12th of February, 2014
Wednesday → 9:29 A.M

⇒ Back Propagation

→ Back propagation is a supervised learning algorithm & is mainly used by Multi-Layer-Perceptrons to change the weights connected to the net's hidden neuron layer(s).

→ The back propagation algorithm uses a computed error o/p to change weight values in backward direction.

→ To get this error, a forward propagation phase must have been done before.

→ while propagating in the forward direction, the neurons are activated by using a sigmoid function.

↳ $f(x) = \frac{1}{1+e^{-o/p}}$

⇒ The algorithm works as follows :-

1. Perform the forward propagation phase for an i/p pattern & calculate the o/p error.

2. Change all weight values of each wt. matrix using :

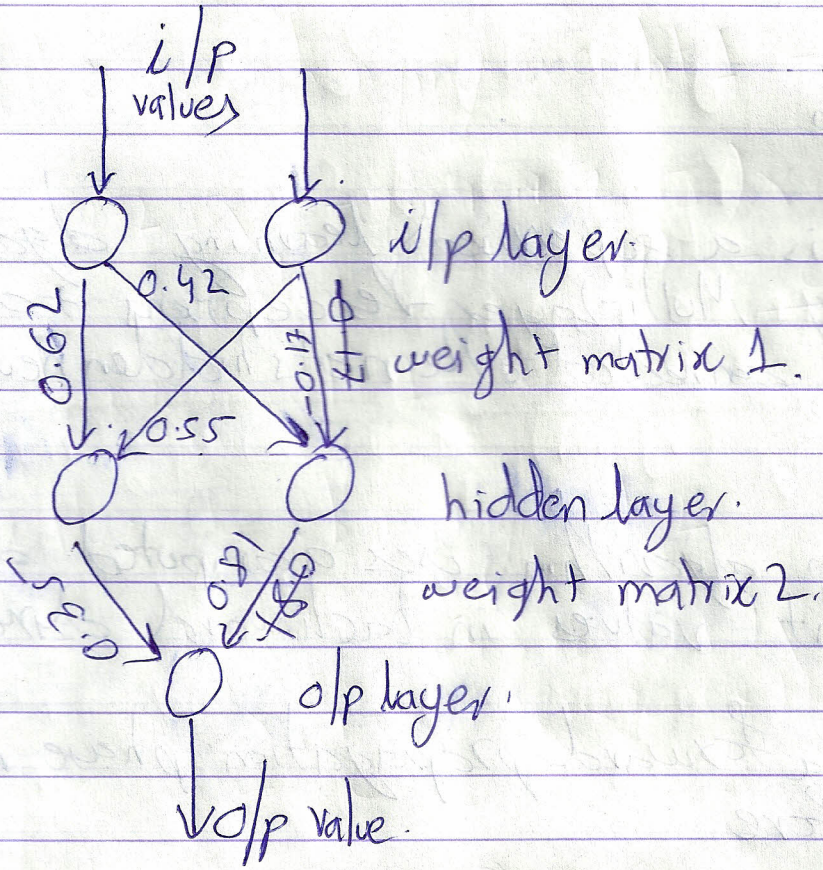
weight(old) + learning rate * o/p error * o/p(neurons i) * o/p (neurons i+1) * (1 - o/p(neurons i+1))

↳ (I)

3. Goto Step 1.

4. Algorithm ends if all o/p patterns match target patterns

Example:-



Pattern to be learned:

i/p		Target
0	1	0
1	1	1

$\eta = 0.25$

→ Step 1 :- i/p → {0, 1}.

⇒ Hidden neurons

→ i/p at hidden neuron 1 → 0×0.62 + 1×0.55 = 0.55.

o/p → $\frac{1}{1+\exp^{-}(0.55)}$ → 0.634135591.

→ i/p at hidden neuron 2 → 0×0.42 + 1×(-0.17) = -0.17.

o/p → $\frac{1}{1+\exp(+0.17)}$ → 0.457602059.

⇒ o/p layer.

i/p → (0.634135591 × 0.35) + (0.457602059 × 0.81)
= 0.592605124.

o/p → $\frac{1}{1+\exp(-0.592605124)}$ = 0.643962658.

⇒ Compute Error.

→ Target = 0.
o/p = 0.643962658.

→ e = 0 − 0.643962658.
e = −0.643962658.

⇒ Start Back Propagation.
2> change wts in weight matrix 2.

From formula (I) :−

→ weight(old) + learning rate * o/p error * o/p(neuron i) * o/p (neuron j) * (1 − o/p (neuron j)).

→ Value for changing wt. 1 :− (0.35).

= 0.25 X (−0.643962658) X 0.634135591
X (0.643962658) X (1 − (0.643962658).)
= −0.023406638.

wt change = 0.35 + (−0.023406638) = 0.326593362.

→ Value for changing wt 2 :− (0.81)

= 0.25 (−0.643962658) X 0.457602059 X
(0.643962658) X (1 − (0.643962658)).
= −0.016890593.

wt. change = 0.818 + (−0.016890593)
= 0.793109407.

→ Change wts of weight matrix 1 :-

→ Value for changing weight 1 :- (0.62)

$= 0.25 \times (-0.64396265 8) \times 0 \times 0.6341355 4) \times (1 - (0.6341355 41)) = 0.$

change in wt 1 = 0.62 + 0 = 0.62 (no change)

→ Value for changing wt 2 :- (0.42).

$= 0.25 \times (-0.64396265 8) \times 0 \times 0.45760205 4 \times (1 - (0.45760205 4)) = 0.$

change in wt. 2 = 0.42 + 0 = 0.42 (no change)

→ Value for changing wt 3 :- (0.55)

$= 0.25 \times (-0.64396265 8) \times 1 \times 0.6341355 41 \times (1 - 0.6341355 41) = -0.037351064$

change in wt 3 = 0.55 - 0.037351064 = 0.51264936.

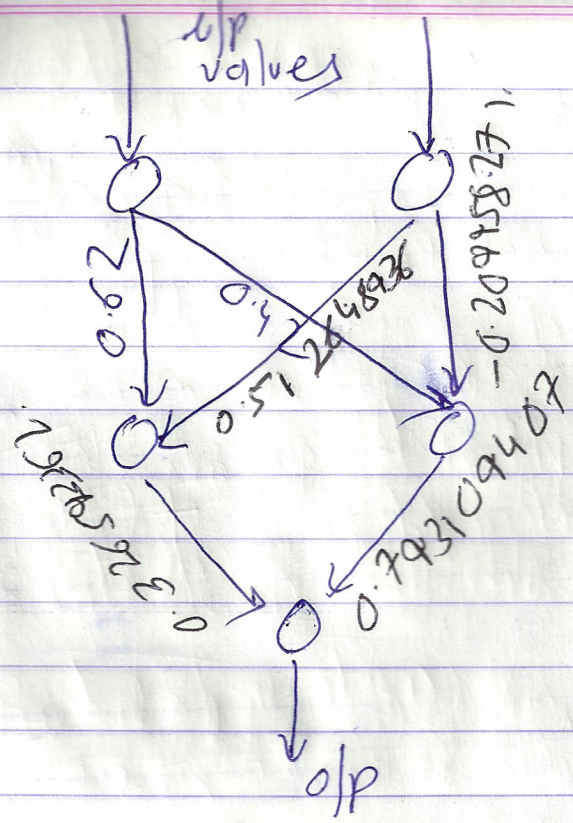
→ Value for changing wt 4 :- (-0.17)

$= 0.25 (-0.64396265 8) \times 1 \times 0.45760205 4 \times (1 - 0.45760205 9) = -0.03945827 1.$

change in wt 4 = -0.17 + (-0.03945827 1)

= -0.20945827 1.

Now,



→ Now, pass {1, 1}.

→ Change wts accordingly

→ This completes one 'Epoch'.

→ Calc net error by adding squared o/p errors of each pattern.

→ Error gets smaller & smaller.

→ Continue till net error 0 (exactly ≈)