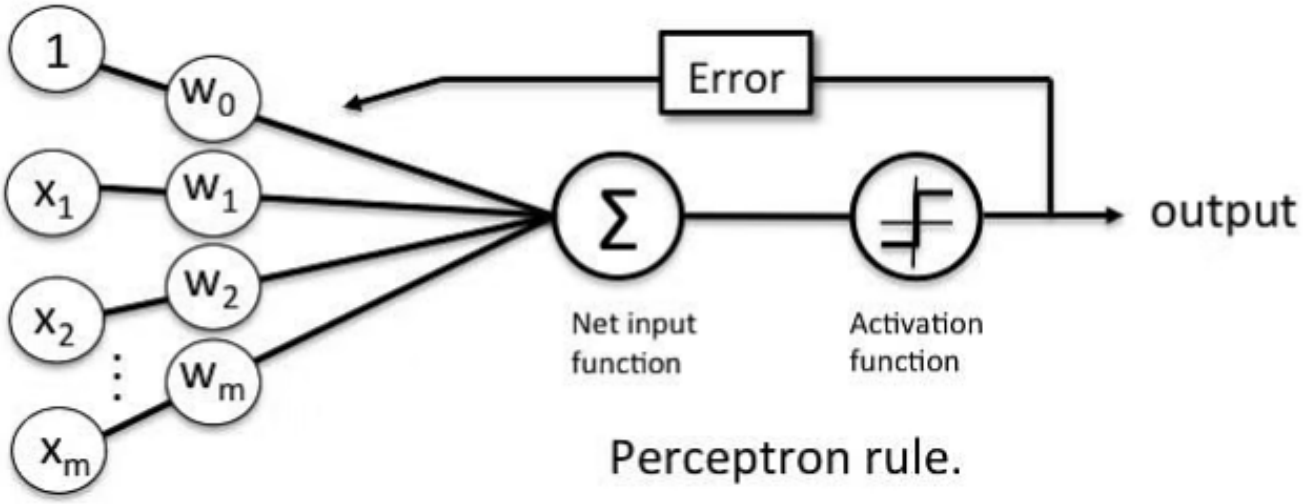


Understanding Perceptron Algorithm - Handwritten Notes

Table of Contents

→ The neural networks are further developed in various architectures, and now these

> Artificial neural networks (ANNs) are the very core of Deep Learning.

They are very powerful, making them ideal to tackle large and highly complex machine learning tasks such as classifying billions of images (eg - Google Images), recommending the best videos to watch, etc..

Simplest Artificial neural network Architecture

The Perceptron

→ The Perceptron is one of the simplest ANN Architecture, it is based on a artificial neuron called — "Threshold Logic Unit (TLU)".

→ The inputs and outputs are now numbers (instead of binary on/off values) and each input connection is associated with a weight.

→ Before seeing this architecture, it would be much clear if we can understand the Perceptron Arch.

perceptron Algorithm

Input:- $\{(x_1, y_1), \ldots (x_n, y_n)\}$ $x_i \in \mathbb{R}^d$

Iteration:- $w^0 = 0 \in \mathbb{R}^d$ $[0, \ldots, 0]$

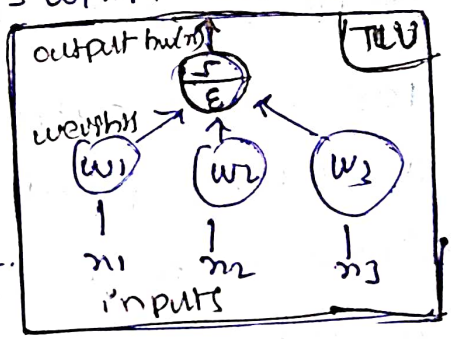
until convergence:

- pick (x_i, y_i) pair from the dataset
- If $\text{sign}(w_t^T x_i) = y_i$ (do nothing)
- Else, $w_{t+1} = w_t + x_i y_i$ update Rule.
  - $x_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$
- End.

Assumptions:- the dataset is linearly seperable.

→ The TLU computes a weighted sum of its inputs, then applies a step function to that sum and outputs the result.

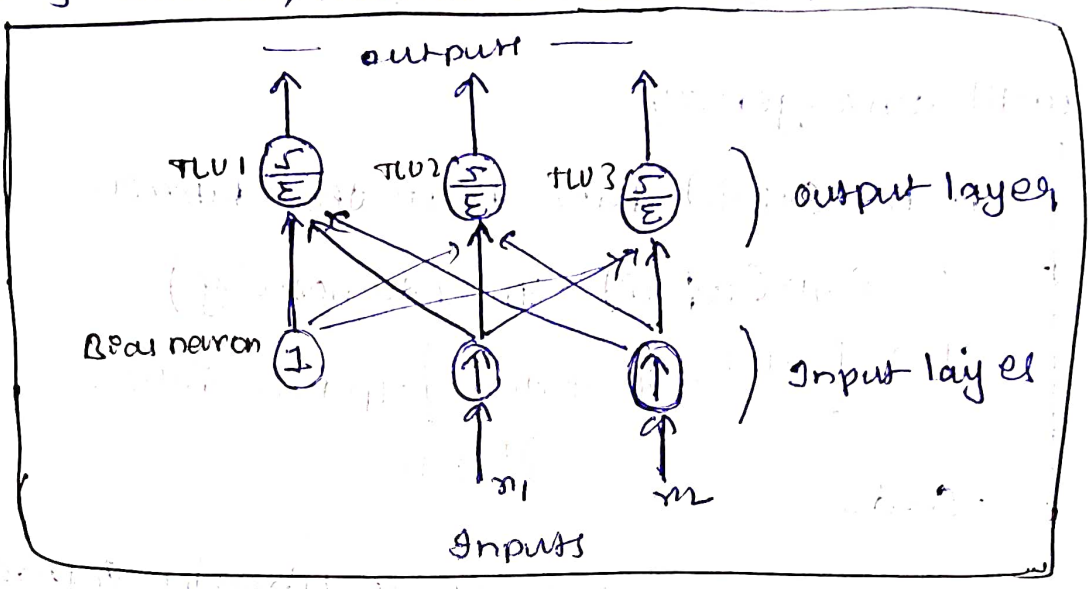
$h_w(x) = \text{step}(z)$, where $z = x^T w = w_1 x_1 + \cdots + w_n x_n$



→ Training a TLU in this case means that finding the right values for w_0, w_1 & w_2.

→ Here, a perceptron is simply composed of a single layer TLUs with each TLU connected to all the inputs. An extra bias feature is generally added ($x_0 = 1$), which just outputs 1 all the time, called a bias neuron.

→ Here is a perceptron with two inputs and 3 outputs, this perceptron can classify instances simultaneously into 3 different binary classes, making it multioutput classifier.



→ So how this perceptron is trained?
Just like update rule, we have seen before,

→ Perceptron is fed one training instance at a time, and for each instance it makes its predictions.
For every output neuron that produced a wrong prediction, it reinforces the connection weights from the inputs that would have contributed to correct prediction.

→ And if the training instances are linearly seperable, it is said that perceptron would converge to a solution.

HOW?

Update Rule

$w^{t+1} = w^t + x_i y_i$ → if mistake $sign(x_i^T w) \neq sign(y_i)$

Now, we need to understood how by convergence we will get a bet for w^t!!

→ If we think about it, there are two different ways this update rule can go wrong.

mistake ①

Predicted = 1 $sign(w^{t^T} x_i) \geq 0$

Actual = -1 $y_i = -1$

mistake ②

predicted = -1 $sign(w^{t^T} x_i) < 0$

Actual = +1 $y_i = +1$

→ Say, we are using the update rule as $w^{t+1} = w^t + x_i y_i$

→ So, if we take update rule as a good one, then it should not make mistake for the input for which previous weight had made a mistake.

→ Let's find the new weight dot product with the point where old weight made a mistake.

→ $(w^{t+1})^T x_i = (w^t + x_i y_i)^T x_i = w^{tT} x_i + y_i \|x_i\|^2$

→ Now, considering type 1 mistake,

we have, $w^{tT} x_i \geq 0$

$y_i = -1$

Substitute them in new weight,

And we get,

$$(w^{t+1})^T x_i = \underbrace{w^{tT} x_i}_{\geq 0} + \underbrace{y_i}_{-1} \underbrace{\|x_i\|^2}_{\geq 0}$$

This whole thing is negative.

→ From this, what we can say is that the new weight's dot product with x_i is equal to,

old weight's dot product with x_i − Something.

→ So, we are subracting out something from old w's dot product.

→ well, the old w's was positive, and by subtracting something, we can say that the value is reducing.

→ ofcourse, it doesn't immediately become negative, but however it is moving in the right direction as we are subtracting!!

→ Similarly if we consider Type 2,

where,

$$(w^{t^T} x_i) < 0$$
$$y_i = +1$$

substitute them in new weight,

And we get,

$$(w^{t+1})^T x_i = \underbrace{w^{t^T} x_i}_{<0} + \underbrace{y_i \|x_i\|^2}_{>0}$$

→ and here we are adding something.

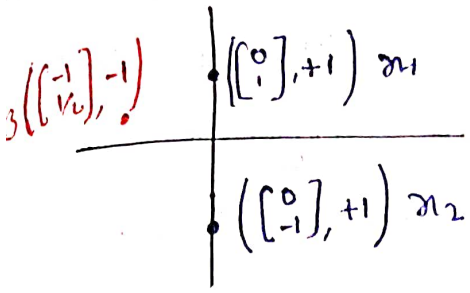
→ overall we can say that,

update Rule pushes w in the right direction for x_i.
i.e; if dotproduct is -ve, when it should have been +ve, the update rule is subracting something.

→ But then, the question raised that, okay we fixed the previous mistake, but what's the guarantee that this newweight didn't breaks anything that was previously correct!!

→ So Problem is – fixing w for one x_i, might affect decision for other data points. So we need more careful argument for convergence.

Let's solve an example..



Is this a linear seperable data or?

yes, there is $w \in \mathbb{R}^2$

s.t. $w^T x_i \geq 0 \Rightarrow y_i = +1$

$w^T x_i < 0 \Rightarrow y_i = -1$

Solving with perceptron as it's linearly seperable.

Initially we can take the weight $[0, 0]$

And that given us, $w^0 = [0, 0]$

$w^{0T} x_1 = 0$ | $w^{0T} x_2 = 0$ | $w^{0T} x_3 = 0$ → This is a mistake.

$\hat{y}_1 = +1$ | $\hat{y}_2 = +1$ | $\hat{y}_3 = +1$

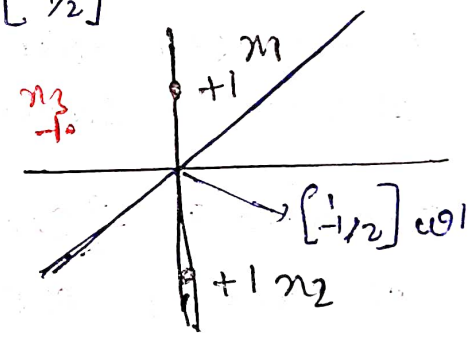
$y_3 = -1$

So we find w^{t+1}

$w^1 = w^0 + x_3 y_3 = \begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}-1\\1/2\end{bmatrix} \times -1 = \begin{bmatrix}1\\-1/2\end{bmatrix}$

→ Now with the new w_1 line, we can see that it made mistake for x_1.

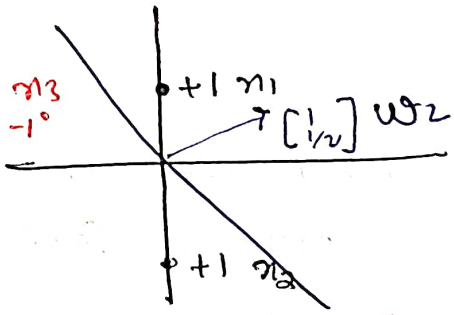
So we find w_2



$w^2 = w^1 + x_1 y_1 = \begin{bmatrix}1\\-1/2\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} \cdot 1 = \begin{bmatrix}1\\1/2\end{bmatrix}$

→ Now with the new w_2 line, we can see that it made mistake for x_2.

So we find w^3.



$w^3 = w^2 + x_2 y_2 = \begin{bmatrix}1\\-1/2\end{bmatrix} = w^1$

And now w^3 has become w_1, and then w^4 will become w_2 & it will keep looping like this & it will not have a convergence!!

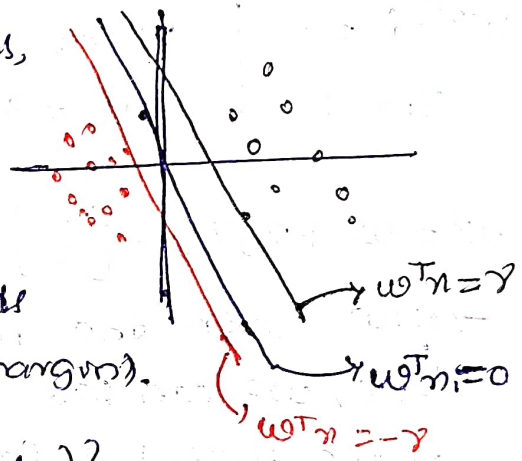
→ Now if we wonder is perceptron wrong or data wrong. And if we closely observe the data, the points actually lie on the decision boundary, we are saying $w^T x_i \geq 0$ → it can also be equal to "0".

i.e, $w^T x_i$ can be "0". so we need to make sure it can't be zero.

So the data Assumption has to be changed.

① Linear seperability with "γ" margin.

→ If we take the data like this, where there is a region with no datapoints. Then data can be modelled well with perceptron. It needs atleast a small region (γ-margin).



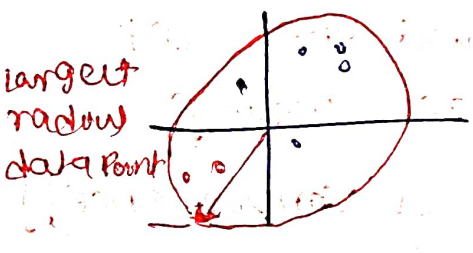
→ A dataset $\{(x_1, y_1), \ldots (x_n, y_n)\}$ is linearly seperable with γ-margin.

→ If $\exists w^* \in \mathbb{R}^d$ s.t, $(w^{*T} x_i) y_i \geq \gamma \ \forall i$ for some $\gamma > 0$.

→ For ease or simplification or Proof, we can consider few harmless assumptions such as

② Radius Assumption

$\forall i$ (every point) $\in D$, $\|x_i\|_2 \leq R$ for some $R > 0$



→ For a largest point, we are away, all other points we could that circle.

3) without loss of generality, assume $\|w^*\| = 1$,

$w^* \rightarrow$ The actual best line that seperates the data.

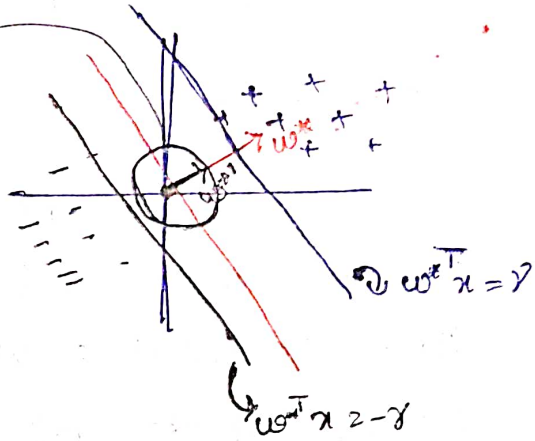
we know that there will be w^*, but we can't be sure will norm ($\|w^*\|$) will be equal to 1.

However, we can say that there will exist a w^{*1}, which will have norm = 1.

we have,

$w^{*T} x = \gamma$

considering $\|w^*\| = 1$



divide by $\|w^*\|$

$$\frac{w^{*T} x}{\|w^*\|} = \frac{\gamma}{\|w^*\|}$$

$$\left(\frac{w^{*T}}{\|w^*\|}\right) \cdot x = \gamma'$$

$$\therefore \gamma' = \frac{\gamma}{\|w^*\|}$$

↳ considered w^{*1}

$$(w^{*1})^T x = \gamma'$$

↳ Here we just scaled the w^*.

But the advantage is that, the new w^{*1} we have now has norm = 1.

i.e, If you give me a dataset, with w^* & some γ, say $\gamma = 10$.

Now I can create another w^* by rescaling the w^*, to make sure it has length 1, and the γ will rescale accordingly. Then I will have a w^{*1} which also linearly seperates data with a different γ', so this assumption holds ✓

⊙ Proof of Perceptron ←→

→ the way we are gonna prove that this algorithm converges is by following idea,

> we say that perceptron algorithm cannot make more than a certain mistakes.

↳ Idea of convergence.

→ If we think about it, it makes sense that if the mistakes are gonna be infinite, then it will never converge. So, if we can prove that the mistakes are gonna be finite, then we can say it converges.

Analysis of mistakes of Perceptron

→ we know that, an update in the perceptron happens only when a mistake happens.

→ let's say w^t is the current guess, and then a mistake happens wrt (x, y).

Then, $w^{t+1} = w^t + x \cdot y$

let's understand the [growth or length] of this update.

$$\|w^{t+1}\|^2 = \|w^t + xy\|^2 = (w^t + xy)^T (w^t + xy)$$

$$= \|w^t\|^2 + \underbrace{2 \cdot (w^{tT} x) y}_{\leq 0 \text{ (because mistake)}} + \underbrace{\|x\|^2 \cdot y^2}_{\leq R^2 \;\leftarrow\; +1 \;\rightarrow \text{ assumed} \in \mathbb{R}}$$

→ If we observe, why we said 2nd term $(2 \cdot (w^{(t)T} x) y) \leq 0$, it is because that, $w^{(t)T} x$ is the prediction (−1/+1), and y is actual (−1/+1).

And considering update happens only on mistake, there both have to be opposite values in case, and the total value will turn out to be negative.

Hence $2(w^{(t)T} x) y \leq 0$, ——→ ignored

So we can upper bound whole thing as,

$$\|w^{(t+1)}\|^2 \leq \|w^{(t)}\|^2 + R^2$$

→ This means that the new weight length will be upper — previous length + R^2,

→ same applies for previous lengths.

$$\|w^{(t+1)}\|^2 \leq (\|w^{(t-1)}\|^2 + R^2) + R^2$$

..... If we continue, we set ((for t mistakes))

$$\|w^{(t+1)}\|^2 \leq \underbrace{\|w^{(0)}\|^2}_{=0} + t R^2$$

$$\therefore \boxed{\|w^{(t+1)}\|^2 \leq t R^2} \quad \text{——— (1)}$$

→ No. of mistakes

→ So we say that the max length it can grow is 't' times R^2, and t is the no. of updates / mistakes.

→ From the other side, to understand about (w^{l+1}) quantity. Let's use w^*.

we have,

$$w^{l+1} = w^l + n \cdot y$$

$$(w^{l+1})^T \cdot w^* = (w^l + n \cdot y)^T \cdot w^*$$

$$= w^{lT} w^* + \underbrace{(w^{*T} n) \cdot y}$$

$\geq \gamma$ (based on w^* assumption)

$$(w^{l+1})^T \cdot w^* \geq w^{lT} w^* + \gamma$$

→ we can take same argument for w^l,

$$(w^{l+1})^T \cdot w^* \geq (w^{l-1^T} w^* + \gamma) + \gamma$$

..... after l mistakes.

$$(w^{l+1})^T \cdot w^* \geq \underbrace{w^{0T} w^*} + l\gamma$$

$\downarrow$

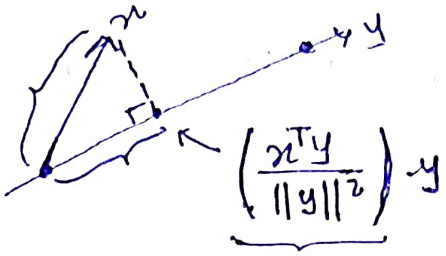
0 ($\because w^0 = 0$)

$$\boxed{(w^{l+1})^T \cdot w^* \geq l\gamma} \rightarrow (2)$$

→ It says that as we make mistakes, the dot product with w^* is increasing.

Now, if we try to make use of these two eqn. First we can recall the Cauchy-Schwartz inequality,

for any x, y



$$\left\| \left(\frac{x^T y}{\|y\|^2}\right) \cdot y \right\|^2 \leq \|x\|^2$$

Pythagorous theorem...

$$(x^T y)^2 \cdot \frac{\|y\|^2}{\|y\|^4} \leq \|x\|^2$$

$$\Rightarrow \boxed{(x^T y)^2 \leq \|x\|^2 \cdot \|y\|^2}$$

From eqn (2),

$$l\gamma \leq (w^{l+1})^T \cdot w^* \quad \text{, Apply square.}$$

$$l^2\gamma^2 \leq \left[(w^{l+1})^T \cdot w^*\right]^2 \leq \|w^{l+1}\|^2 \cdot \underbrace{\|w^*\|^2}_{1}$$ From CS

$$l^2\gamma^2 \leq \|w^{l+1}\|^2$$

$$\therefore \boxed{\|w^{l+1}\|^2 \geq l^2\gamma^2} \rightarrow \text{From (3)}$$

$$l^2\gamma^2 \leq \|w^{l+1}\|^2 \leq lR^2$$

$\uparrow$ From (3) $\quad\quad$ $\uparrow$ From (1)

$$l^2\gamma^2 \leq lR^2$$

$$\boxed{l \leq R^2/\gamma^2}$$ → Radius Margin Bound.

→ l — no. of mistakes.

→ So, by observing $\boxed{k \leq R^2/\gamma^2}$,

This tells us that k has to be utmost or R^2/γ^2,

and at the first assumption, we said $\gamma > 0$.

And so we clearly say 'k' is gonna be finite.

✱ And k is the number of mistakes, so we can finally say that the no. of mistakes is a finite quantity ξ

Hence the "Perceptron will be converged."

→ If we just take linearly seperable assumption without considering γ, i.e, $\gamma = 0$.

Then $k \leq R^2/0 \leq \infty$ (Infinity).

In this case it is not certain ξ. hence it can't be converged.

Perceptron

no. of mistakes $(k) \leq \frac{R^2}{\gamma^2}$

- under assumptions,

→ $\|x_i\|^2 \leq R^2$

→ Dataset — linearly seperated with margin γ.

$$(w^{*T} x_i) y_i \geq \gamma \quad \forall \; i, \; \gamma > 0$$