

FROM LINEAR ALGEBRA TO MACHINE LEARNING

Omar Gutiérrez

July 27, 2018

@trinogz

OVERVIEW

Motivation

Vectors

Examples

Conclusions

MOTIVATION

MOTIVATION

- **Linear algebra** is important to understand machine learning.
- As well as **calculus**, **probability theory**, and **statistics**.
- It is rewarding to take the **hard path** to learn machine learning (IMHO).

LEARNING FROM ERRORS

```
30 # compute distances using self-created function
31 distances2 <- matrix(nrow=size, ncol=size)
32 for (p in 1:size) {
33   for (q in 1:size) {
34     row_p = iris2[p,]
35     row_q = iris2[q,]
36     distances2[p, q] <- euclidean_distance(row_p, row_q)
37   }
38 }
```

LEARNING FROM ERRORS

```
14 # function for calculating Euclidean Distance
15 euclidean_distance <- function(p, q) {
16     ed = 0
17     for (i in 1:4) {
18         ed <- ed + (p[,i] - q[,i]) ^ 2
19     }
20     ed <- sqrt(ed)
21     return(ed)
22 }
```

VECTORS

A VECTOR IS A COLLECTION OF NUMBERS

$$\vec{a} = \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

LENGTH OF A VECTOR

$$|a| = \sqrt{\sum_{i=1}^n a_i^2}$$

DISTANCE BETWEEN VECTORS

$$\begin{aligned}d(\mathbf{a}, \mathbf{b}) &= \|\mathbf{a} - \mathbf{b}\| \\&= \sqrt{\sum_{i=0}^n (a_i - b_i)^2}\end{aligned}$$

DOT PRODUCT

$$\begin{aligned} a \cdot b &= \sum_{i=0}^n a_i b_i \\ &= a_0 b_0 + a_1 b_1 + \dots + a_n b_n \end{aligned}$$

So,

$$\begin{aligned} a \cdot a &= a_0 a_0 + a_1 a_1 + \dots + a_n a_n \\ &= a_0^2 + a_1^2 + \dots + a_n^2 \\ &= |\mathbf{a}|^2 \end{aligned}$$

EXAMPLES

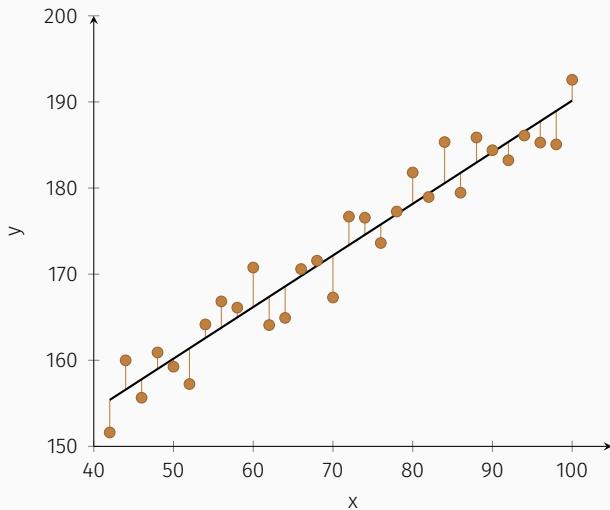
WINTER ^{IS} COMING

A white sword graphic is positioned vertically, acting as a separator between the words 'WINTER' and 'COMING'. The sword has a simple hilt with a crossguard and a long, straight blade pointing downwards. The entire graphic is set against a black background with a subtle, grainy texture.

THE AI WINTER IS COMING

- Is really coming? No.
- However, we already had an AI winter.
- The research on neural nets was stopped for many years, after Minsky and Papert proved that a single layer perceptron was not able to deal with the exclusive-or problem.

LINEAR REGRESSION



LINEAR REGRESSION

- We want to calculate the intercept a and the slope b .

$$\arg \min_{a,b} \sum_i (y_i - (ax_i + b))^2 = \arg \min_w ||Xw - y||^2$$

- The solution to this optimization problem is:

$$w^* = (X^T X)^{-1} X^T y .$$

LINEAR REGRESSION

CONCLUSIONS

REFERENCES

- Mathematics for Machine Learning: Linear Algebra offered by Coursera.
- The Math of Intelligence offered by Siraj Raval.

Thank you
Questions?