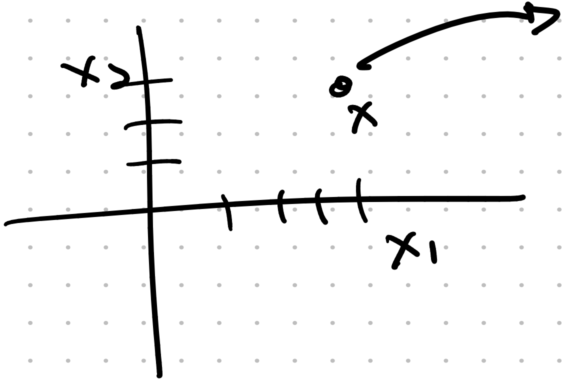
$$x_1 = -\frac{w_0}{w_1} x_0$$

$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$ always on line



ADDING BIAS TERM TO DATA

LETS REPRESENT OUR DATA AS VECTOR WHOSE FIRST ENTRY IS ALWAYS 1



OLD

$$x = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

NEW

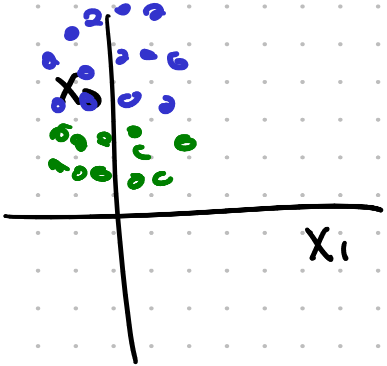
$$x = \begin{bmatrix} 1 \\ 4 \\ 3 \end{bmatrix}$$

BY CONVENTION, BIAS IS ALWAYS IN FIRST POSITION

Revisiting Decision Boundary

Remember: Boundary is all points with

$$x \cdot w = 0$$



Bias

$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = 0$$

Slope-intercept

$$x_2 = b + m x_1$$

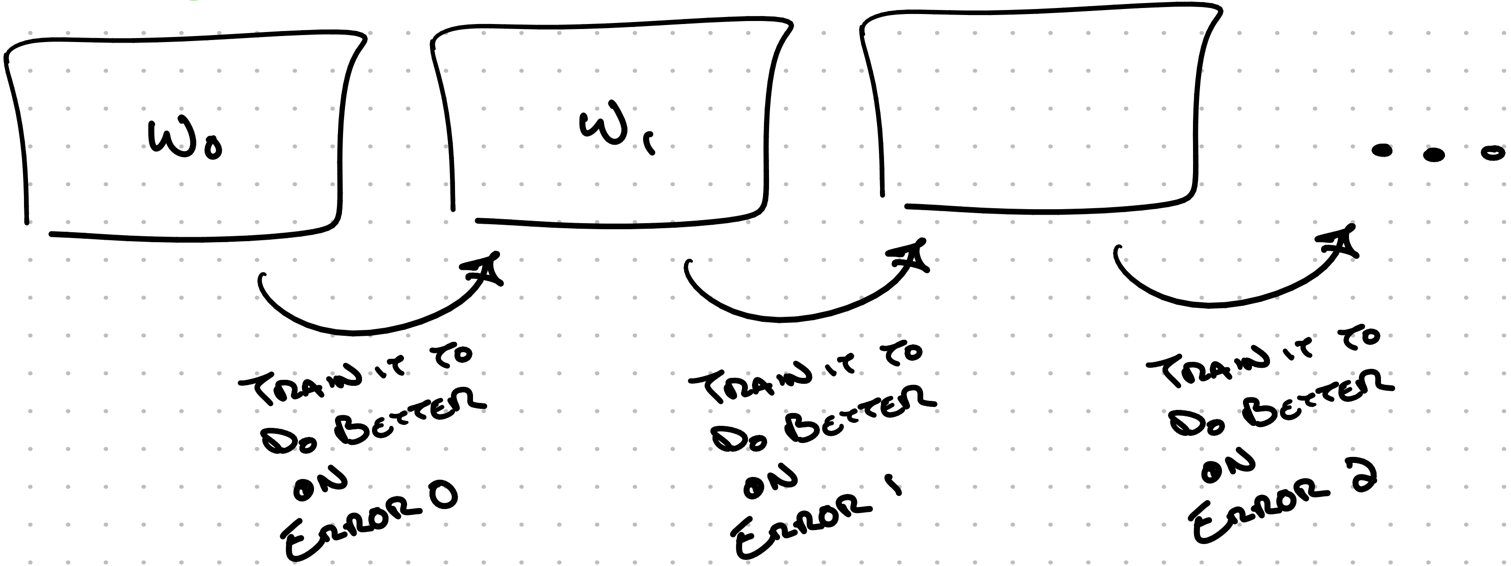
$$w_0 + w_1 x_1 + w_2 x_2 = 0 \iff x_2 = -\frac{w_0}{w_2} - \frac{w_1}{w_2} x_1$$

Let's summarize:

- A Perceptron creates a line boundary between two regions
    - estimate x as class 1 if x dot w is positive
    - we add a "bias" to x to allow for more flexible boundary lines
        - needn't pass through origin

Next Question:
- how do I find a good w for my data?
    (we don't really compute explicitly, ICA3 was just for practice)

Machine Learning: Reinforcement Learning



Penguin Demo

BIG QUESTION

HOW DO WE ADJUST w TO DO BETTER ON SOME ERROR?

CENTRAL IDEA OF LESSON

Aside Length and Dot Product

$$\|x\| = \sqrt{\sum_i x_i^2}$$

$$x \cdot x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \end{bmatrix} = x_0^2 + x_1^2 + \ldots$$

$\|x\|^2 = x \cdot x$ Always Positive

Aside 2 Dot Product is Linear

$$x \cdot (y+z) = x \cdot y + x \cdot z$$

Addition before dot product with x

Addition after dot product with x

Aside 2: Dot Product is Linear

$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $y = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ $z = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$

$x \cdot (y + z) = x \cdot y + x \cdot z$

An example (not proof)

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \left(\begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} \right)$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

UPDATING PERCEPTRON WEIGHTS

Problem:
sample x is estimated as class 0 with our current w but sample x really belongs to class 1

WE HAVE $x \cdot w < 0$

WE WANT $x \cdot w' \geq 0$
FOR SOME NEW w'

Algebraically, x dot w is too small, we'd like it to be bigger

UPDATING PERCEPTRON WEIGHTS

Problem:
sample x is estimated as class 0 with our current w but sample x really belongs to class 1

WE HAVE $x \cdot w < 0$

WE WANT $x \cdot w' \geq 0$ FOR SOME NEW w'

Algebraically, x dot w is too small, we'd like it to be bigger

$w' = w + x$

NEW w OLD w

THEN $x \cdot w' = x \cdot (w + x)$

ASIDE 2 $= x \cdot w + x \cdot x$

$= x \cdot w + \|x\|^2 > x \cdot w$

ASIDE 1

UPDATING PERCEPTRON WEIGHTS (OPPOSITE CASE)

Problem:
sample x is estimated as class 1 with our current w but sample x really belongs to class 0

WE HAVE $x \cdot w \geq 0$

WE WANT $x \cdot w' < 0$ FOR SOME NEW w'

Algebraically, x dot w is too large, we'd like it to be smaller

$w' = w - x$

NEW w — OLD w

THEN $x \cdot w' = x \cdot (w - x)$

ASIDE 2 $= x \cdot w - x \cdot x$

$= x \cdot w - \|x\|^2 < x \cdot w$

ASIDE 1

LOOK AT

UPDATE_PERCEPTRON()

- A Perceptron creates a line boundary between two regions
    - estimate x as class 1 if x dot w is positive
    - we add a "bias" to x to allow for more flexible boundary lines
        - needn't pass through origin

- A Perceptron can be trained by updating w per every x found in error

    w = w + x if class estimate = 0 but true class = 1

        (new w will have bigger dot product with x)

    w = w - x if class estimate = 1 but true class = 0

        (new w will have smaller dot product with x)

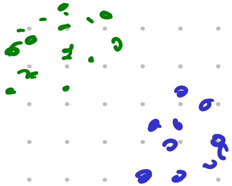
Next Questions:

- When do I stop training?
- Changing the units of a feature causes problems
    - why?
    - what can we do to avoid these issues?

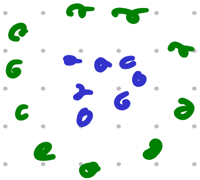
STOPPING CRITERIA

A DATASET IS LINEARLY SEPERABLE IF A LINE SEPERATES CLASSES PERFECTLY

LINEARLY SEPERABLE



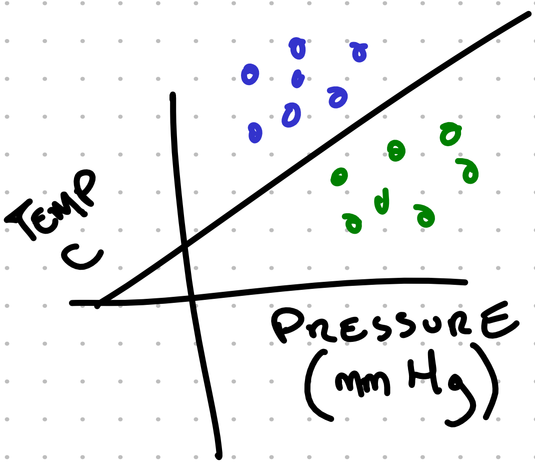
NOT LINEARLY SEPERABLE



IF DATA IS LINEARLY SEPERABLE WE SHOULD RUN TRAINING UNTIL WE FIND A 100% ACCURATE BOUNDARY

DEFINE "LOSS" FUNCTION (HOW MANY WRONG) AND STOP WHEN YOU THINK LOSS IS MINIMIZED

Scale (if time)

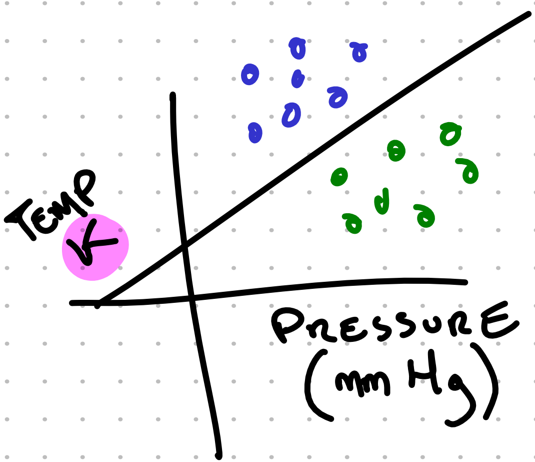


$$X = \begin{bmatrix} \text{Temp C} \\ \text{Pressure} \end{bmatrix}$$

$$X \cdot W = \text{Temp C} \cdot W_0 + \text{Pressure } W_1$$

Scale (if time)

Same Data, Different units... How does Perceptron change? (It shouldn't... but it does)



$$X = \begin{bmatrix} \text{Temp K} \\ \text{Pressure} \end{bmatrix}$$

$$X \cdot W = \text{Temp K} \cdot W_0 + \text{Pressure}\ W_1$$

As feature 1 gets larger it grows increasingly important in decision,
other units (pressure) are effectively ignored if feature 1 continues to grow