Backpropagation Algorithm - Outline

The Backpropagation algorithm comprises a forward and backward pass through the network.

For each input vector $\boldsymbol{x}$ in the training set...

1. Compute the network's response $\boldsymbol{a}$,
   - Calculate the activation of the hidden units $\boldsymbol{h} = \mathrm{sig}(\boldsymbol{x} \bullet \boldsymbol{w}^1)$
   - Calculate the activation of the output units $\boldsymbol{a} = \mathrm{sig}(\boldsymbol{h} \bullet \boldsymbol{w}^2)$

2. Compute the error at each output $\boldsymbol{e}^2 = \boldsymbol{a} - \boldsymbol{t}$

   Take the derivative of the activation $\boldsymbol{d}^2 = \boldsymbol{a}(1-\boldsymbol{a})\, \boldsymbol{e}^2\, \eta$
   This gives us the 'direction' that we should move towards

3. Pass back the error from the output to the hidden layer $\boldsymbol{d}^1 = \boldsymbol{h}(1-\boldsymbol{h})\, \boldsymbol{w}^2\, \boldsymbol{d}^2\, \eta$

4. Adjust the weights from the hidden to output layer. $\boldsymbol{w}^2 = \boldsymbol{w}^2 + (\boldsymbol{h} \times \boldsymbol{d}^2)$

5. Adjust the weights from the inputs to the hidden layer. $\boldsymbol{w}^1 = \boldsymbol{w}^1 + (\boldsymbol{x} \times \boldsymbol{d}^1)$

Backpropagation - The Forward Pass

During the forward pass all weight values are unchanged.
The inputs $x_i,...,x_n$ are multiplied by the receptive weights for each hidden unit - W^1_j

$(W^1_j \bullet X) \rightarrow$ ***net_h*** value of H_j (for H_1 - H)

Each Hidden unit sums the activation that it receives, from the weights that fan into it from the units it is connected to in the input layer.

The summation of the inner product is passed through the (sigmoid) activation function which for the hidden units is $H_j = 1/1+e^{-net\_h}$

The hidden units $h_i,...,h_p$ are multiplied by the receptive weights for each output unit - W^2_k

$(W^2_k \bullet H) \rightarrow$ ***net_o*** value of O_k (for O_1 - O_q)

Each Output unit sums the activation it receives, from the weights that fan into it from the hidden units it is connected to.

The summation of the inner product is passed through the (sigmoid) activation function (also known as the threshold logistic function) which for the output units is $O_k = 1/1+e^{-net\_o}$

Hidden to Output Weight Adjustment

The connectivity within an MLP network is complete.
A weight connects each **j'th** unit in the hidden layer to the **k'th** unit in the output layer.

The result of the forward pass through the net is an output value a_k for each **k'th** output unit.

This value is subtracted from its equivalent value in the Target giving

the raw error signal $(t_k - a_k)$.

The error measure at each unit is calculated by multiplying the raw error by the 1st derivative (gradient) of the squashing function, (a_k (1 - a_k)),

calculated for the unit k, $\delta_k = a_k (1 - a_k) (t_k - a_k)$

The δ_k value is then multiplied by the value of **out** for the **j'th** unit in the *preceding* (hidden) layer and by η, which scales the value by which the weights should be adjusted, to give ΔW.

$$\Delta W_{kj} = \eta \, \delta_k \, h_j$$

The weights are then changed using this $W_{kj}(t+1) = W_{kj}(t) + \Delta W$

Where W_k is the weight value from the **j'th** unit in the hidden layer to the **k'th** unit in the output layer , $W_{kj}(t+1)$ is the new value of the weight, δ_k is the value of for the **k'th** unit in the output layer, η is the learning rate, a_k is the output value for the **j'th** unit in the hidden layer.

Passing Back the Error – Output to Hidden

The Hidden units have no target vector, therefore it is not possible to calculate an error for them by subtracting the output from the target

BP propagates the error computed over the output layer *back* through the network to the hidden units.

To achieve this the value calculated over the output layer, is *propagated* back through the *same* weights to generate a value for each hidden unit.
This process is performed for all units between the hidden and output layer

During the reverse pass, the weights multiply the value from the ***k'th*** unit in the output and pass it back to the ***j'th*** hidden unit.

The value of for the ***j'th*** hidden unit is produced by summing all such products from each output unit, and then multiplying by the derivative of the squashing function.

$$D^1 j = h_j(1-h_j) \Sigma_{k=0} w^2{}_{jk} d^2{}_k$$

Input to Hidden Weight Adjustment

The propagated values are used in turn to adjust the Input to Hidden weights.
The Input to Hidden weights are adjusted using this Delta, as for the output layer
The result is the value by which the weights should be adjusted, i.e. ΔW.

To change the weights between the inputs and hidden units as follows

$$\Delta W_{ji} = \eta\, \delta_{ji}\, x_i \qquad W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}$$

Note

GDR usually includes a *momentum* parameter
This speeds up the training time and stabilises the learning process by scaling the current weight adjustment so that it is proportional to previous weight change.
This means that the algorithm becomes directionally optimistic. When it finds a clear descent it progresses rapidly.

The learning equations including momentum are shown here for the input to hidden unit weights. The same applies for the second layer of weights

$$\Delta W_{ji} = \eta\, \delta_{ji}\, x_i + \alpha \Delta W_{ji}$$
$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}$$

Example

Given the following weight values

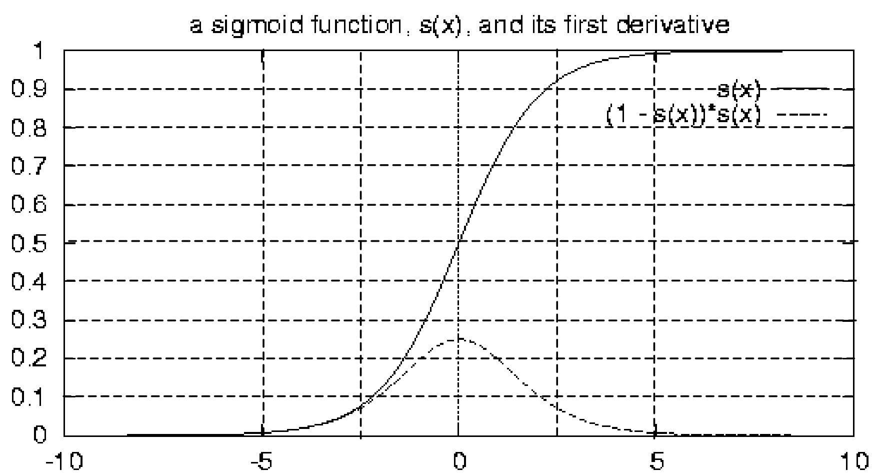
Input to Hidden 1 -5.851 -5.880 +2.184
Input to Hidden 2 -3.809 -3.813 +5.594
Hidden to Output -7.815 +7.508 -3.417

apply the Generalised Delta Rule For 3 cycles to the XOR training set,

0 1 → 1
1 0 → 1
1 1 → 0
0 0 → 0

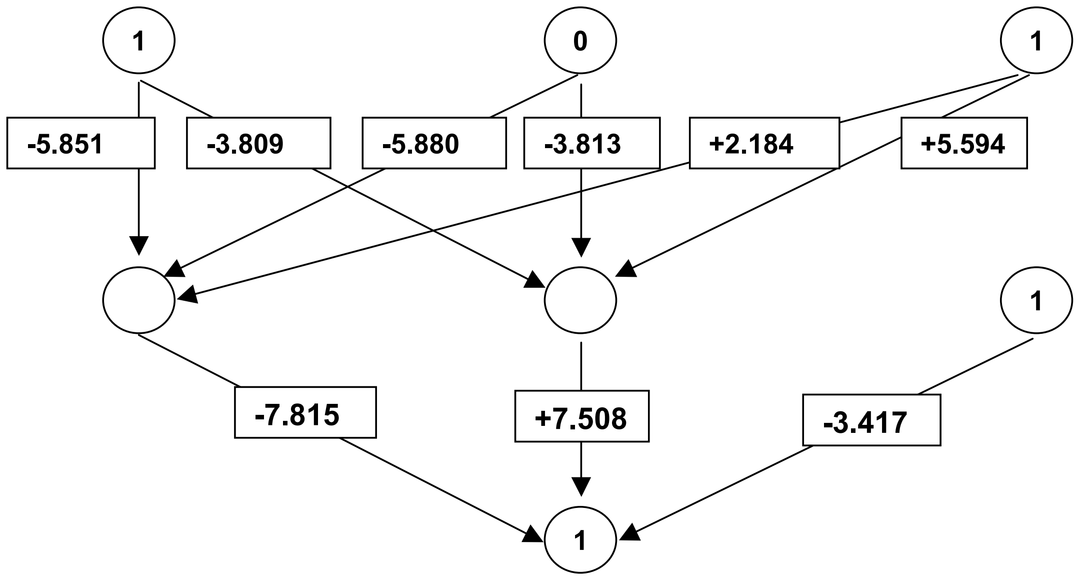
use an η of 0.4.

Use the following graph or tables to approximate the sigmoid and its derivative



Look at the example and use the template provided

Example



Pattern No.	***(1)***				
Input: Target		**1**	**0**	**1**	**.product**
Weights:Inp-Hid0		-5.85100	-5.88	2.184	-3.667
Weights:Inp-Hid1		-3.80900	-3.813	5.594	1.785
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$	0.02492	0.85631	**1.00**	
Weights:Hid-Out		-7.81500	7.508	-3.417	2.8174076
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$		0.94360		
Target	t		**1.00000**		
Error	$e^2 = (a - t)$		**-0.05640**		
Out Derivative	$a(1-a)$		**0.05321**		
Delta Out: 1	$d^2 = a(1-a) \text{ x } e^2 \text{ x } \eta$		-0.0012		
Delta Out: 2	$d^2 = d^2 * h$	-0.00003	-0.00103	-0.00120	
Delta Hid	$d^1 = h(1-h) \text{ x } w^2 \text{ x } d^2 \text{ x } \eta$	0.00000	-0.00038	0.00000	
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \text{ x } d^1)$	-5.85100	-5.88000	2.18400	
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \text{ x } d^1)$	-3.80900	-3.81300	5.59400	
NewWts:Hid-Out	$w^2 = w^2 + (h \text{ x } d^2)$	-7.81500	7.50767	-3.41700	

raw	sigmoid	1st deriv	raw	sigmoid	1st deriv	raw	sigmoid	1st deriv	raw	sigmoid	1st deriv
-10	0.00005	0.00005	-5	0.00669	0.00665	0	0.50000	0.25000	5	0.99331	0.00665
-9.9	0.00005	0.00005	-4.9	0.00739	0.00734	0.1	0.52498	0.24938	5.1	0.99394	0.00602
-9.8	0.00006	0.00006	-4.8	0.00816	0.00810	0.2	0.54983	0.24752	5.2	0.99451	0.00546
-9.7	0.00006	0.00006	-4.7	0.00901	0.00893	0.3	0.57444	0.24446	5.3	0.99503	0.00494
-9.6	0.00007	0.00007	-4.6	0.00995	0.00985	0.4	0.59868	0.24026	5.4	0.99550	0.00448
-9.5	0.00007	0.00007	-4.5	0.01099	0.01087	0.5	0.62246	0.23500	5.5	0.99593	0.00405
-9.4	0.00008	0.00008	-4.4	0.01213	0.01198	0.6	0.64565	0.22879	5.6	0.99632	0.00367
-9.3	0.00009	0.00009	-4.3	0.01339	0.01321	0.7	0.66818	0.22171	5.7	0.99666	0.00332
-9.2	0.00010	0.00010	-4.2	0.01478	0.01456	0.8	0.68997	0.21391	5.8	0.99698	0.00301
-9.1	0.00011	0.00011	-4.1	0.01630	0.01604	0.9	0.71094	0.20550	5.9	0.99727	0.00272
-9	0.00012	0.00012	-4	0.01799	0.01766	1	0.73105	0.19661	6	0.99753	0.00247
-8.9	0.00014	0.00014	-3.9	0.01984	0.01945	1.1	0.75025	0.18737	6.1	0.99776	0.00223
-8.8	0.00015	0.00015	-3.8	0.02188	0.02140	1.2	0.76852	0.17790	6.2	0.99797	0.00202
-8.7	0.00017	0.00017	-3.7	0.02413	0.02355	1.3	0.78583	0.16830	6.3	0.99817	0.00183
-8.6	0.00018	0.00018	-3.6	0.02660	0.02589	1.4	0.80218	0.15869	6.4	0.99834	0.00166
-8.5	0.00020	0.00020	-3.5	0.02932	0.02846	1.5	0.81757	0.14915	6.5	0.99850	0.00150
-8.4	0.00022	0.00022	-3.4	0.03230	0.03126	1.6	0.83201	0.13977	6.6	0.99864	0.00136
-8.3	0.00025	0.00025	-3.3	0.03557	0.03431	1.7	0.84553	0.13061	6.7	0.99877	0.00123
-8.2	0.00027	0.00027	-3.2	0.03917	0.03764	1.8	0.85814	0.12173	6.8	0.99889	0.00111
-8.1	0.00030	0.00030	-3.1	0.04311	0.04125	1.9	0.86989	0.11319	6.9	0.99899	0.00101
-8	0.00034	0.00034	-3	0.04743	0.04518	2	0.88079	0.10500	7	0.99909	0.00091
-7.9	0.00037	0.00037	-2.9	0.05216	0.04944	2.1	0.89090	0.09720	7.1	0.99918	0.00082
-7.8	0.00041	0.00041	-2.8	0.05733	0.05404	2.2	0.90024	0.08981	7.2	0.99925	0.00075
-7.7	0.00045	0.00045	-2.7	0.06298	0.05901	2.3	0.90887	0.08282	7.3	0.99932	0.00067
-7.6	0.00050	0.00050	-2.6	0.06914	0.06436	2.4	0.91682	0.07626	7.4	0.99939	0.00061
-7.5	0.00055	0.00055	-2.5	0.07586	0.07011	2.5	0.92414	0.07011	7.5	0.99945	0.00055
-7.4	0.00061	0.00061	-2.4	0.08318	0.07626	2.6	0.93086	0.06436	7.6	0.99950	0.00050
-7.3	0.00068	0.00067	-2.3	0.09113	0.08282	2.7	0.93702	0.05901	7.7	0.99955	0.00045
-7.2	0.00075	0.00075	-2.2	0.09976	0.08981	2.8	0.94267	0.05404	7.8	0.99959	0.00041
-7.1	0.00082	0.00082	-2.1	0.10910	0.09720	2.9	0.94784	0.04944	7.9	0.99963	0.00037
-7	0.00091	0.00091	-2	0.11921	0.10500	3	0.95257	0.04518	8	0.99966	0.00034
-6.9	0.00101	0.00101	-1.9	0.13011	0.11319	3.1	0.95689	0.04125	8.1	0.99970	0.00030
-6.8	0.00111	0.00111	-1.8	0.14186	0.12173	3.2	0.96083	0.03764	8.2	0.99973	0.00027
-6.7	0.00123	0.00123	-1.7	0.15447	0.13061	3.3	0.96443	0.03431	8.3	0.99975	0.00025
-6.6	0.00136	0.00136	-1.6	0.16799	0.13977	3.4	0.96770	0.03126	8.4	0.99978	0.00022
-6.5	0.00150	0.00150	-1.5	0.18243	0.14915	3.5	0.97068	0.02846	8.5	0.99980	0.00020
-6.4	0.00166	0.00166	-1.4	0.19782	0.15869	3.6	0.97340	0.02589	8.6	0.99982	0.00018
-6.3	0.00183	0.00183	-1.3	0.21417	0.16830	3.7	0.97587	0.02355	8.7	0.99983	0.00017
-6.2	0.00203	0.00202	-1.2	0.23148	0.17790	3.8	0.97812	0.02140	8.8	0.99985	0.00015
-6.1	0.00224	0.00223	-1.1	0.24975	0.18737	3.9	0.98016	0.01945	8.9	0.99986	0.00014
-6	0.00247	0.00247	-1	0.26895	0.19661	4	0.98201	0.01766	9	0.99988	0.00012
-5.9	0.00273	0.00272	-0.9	0.28906	0.20550	4.1	0.98370	0.01604	9.1	0.99989	0.00011
-5.8	0.00302	0.00301	-0.8	0.31003	0.21391	4.2	0.98522	0.01456	9.2	0.99990	0.00010
-5.7	0.00334	0.00332	-0.7	0.33182	0.22171	4.3	0.98661	0.01321	9.3	0.99991	0.00009
-5.6	0.00368	0.00367	-0.6	0.35435	0.22879	4.4	0.98787	0.01198	9.4	0.99992	0.00008
-5.5	0.00407	0.00405	-0.5	0.37754	0.23500	4.5	0.98901	0.01087	9.5	0.99993	0.00007
-5.4	0.00450	0.00448	-0.4	0.40132	0.24026	4.6	0.99005	0.00985	9.6	0.99993	0.00007
-5.3	0.00497	0.00494	-0.3	0.42556	0.24446	4.7	0.99099	0.00893	9.7	0.99994	0.00006
-5.2	0.00549	0.00546	-0.2	0.45017	0.24752	4.8	0.99184	0.00810	9.8	0.99994	0.00006
-5.1	0.00606	0.00602	-0.1	0.47502	0.24938	4.9	0.99261	0.00734	9.9	0.99995	0.00005
-5	0.00669	0.00665	0	0.50000	0.25000	5	0.99331	0.00665	10	0.99995	0.00005

Iteration 1

Pattern No.	***(1)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0		-5.85100	-5.88	2.184	
Weights:Inp-Hid1		-3.80900	-3.813	5.594	
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			**1.00**	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Pattern No.	***(2)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			**1**	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Pattern No.	***(3)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			**1**	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Iteration 2

Pattern No.	*(1)*				
Input: Target				1	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			1	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Pattern No.	*(2)*				
Input: Target				1	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			1	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Pattern No.	*(3)*				
Input: Target				1	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	$h = sigmoid(x \bullet w^1)$			1	
Weights:Hid-Out					
Out (Sigmoid):	$a = sigmoid(h \bullet w^2)$				
Target	t				
Error	$e^2 = (a - t)$				
Out Derivative	$a(1-a)$				
Delta Out: 1	$d^2 = a(1-a) \times e^2 \times \eta$		-		
Delta Out: 2	$d^2 = d^2 * h$				
Delta Hid	$d^1 = h(1-h) \times w^2 \times d^2 \times \eta$				
NewWts:Inp-Hid0:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Inp-Hid1:	$w^1 = w^1 + (x \times d^1)$				
NewWts:Hid-Out	$w^2 = w^2 + (h \times d^2)$				

Iteration 3

Pattern No.	***(1)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	*h = sigmoid(x• w^1)*			**1**	
Weights:Hid-Out					
Out (Sigmoid):	*a = sigmoid(h• w^2)*				
Target	*t*				
Error	*e^2 = (a - t)*				
Out Derivative	*a(1-a)*				
Delta Out: 1	*d^2 = a(1-a) x e^2 x η*		-		
Delta Out: 2	*d^2 = d^2 * h*				
Delta Hid	*d^1= h(1-h) x w^2 x d^2 x η*				
NewWts:Inp-Hid0:	*w^1 = w^1+ (x x d^1)*				
NewWts:Inp-Hid1:	*w^1 = w^1+ (x x d^1)*				
NewWts:Hid-Out	*w^2 = w^2+ (h x d^2)*				

Pattern No.	***(2)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	*h = sigmoid(x• w^1)*			**1**	
Weights:Hid-Out					
Out (Sigmoid):	*a = sigmoid(h• w^2)*				
Target	*t*				
Error	*e^2 = (a - t)*				
Out Derivative	*a(1-a)*				
Delta Out: 1	*d^2 = a(1-a) x e^2 x η*		-		
Delta Out: 2	*d^2 = d^2 * h*				
Delta Hid	*d^1= h(1-h) x w^2 x d^2 x η*				
NewWts:Inp-Hid0:	*w^1 = w^1+ (x x d^1)*				
NewWts:Inp-Hid1:	*w^1 = w^1+ (x x d^1)*				
NewWts:Hid-Out	*w^2 = w^2+ (h x d^2)*				

Pattern No.	***(3)***				
Input: Target				**1**	**.product**
Weights:Inp-Hid0					
Weights:Inp-Hid1					
Hid: (Sigmoid)	*h = sigmoid(x• w^1)*			**1**	
Weights:Hid-Out					
Out (Sigmoid):	*a = sigmoid(h• w^2)*				
Target	*t*				
Error	*e^2 = (a - t)*				
Out Derivative	*a(1-a)*				
Delta Out: 1	*d^2 = a(1-a) x e^2 x η*		-		
Delta Out: 2	*d^2 = d^2 * h*				
Delta Hid	*d^1= h(1-h) x w^2 x d^2 x η*				
NewWts:Inp-Hid0:	*w^1 = w^1+ (x x d^1)*				
NewWts:Inp-Hid1:	*w^1 = w^1+ (x x d^1)*				
NewWts:Hid-Out	*w^2 = w^2+ (h x d^2)*				