Perceptron Learning Algorithm

Sou. Sarathi
IPHD20

① Perceptron: Algorithm used for binary classification of data.

ip: is a vector $x \in \mathbb{R}^d$ op: is a label $y \in \{+1, -1\}$

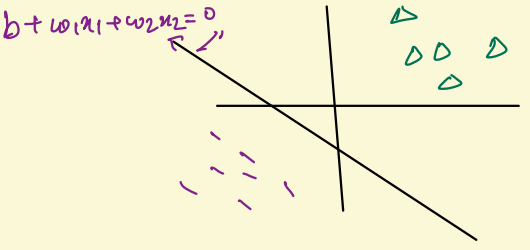
$$op = \text{sgn}(w^T x + b) = \text{sgn}(b + \sum w_i x_i)$$

b = bias term
w_i = weights

ip x & if $\text{sgn}(w^T x + b) \geq 0 \rightarrow$ predict $y = 1$
$\text{sgn}(w^T x + b) < 0 \rightarrow$ predict $y = -1$

try to find a plane $w^T x + b$ s.t

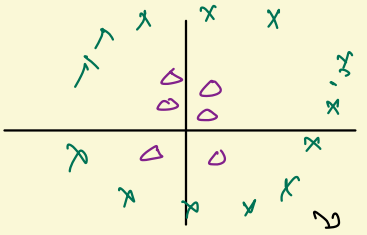
- gives a linear classifier that represents a hyperplane which seperates the space into 2 half-spaces.



Goal: to find a hyperplane

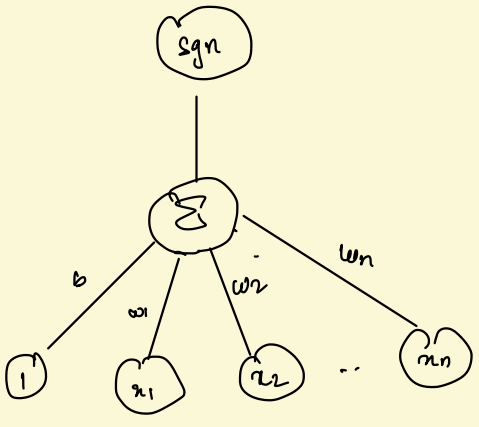
* Parameters: $w = \begin{pmatrix} b \\ w_1 \\ w_2 \end{pmatrix}$, η = learning rate → higher the learning rate faster is the convergence to the hyperplane.
coefficient $x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$

$$w^T x = (b, w_1, w_2) \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = w^T x.$$



This dataset seperable but not with perceptron.

Checkout → Youtube: Ritvik. watch



We give training data and try to create an hyperplane using it through the algorithm so that for the new data, x_i, it can classify it using the hyperplane.

The Perceptron Algorithm

- Initialize $w_0 = 0 \in \mathbb{R}^n$
- For each training example (x_i, y_i)

$y' = \text{sgn}(w_t^T x_i)$

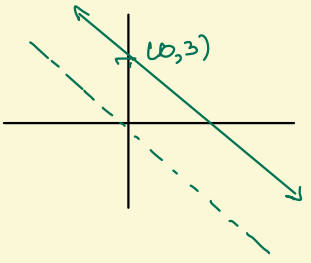
If $y_i \neq y'$

Update $w_{t+1} \leftarrow w_t + r(y_i x_i)$

- Return final Vector

- **Some Prerequisites:**

★ Hyperplane : The eqn is $wx + b$, where w is the normal to the hyperplane and b is an offset.



Example : (i) $x + y = 3$: $y = -x + 3$

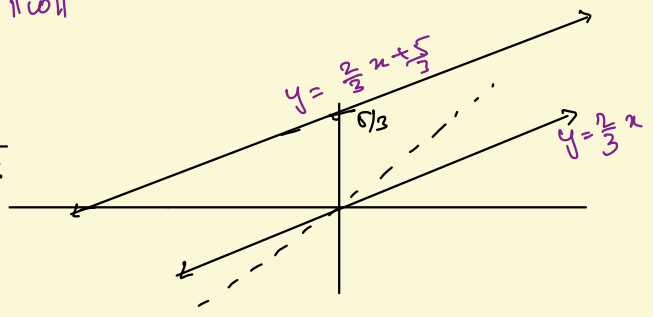
If we multiply $wx+b$ by $\frac{1}{\|w\|}$, then we have,

$\hat{w} = \frac{w}{\|w\|}$, $\hat{w}x + b' = 0$.

$\hat{w}$ = unit normal to the hyperplane.

$b' = \frac{b}{\|w\|}$ = distance of hyperplane from origin

(ii) $-2x + 3y = 5$: $y = \frac{2}{3}x + \frac{5}{3}$



given a hyperplane, $wx + b$

in 2D, $w_1x_1 + w_2x_2 + w_3x_3 + 1 \cdot b$

Distance of point y from $wx + b$ is: $\frac{wy + b}{\|w\|}$ i.e

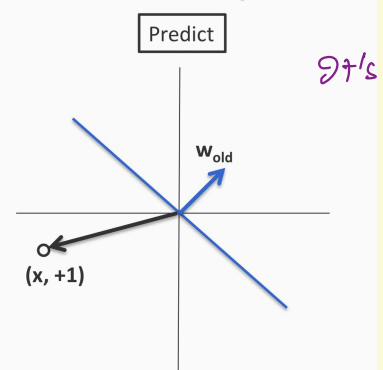
$$\frac{w_1y_1 + w_2y_2 + w_3y_3 + b}{\sqrt{w_1^2 + w_2^2 + w_3^2}}$$

in (ii) dis. $[0,0] = \frac{-2 \cdot (0) + 3(0) + 5}{\sqrt{2^2 + 3^2}} = \frac{5}{\sqrt{13}}$

↳ Idea Behind Perceptron Update:

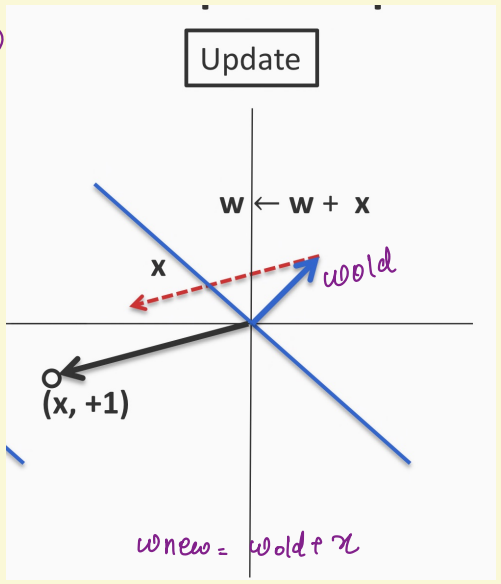
Look @ utah slide 19.

(1)



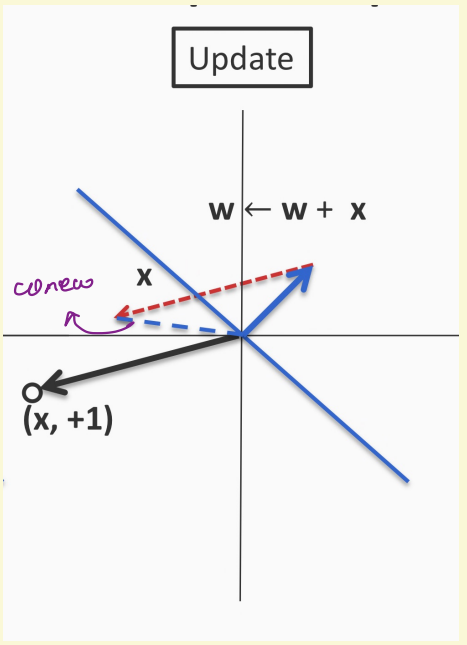
y+'s +1 but predicted −1

(2)

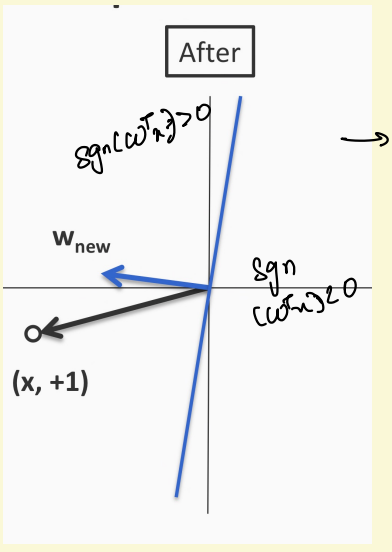


∴ Find a hyperplane w so that wnew is a normal to that hyperplane

(3)



(4)



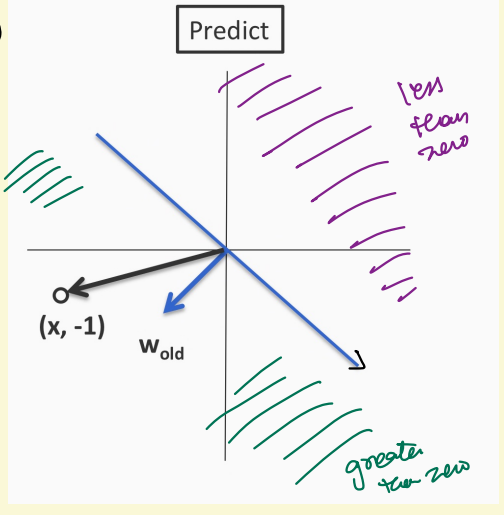
→ After the update. we get hyperplane whose normal is wnew

Imp

In this eg, since the label of x is +1, we want it to be @ the right of hyperplane But in the beginning it was on the left of the hyperplane (wold) Aim: is to find a hyperplane so that x lies right to the hyperplane. After update we found a new hyperplane, s.t the point x is to the right of the hyperplane.

Another case Mistake on a negative eg

(1)

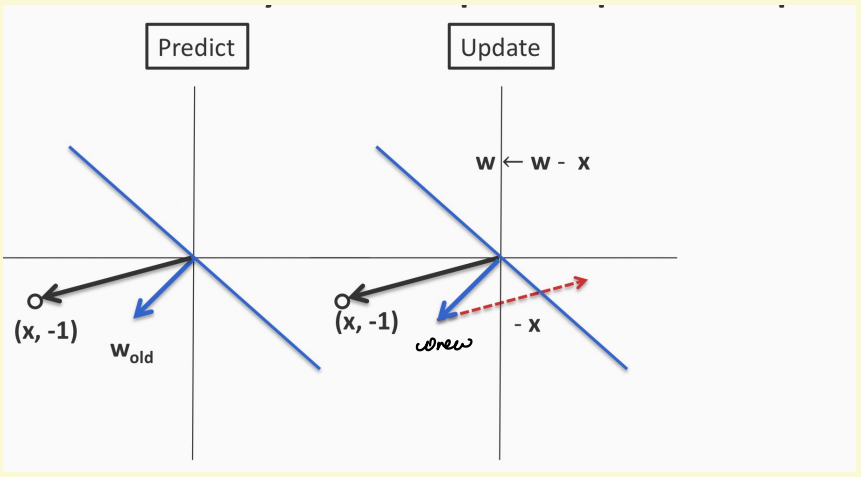


For this hyper plane, the normal is wold. ∴ To the left of hyperplane sgn(wTx) > 0 & right sign(wTx) < 0

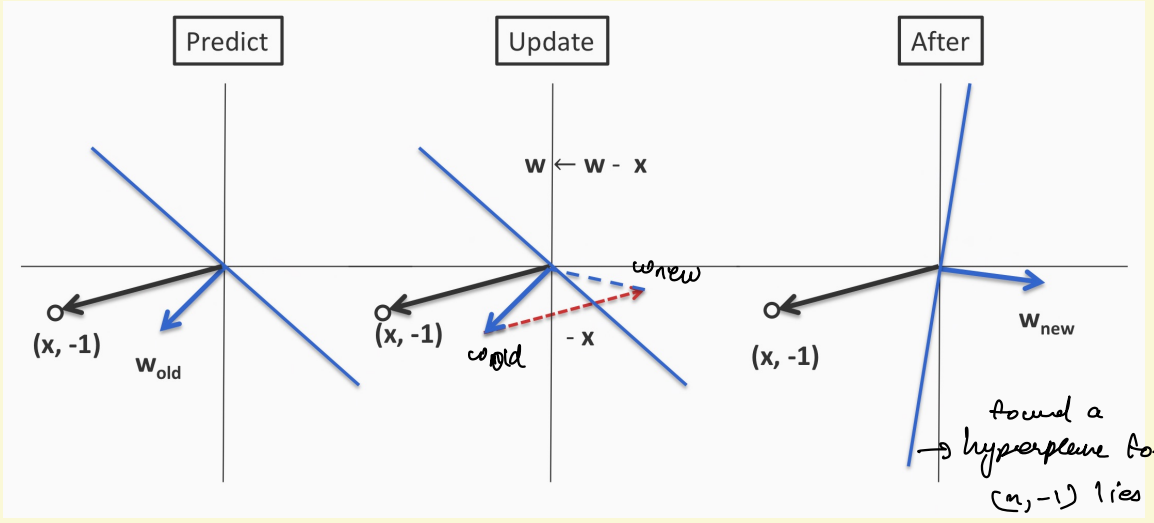
Given: (x, -1) But this lies in the wrong side of the hyperplane.

Aim: To rotate the hyperplane so that (x, -1) lies on the left of the hyper. ∵ the label is -1

(2)



(3)



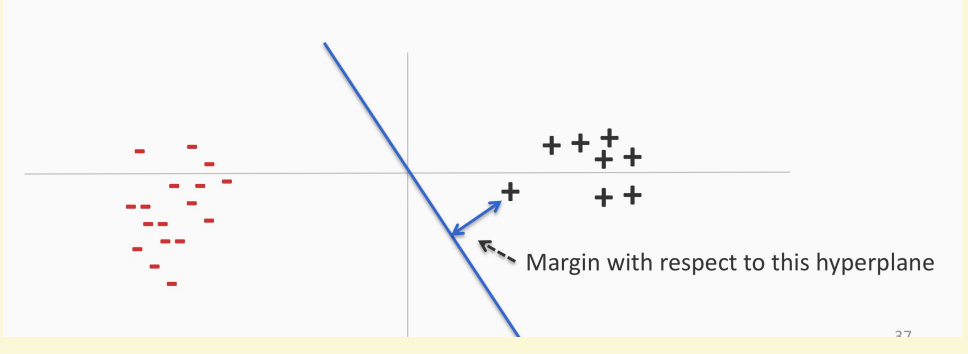
found a hyperplane for which (x, -1) lies to the left of it

Convergence:

(1) If the data is linearly seperable, perceptron algorithm will converge.

(2) Cycling thm:
If the training data isn't seperable, then learning algo. will eventually repeat the same set of weights & enter an ∞ loop.

(3) Margin thm:
Margin of a hyperplane is the dist. b/w the hyperplane & data point nearest to it.
Margin of a dataset (γ): is the max. margin for that dataset using any weight vector.

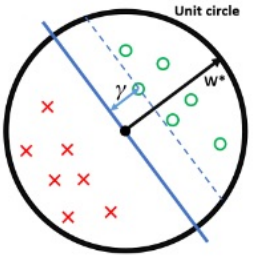


Convergence : cornell.edu - lect3

Stmt : If a data set is linearly seperable, then perceptron will find a seperating hyperplane. (If a data set is linearly seperable, it'll loop forever.

Argument : Suppose $\exists\ w^*$ s.t $y_i (w^{*T} x_i) > 0 \quad \forall (x_i, y_i) \in D$

s.t $\|w^*\| = 1$ and $\|x_i\| \leq 1. \quad \forall x_i \in D.$ margin : $\gamma = \min_{(x_i, y_i) \in D} |x_i^T w^*|$



To summarize our setup:
- All inputs $\mathbf{x}_i$ live within the unit sphere
- There exists a separating hyperplane defined by $\mathbf{w}^*$, with $\|\mathbf{w}\|^* = 1$ (i.e. $\mathbf{w}^*$ lies exactly on the unit sphere).
- γ is the distance from this hyperplane (blue) to the closest data point.

// These assumptions essentially mean that data is linearly seperable.

∴ If all the above holds, then the perceptron will make atmost $(1/\gamma^2)$ mistakes.

Suppose we have a binary classification dataset with n dimensional inputs.

If the data is separable,... *...then the Perceptron algorithm will find a separating hyperplane after making a finite number of mistakes*

Proof

After t mistakes, we make 2 claims.

Claim 1:	$w^{*T} w_t \geq t\gamma$
Claim 2:	$\|w_t\|^2 \leq t$

Claim 1 Proof:

$$w^{*T} w_{t+1} = w^{*T} (w_t + y_i x_i) = w^{*T} w_t + y_i (w^{*T} x_i) \geq w^{*T} w_t + \gamma$$

($y_i (w^{*T} x_i) \geq 0$)

$$\therefore w^{*T} w_{t+1} \geq w^{*T} w_t + \gamma$$

$w_0 = 0 \Rightarrow w^{*T} w_1 \geq \gamma, \quad w^{*T} w_2 \geq 2\gamma, \quad \dots \quad w^{*T} w_t \geq t\gamma$

Hence $w^{*T} w_t \geq t\gamma$

Claim 2 Proof:

$$\|w_{t+1}\|^2 = \|w_t + y_i x_i\|^2 \quad \text{(In case of update)}$$

$$= \|w_t\|^2 + 2y_i (w_t^T x_i) + \|x_i\|^2$$

Observe $2y_i(w_t^T x_i) < 0$ ($\because$ it's the reason update was made!)

and $\|x_i\|^2 \leq 1$

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1$$

$w_0 = 0 \Rightarrow \|w_1\|^2 \leq 1 \quad \|w_2\|^2 \leq 2 \quad \dots \quad \|w_t\|^2 \leq t$

$$\therefore \quad t\gamma \leq w_t^T w^* = \|w_t^T\| \|w^*\| \cos(\theta)$$

(Definition of Ip)
θ = angle b/w w^* & w_t^T)

$$\leq \|w_t^T\|$$

$$\leq \sqrt{t}$$

$$\Rightarrow \quad t\gamma \leq \sqrt{t} \quad \Rightarrow \quad \sqrt{t} \leq \frac{1}{\gamma}$$

$$\Rightarrow \quad t \leq \frac{1}{\gamma^2}$$

$\therefore$ # of mistakes t is bounded above by a constant.

The Algorithm:

Perceptron Algorithm

```
1  Initialize w⃗ = 0⃗                          // Initialize w⃗. w⃗ = 0⃗ misclassifies everything.
2  while TRUE do                              // Keep looping
3      m = 0                                  // Count the number of misclassifications, m
4      for (x_i, y_i) ∈ D do                  // Loop over each (data, label) pair in the dataset, D
5          if y_i(w⃗^T · x⃗_i) ≤ 0 then          // If the pair (x⃗_i, y_i) is misclassified
6              w⃗ ← w⃗ + y x⃗                   // Update the weight vector w⃗
7              m ← m + 1                      // Counter the number of misclassification
8          end if
9      end for
10     if m = 0 then                          // If the most recent w⃗ gave 0 misclassifications
11         break                              // Break out of the while-loop
12     end if
13 end while                                  // Otherwise, keep looping!
```

Line 5 checks if predicted sign & actual sign are equal.

line 10 check if there is no update in that cycle, it means the hyperplane has been found.

Margin Perceptron

6. Given a training set D with a separation margin γ_0, the original perceptron algorithm predicts a mistake when $y\mathbf{w}^{(n)^T}\mathbf{x} < 0$. As we have discussed that the perceptron algorithm converges to a linear classifier that can perfectly separate D but does not necessarily achieve the maximum margin. The **margin perceptron algorithm** extends presented perceptron Algorithm to approximately maximize the margin in the perceptron algorithm, where it is considered to be a mistake when $\frac{y\mathbf{w}^{(n)^T}\mathbf{x}}{\|\mathbf{w}^{(n)}\|} < \frac{\gamma}{2}$, where $\gamma > 0$ is a parameter. Prove that the number mistakes made by the margin perceptron algorithm is at most $\frac{8}{\gamma_0^2}$ if $\gamma \leq \gamma_0$.

We normalize all the examples (x_i, y_i) so that $\|x_i\| \le 1 \quad \forall i$

Why Margin Perceptron Algorithm? We iterate through the data using perceptron algorithm updating the weights not only on mistakes but also on examples x that the current hypothesis gets correct by a margin $< \frac{\gamma}{2}$ [so if an ex is with margin $\frac{\gamma}{2}$ then it's incorrect]

predict +ive $\frac{w_t^T x_i}{\|w_t\|} \ge \frac{\gamma}{2}$, negative if $\frac{w_t^T x_i}{\|w_t\|} < -\frac{\gamma}{2}$

$\frac{y_i w^{(m)T} x_i}{\|w^m\|} < \frac{\gamma}{2}$ then it's a mistake

$$\|w_{t+1}\|^2 \le \|w_t\|^2 + \|x_i\|^2 + 2 y_i w_t^T x_i$$

$$\|w_{t+1}\|^2 \le \|w_t\|^2 + \|x_i\|^2 + 2 y_i w^T x_i$$

$$\le \|w_t\|^2 + 2 w^T x_i + \|x_i\|^2$$

$$\le \|w_t\|^2 \left(1 + \frac{1}{\|w_t\|} \frac{w_t x_i}{\|w_t\|} + \frac{\|x_i\|^2}{\|w_t\|^2} \right)$$

$$\|w_{t+1}\| \le \|w_t\| \left[1 + \underbrace{\frac{2}{\|w_t\|} \frac{\gamma}{2}}_{\ge 0} + \frac{1}{\|w_t\|^2} \right]^{1/2}$$

(for $x > 0 \quad \sqrt{1+x} < 1 + \frac{x}{2}$) ↳ Verify by MVT

$$\|w_{t+1}\| \le \|w_t\| \left(1 + \frac{\gamma}{2\|w_t\|} + \frac{1}{2\|w_t\|^2} \right)$$

Hence $$\|w_{t+1}\| \le \|w_t\| + \frac{1}{2\|w_t\|} + \frac{\gamma}{2}$$

If the classification is $\frac{y_i w_t^T x_i}{\|w_t\|} < \frac{\gamma}{2}$ it's a mistake

$$\frac{\|w_t\|}{y_i w_t^T x_i} > \frac{2}{\gamma}$$

If for some eg we take $y_i w_t^T x_i = 1$ then, $\|w_t\| > \frac{2}{\gamma}$

$\left(\frac{1}{\|w_t\|} < \frac{\gamma}{2}\right)$

$$\|w_{t+1}\| \leq \|w_t\| + \frac{\gamma}{4} + \frac{\gamma}{2} = \|w_t\| + \frac{3\gamma}{4}$$

Repeating it recursively after m mistakes

$$\|w_{m+1}\| \leq \|w_t\| + \frac{3M\gamma}{4} < \frac{2}{\gamma} + \frac{3M\gamma}{4}$$

Also for a perceptron algo recall $M\gamma \leq \|w_{M+1}\|$

So, $M\gamma \leq \|w_{M+1}\| < \frac{2}{\gamma} + \frac{3M\gamma}{4}$

$$M\gamma \leq \frac{2}{\gamma} + \frac{3M\gamma}{4} \Rightarrow \frac{M\gamma}{4} \leq \frac{2}{\gamma} \Rightarrow \boxed{M \leq \frac{8}{\gamma^2}}$$