CS344: Introduction to Artificial Intelligence (associated lab: CS386)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

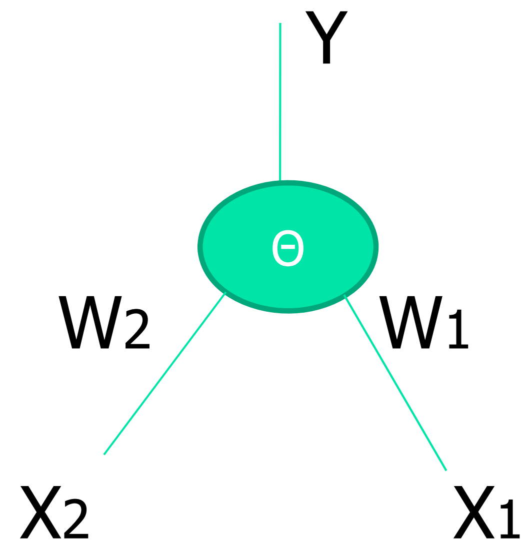
Lecture 30: Perceptron training convergence; Feedforward N/W

24th March, 2011

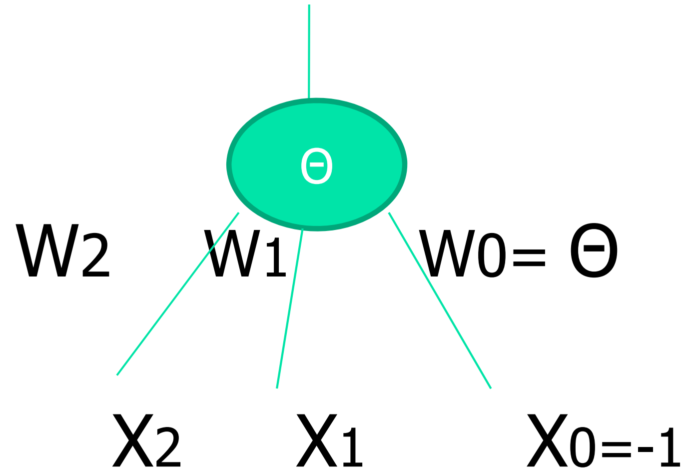
PTA on NAND

NAND:

X_2	X_1	Y
0	0	1
0	1	1
1	0	1
1	1	1



Converted To



Preprocessing

NAND Augmented:

X_2	X_1	X_0	Y
0	0	-1	1
0	1	-1	1
1	0	-1	1
1	1	-1	0

NAND-0 class Negated

	X_2	X_1	X_0
V_0:	0	0	-1
V_1:	0	1	-1
V_2:	1	0	-1
V_3:	-1	1	-1

Vectors for which W=<W2 W1 W0> has to be found such that W. Vi > 0

PTA Algo steps

Algorithm:

1. Initialize and Keep adding the failed vectors until W. Vi > 0 is true.

Step 0: $W = <0, 0, 0>$

$W_1 = <0, 0, 0> + <0, 0, -1>$ {V_0 Fails}

$= <0, 0, -1>$

$W_2 = <0, 0, -1> + <-1, -1, 1>$ {V_3 Fails}

$= <-1, -1, 0>$

$W_3 = <-1, -1, 0> + <0, 0, -1>$ {V_0 Fails}

$= <-1, -1, -1>$

$W_4 = <-1, -1, -1> + <0, 1, -1>$ {V_1 Fails}

$= <-1, 0, -2>$

Trying convergence

$W_5 = \langle -1, 0, -2 \rangle + \langle -1, -1, -1 \rangle$ {V_3 Fails}

$= \langle -2, -1, -1 \rangle$

$W_6 = \langle -2, -1, -1 \rangle + \langle 0, 1, -1 \rangle$ {V_1 Fails}

$= \langle -2, 0, -2 \rangle$

$W_7 = \langle -2, 0, -2 \rangle + \langle 1, 0, -1 \rangle$ {V_0 Fails}

$= \langle -1, 0, -3 \rangle$

$W_8 = \langle -1, 0, -3 \rangle + \langle -1, -1, -1 \rangle$ {V_3 Fails}

$= \langle -2, -1, -2 \rangle$

$W_9 = \langle -2, -1, -2 \rangle + \langle 1, 0, -1 \rangle$ {V_2 Fails}

$= \langle -1, -1, -3 \rangle$

Trying convergence

W_{10} = <-1, -1, -3> + <-1, -1, -1> {V_3 Fails}

= <-2, -2, -2>

W_{11} = <-2, -2, -2> + <0, 1, -1> {V_1 Fails}

= <-2, -1, -3>

W_{12} = <-2, -1, -3> + <-1, -1, -1> {V_3 Fails}

= <-3, -2, -2>

W_{13} = <-3, -2, -2> + <0, 1, -1> {V_1 Fails}

= <-3, -1, -3>

W_{14} = <-3, -1, -3> + <0, 1, -1> {V_2 Fails}

= <-2, -1, -4>

Converged!

W_{15} = <-2, -1, -4> + <-1, -1, -1> {V_3 Fails}
= <-3, -2, -3>
W_{16} = <-3, -2, -3> + <1, 0, -1> {V_2 Fails}
= <-2, -2, -4>
W_{17} = <-2, -2, -4> + <-1, -1, -1> {V_3 Fails}
= <-3, -3, -3>
W_{18} = <-3, -3, -3> + <0, 1, -1> {V_1 Fails}
= <-3, -2, -4>

$W_2 = -3$, $W_1 = -2$, $W_0 = \Theta = -4$

PTA convergence

Statement of Convergence of PTA

- Statement:

  Whatever be the initial choice of weights and whatever be the vector chosen for testing, PTA converges if the vectors are from a linearly separable function.

Proof of Convergence of PTA

- Suppose w_n is the weight vector at the n^{th} step of the algorithm.
- At the beginning, the weight vector is w_0
- Go from w_i to w_{i+1} when a vector X_j fails the test $w_i X_j > 0$ and update w_i as

  $$w_{i+1} = w_i + X_j$$

- Since Xjs form a linearly separable function,

  $$\exists\ w^* \text{ s.t. } w^* X_j > 0\ \forall j$$

Proof of Convergence of PTA (cntd.)

- Consider the expression

$$G(w_n) = \frac{w_n \,.\, w^*}{|\, w_n|}$$

where w_n = weight at nth iteration

- $G(w_n) = \dfrac{|w_n| \,.\, |w^*| \,.\, \cos\theta}{|w_n|}$

where θ = angle between w_n and w^*

- $G(w_n) = |w^*| \,.\, \cos\theta$
- $G(w_n) \le |w^*|$ (as $-1 \le \cos\theta \le 1$)

Behavior of Numerator of G

$w_n \,.\, w^* = (w_{n-1} + X^{n-1}_{fail})\,.\, w^*$

= $w_{n-1} \,.\, w^* + X^{n-1}_{fail} \,.\, w^*$

= $(w_{n-2} + X^{n-2}_{fail})\,.\, w^* + X^{n-1}_{fail} \,.\, w^*$ .....

= $w_0 \,.\, w^* + (X^0_{fail} + X^1_{fail} + ... + X^{n-1}_{fail}).\, w^*$

$w^*.X^i_{fail}$ is always positive: note carefully

- Suppose $|X_j| \geq \delta$, where δ is the minimum magnitude.
- Num of G $\geq |w_0 \,.\, w^*| + n\,\delta \,.\, |w^*|$
- So, numerator of G grows with n.

Behavior of Denominator of G

- $|w_n| = \sqrt{w_n \,.\, w_n}$

$= \sqrt{(w_{n-1} + X^{n-1}_{fail})^2}$

$= \sqrt{(w_{n-1})^2 + 2.\, w_{n-1}.\, X^{n-1}_{fail} + (X^{n-1}_{fail})^2}$

$\leq \sqrt{(w_{n-1})^2 + (X^{n-1}_{fail})^2}$ (as $w_{n-1}.\, X^{n-1}_{fail} \leq 0$)

$\leq \sqrt{(w_0)^2 + (X^0_{fail})^2 + (X^1_{fail})^2 + \ldots + (X^{n-1}_{fail})^2}$

- $|X_j| \leq \rho$ (max magnitude)
- So, Denom $\leq \sqrt{(w_0)^2 + n\rho^2}$

Some Observations

- Numerator of G grows as n
- Denominator of G grows as $\sqrt{n}$

  => Numerator grows faster than denominator
- If PTA does not terminate, $G(w_n)$ values will become unbounded.

Some Observations contd.

- But, as $|G(w_n)| \leq |w^*|$ which is finite, this is impossible!
- Hence, PTA has to converge.
- Proof is due to Marvin Minsky.

Convergence of PTA proved

- *Whatever be the initial choice of weights and whatever be the vector chosen for testing, PTA converges if the vectors are from a linearly separable function.*

Feedforward Network

Limitations of perceptron

- Non-linear separability is all pervading
- Single perceptron does not have enough computing power
- Eg: XOR cannot be computed by perceptron

Solutions

- Tolerate error (Ex: *pocket algorithm* used by connectionist expert systems).
  - Try to get the best possible hyperplane using only perceptrons
- Use higher dimension surfaces

  Ex: Degree - 2 surfaces like parabola
- Use layered network

Pocket Algorithm

- Algorithm evolved in 1985 – essentially uses PTA
- Basic Idea:
  - Always preserve the best weight obtained so far in the "pocket"
  - Change weights, if found better (i.e. changed weights result in reduced error).

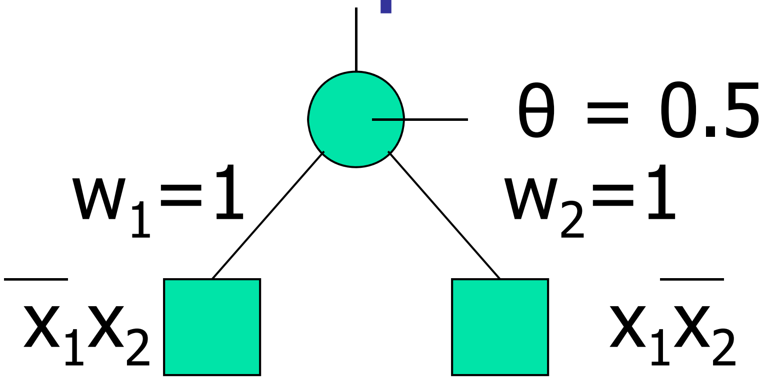
XOR using 2 layers

$$x_1 \oplus x_2 = \left(x_1 \overline{x_2}\right) \vee \left(\overline{x_1} x_2\right)$$

$$= OR(AND(x_1, NOT(x_2)), AND(NOT(x_1), x_2))$$

- Non-LS function expressed as a linearly separable function of individual linearly separable functions.

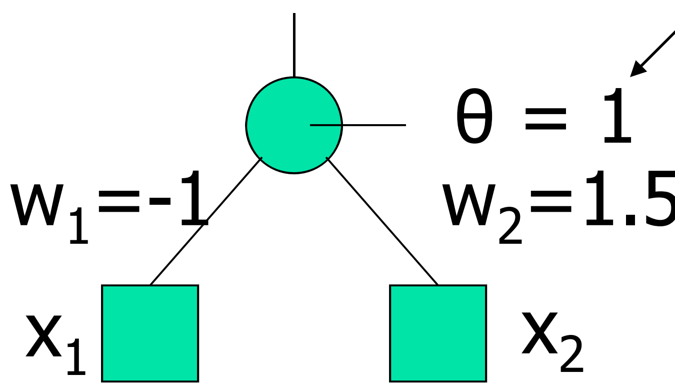
Example - XOR



← Calculation of XOR

x_1	x_2	$\bar{x}_1 x_2$
0	0	0
0	1	1
1	0	0
1	1	0

Calculation of $x_1 \bar{x}_2$



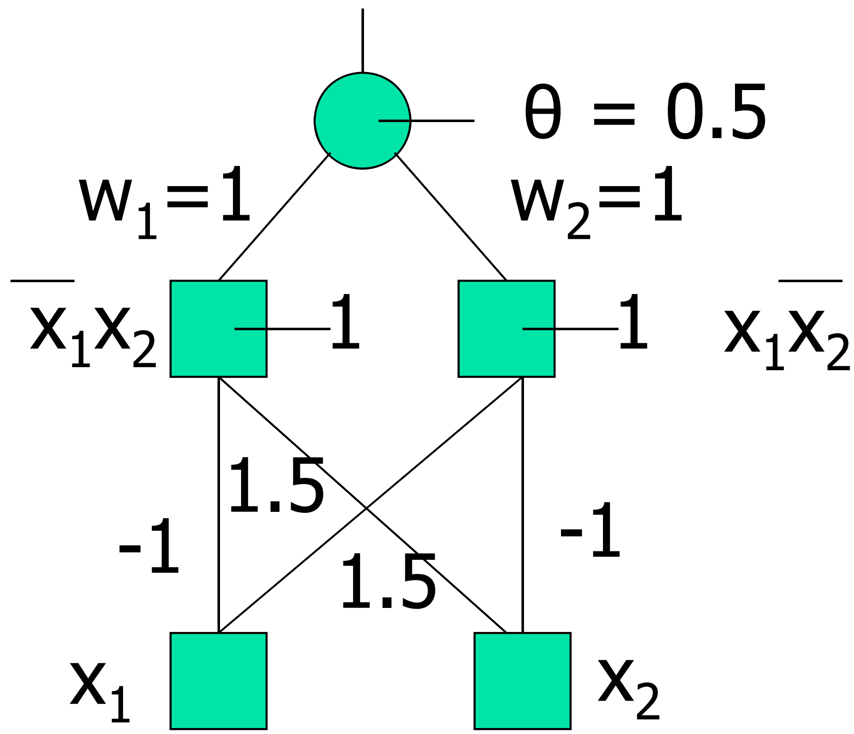
$0 < \Theta$

$w2 \geq \Theta$

$w1 < \Theta$

$w1 + w2 < \Theta$

Example - XOR



Some Terminology

- A multilayer feedforward neural network has
  - Input layer
  - Output layer
  - Hidden layer (assists computation)

Output units and hidden units are called computation units.

Training of the MLP

- Multilayer Perceptron (MLP)

- Question:- How to find weights for the hidden layers when no target output is available?

- Credit assignment problem – to be solved by "*Gradient Descent*"