Lecture 2:

- A linear classifier:

$h(x; \theta, \theta_0) = \text{sign}(\theta^T x + \theta_0)$

When we run our random learning algorithm, we can see that while our training errors don't get ~~wore~~ worse, the rate at which it improves is very slow and inefficient.
[Refer 8:13 to 18:08 lecture-2 Youtube]

* Perceptron Algorithm

```
Perceptron (Dn ; τ)
  Initialize θ = [0 0 ... 0]^T   [It has to have 'd' 0's]
  Initialize θ0 = 0
  for t = 1 to τ        // Represents an upper bound for time provided
    changed = False
    for i = 1 to n   // n = size of data
(*i)  if y^(i) (θ^T x^(i) + θ0) ≤ 0
        set θ = θ + y^(i) x^(i)
        set θ0 = θ0 + y^(i)
        changed = True
    if not changed:
      break
  Return θ, θ0
```

(*i) True if either:
A. Pt. not on line and prediction is wrong.
B. Point is on line
C. Initial step.

What does an update do?

$y^{(i)}(\theta^T_{updated} x^{(i)} + \theta_{0,updated})$

$y^{(i)}((\theta + y^{(i)} x^{(i)})^T x^{(i)} + \theta_0 + y^{(i)})$

$y^{(i)}((\theta x^{(i)} + \theta_0)) + \underbrace{y^{(i)2}(x^{(i)T} x^{(i)} + 1)}_{\text{change}}$

$y^{(i)}(\theta x^{(i)} + \theta_0) + (\|x^{(i)}\|^2 + 1)$ Might change the -ve value to +ve by adding

The algorithm stops when it reaches '0'-training error

Q. How does the classifier move when it has a misclassification

⇒ We're trying to get the angle right, so that we can correctly classify the particular data point

We hope,

$$|(||x^{(i)}||^2+1)| > |y^{(i)}(\theta^T x^{(i)}+\theta_0)|$$

so that the entire value becomes more +ve and the if statement doesn't get executed

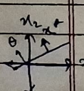
It doesn't have to strictly decrease the error.

* Classifier quality:

- Definition: A training set D_n is linearly separable if there exist θ, θ_0 such that, for every point index $i \in \{1,\dots,n\}$, we have

$$y^{(i)}(\theta^T x^{(i)}+\theta_0) > 0$$

- The signed distance from a hyperplane defined by θ, θ_0 to a point x^* is



$$= \text{projection of } x^* \text{ on } \theta - \text{signed distance of line to origin}$$

$$= \frac{\theta^T x^*}{||\theta||} - \frac{(-\theta_0)}{||\theta||} = \frac{\theta^T x^* + \theta_0}{||\theta||}$$

- Definition: The margin of the labelled point (x^*, y^*) w.r.t. the hyperplane defined by θ, θ_0 is:

$$y^*\left(\frac{\theta^T x^* + \theta}{||\theta||}\right)$$

- Definition: The margin of the training set D_n w.r.t. the hyperplane defined by θ, θ_0 is:

$$\min_{i\in\{1,..,n\}} y^{(i)}\left(\frac{\theta^T x^{(i)} + \theta_0}{||\theta||}\right)$$

* If we get even one point wrong, the margin of ~~tra~~ D_n will be −ve, but if we get each one right margin will be

- Theorem: Perceptron Performance.

- Assumptions:

A. Our hypothesis class = classifiers with separating ~~hyplanes~~ hyperplanes that pass through the origin ($\theta_0=0$)

B. There exists θ^* and γ such that $\gamma>0$ and, for every $i\in\{1,..,n\}$ we have $y^{(i)}\left(\frac{\theta^{*T} x^{(i)}}{||\theta||}\right) > \gamma$

C. There exists R such that, for every $i \in \{1,\dots,n\}$, we have $||x^{(i)}|| \le R$

- Conclusion: Then the perceptron algo. will make at most $(R/\gamma)^2$ updates to θ. Once it goes through a pass of i w/out changes, the training error of its hypothesis will be zero

*. Why classifiers through the origin?

- If we're clever, we don't lose any flexibility

  - Classifier with offset.

    $x \in R^d$, $\theta \in R^d$, $\theta_0 \in R$

    $x : \theta^T x + \theta_0 = 0$

  - Classifier w/out offset.

    $x_{new} \in R^{d+1}$, $\theta_{new} \in R^{d+1}$

    $x_{new} = [x_1, x_2, \ldots x_d, 1]$, $\theta_{new} = [\theta_1, \theta_2, \ldots \theta_d, \theta_0]$

    $x_{new}, 1 \in d : \theta_{new}^T x_{new} = 0$

Problem: Typical Real data sets aren't linearly separable.

- Classification: Mapping to a discrete set.

- Regression: Mapping to continuous values

- Supervised Learning: Learn a mapping from features to labels

- Unsupervised Learning: No Labels; find patterns