The input pattern is (0.52, 0.75, 0.97), and the desired output pattern is (0.24, 0.17, 0.65). The initial weight matrices are as follows:

M-1 Matrix of weights from input layer to hidden layer

```
  0.6     - 0.4
  0.2       0.8
- 0.5       0.3
```

M-2 Matrix of weights from hidden layer to output layer

```
-0.90       0.43       0.25
 0.11     - 0.67     - 0.75
```

The threshold values (or bias) for neurons in the hidden layer are 0.2 and 0.3, while those for the output neurons are 0.15, 0.25, and 0.05, respectively.

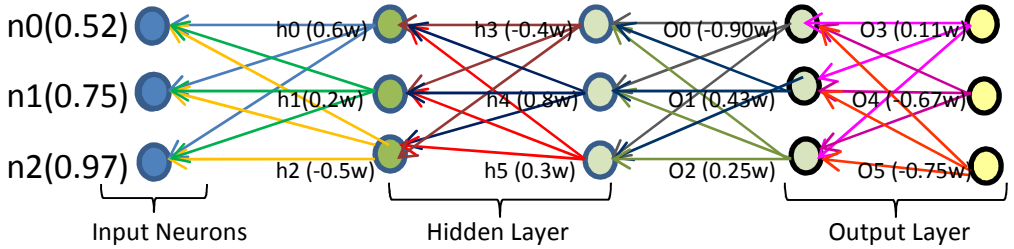
We calculate the output of each neuron in the hidden and output layers as follows. We add a bias or threshold value to the activation of a neuron (call this result x) and use the **sigmoid** function below to get the output.

$$f(x) = 1/ (1 + e^{-x})$$

Learning parameters used are 0.2 for the connections between the hidden layer neurons and output neurons and 0.15 for the connections between the input neurons and the neurons in the hidden layer. These values as you recall are the same as in the previous illustration, to make it easy for you to follow the calculations by comparing them with similar calculations in the preceding sections.

Answer:

Schematic conceptual graph for neurals net of the above mentioned case



Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

Vector data for the Input Neurons $\mathbf{N}^{\rightarrow}$ = (0.52 , 0.75, 0.97)
Vector data for the Desired Output pattern $\boldsymbol{DO}^{\rightarrow}$ = (0.24 , 0.17 , 0.65)

Weights Set for the Hidden Layer 00 Neurons WH00 = { 0.6 , 0.2 , -0.5 }
Weights Set for the Hidden Layer 01 Neurons WH01 = { -0.4 , 0.8 , 0.3 }

Weights Set for the Output Layer 00 Neurons WO00 = { -0.90 , 0.43, 0.25 }
Weights Set for the Output Layer 01 Neurons WO01 = { 0.11 , -0.67, -0.75 }

Bias Value for Neurons in the Hidden Layers is 0.2 and 0.3 respectively, while Threshold limit for Neurons in the Output Layer are 0.15 , 0.25 and 0.05 for each neuron in the row respectively.

Learning rates (μ) used for connection between the input and hidden layer neurons is set to be 0.15 , while connection for the layer between hidden and output neurons is set to be 0.2.

Activating Neurons in the Hidden Layer 00 (NH00)

Neurons in the Hidden Layer 00 are h0, h1 and h2 respectively.

F(h0) = F(n0, n1, n2) = ∑(Wi Ii) = { (h0 * n0) + (h0 * n1) + (h0 * n2) }
= { (0.6 * 0.52) + (0.6 * 0.75) + (0.6 * 0.97) }
= 0.312 + 0.45 + 0.582
= 1.344

h0 Bias : 0.2 + 1.344 = 1.544

$FSigmoid(x) = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-1.544}} = 0.824$

h0 Activation (h0_acti) = **0.824**

F(h1) = F(n0, n1, n2) = ∑(Wi Ii) = { (h1 * n0) + (h1 * n1) + (h1 * n2) }
= { (0.2 * 0.52) + (0.2* 0.75) + (0.2*0.97) }
= 0.104 + 0.15 + 0.194
= 0.448

h1 Bias : 0.2 + 0.448 = 0.648
h1 Activation (h1_acti) = **0.656**

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

F(h2) = F(n0, n1, n2) = ∑(Wi Ii) = { (h2 * n0) + (h2 * n1) + (h2 * n2) }
= { (-0.5 * 0.52) + (-0.5 * 0.75) + (-0.5 * 0.97) }
= -0.26 - 0.375 - 0.485
= -1.12

h2 Bias : 0.2 + (-1.12) = - 0.92

h2 Activation (h2_Acti) = **0.284**

Activating Neurons in the Hidden Layer 01 (NH01)

Neurons in the Hidden Layer 01 are h3, h4 and h5 respectively.

F(h3) = F(h0,h1,h2) = ∑(Wi Ii) = { (-0.4 * h0) + (-0.4 * h1) + (-0.4 * h2) }
= (-0.4 * 0.824) + (-0.4 * 0.656) + (-0.4 * 0.284)
= -0.329 - 0.262 - 0.113
= - 0.704

h3 Bias : 0.3 - 0.704 = - 0.404
h3 Activation (h3_acti) = **0.400**

F(h4) = F(h0,h1,h2) = ∑(Wi Ii) = { (0.8 * h0) + (0.8 * h1) + (0.8 * h2) }
= (0.8 * 0.824) + (0.8 * 0.656) + (0.8 * 0.284)
= 0.659 + 0.524 + 0.227
= 1.41

h4 Bias : 0.3 + 1.41 = 1.71
h4 Activation (h4_acti) = **0.846**

F(h5) = F(h0,h1,h2) = ∑(Wi Ii) = { (0.3 * h0) + (0.3 * h1) + (0.3 * h2) }
= (0.3 * 0.824) + (0.3 * 0.656) + (0.3 * 0.284)
= 0.247 + 0.196 + 0.085
= 0.528

h5 Bias : 0.3 + 0.528 = 0.828
h5 Activation (h5_acti) = **0.695**

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

All neurons' stimuli calculated from the above would then be transmitted to the neurons in the end layer which responsible for finalizing and satisfying the overall neurons' stimulus.

Activating Neurons in the Output Layer 00 (NO00)

Neurons in the output layer NO00 : O0 , O1 and O2 respectively.

F(O0) = F(h3,h4,h5) = ∑(Wj Ij) = { (O0 * h3) + (O0 * h4) + (O5 * h5) }
= (-0.90 * 0.400) + (-0.90 * 0.846) + (-0.90 * 0.695)
= - 0.36 - 0.761 - 0.625
= - 1.746

O0 Bias : 0.15 – 1.746 = - 1.596

$FSigmoid(x) = \frac{1}{1+e^{1.596}}$ = 0.168

O0 Activation (O0_acti) = **0.168**

F(O1) = F(h3, h4, h5) = ∑(Wj Ij) = { (O1 * h3) + (O1 * h4) + (O1 * h5) }
= (0.43 * 0.400) + (0.43 * 0.846) + (0.43 * 0.695)
= 0.172 + 0.363 + 0.298
= 0.833

O1 Bias : 0.25 + 0.833 = 1.083
O1 Activation (O1_acti) = **0.747**

F(O2) = F(h3 , h4, h5) = ∑(Wj Ij) = { (O2 * h3) + (O2 * h4) + (O2 * h5) }
= (0.25 * 0.400) + (0.25 * 0.846) + (0.25 * 0.695)
= 0.1 + 0.211 + 0.173
= 0.484

O2 Bias : 0.05 + 0.484 = 0.534
O2 Activation (O2_acti) = **0.630**

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

Activating Neurons in the Output Layer 01 (NO01)

Neurons in the output layer 01 are O3 , O4 and O5 respectively.

F(O3) = F(O0, O1, O2) = ∑(Wj Ij) = { (O3 * O0) + (O3 * O1) + (O3 * O2) }
= (0.11 * 0.168) + (0.11 * 0.747) + (0.11 * 0.630)
= 0.018 + 0.082 + 0.069
= 0.169

O3 Bias : 0.15 + 0.169 = 0.319
O3 Activation (O3_acti) = **0.579**

F(O4) = F(O0, O1, O2) = ∑(Wj Ij) = { (O4 * O0) + (O4 * O1) + (O4 * O2) }
= (-0.67 * 0.168) + (-0.67 * 0.747) + (-0.67 * 0.630)
= - 0.112 - 0.500 - 0.422
= - 1.034

O4 Bias : 0.25 – 1.034 = - 0.784
O4 Activation (O4_acti) = **0.313**

F(O5) = F(O0, O1, O2) = ∑(Wj Ij) = { (O5 * O0) + (O5 * O1) + (O5 * O2) }
= (-0.75 * 0.168) + (-0.75 * 0.747) + (-0.75 * 0.630)
= - 0.126 - 0.560 - 0.472
= - 1.158

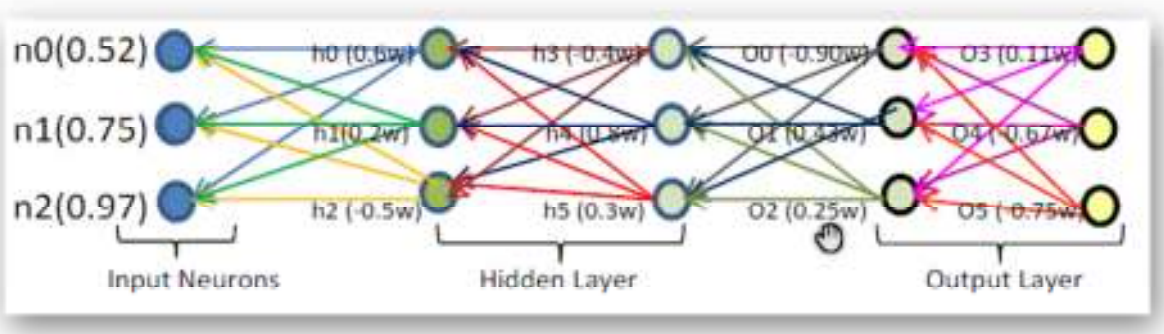
O5 Bias : 0.05 – 1.158 = - 1.108
O5 Activation (O5_acti) = **0.248**

@@

Calculating Errors in the Last End Output Layer

Neurons in the last end layer are O3, O4 and O5 respectively. Look at the diagram charts below depicting position of artificial neurons settled in the layers copulatively arranged for them.

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic



DO$^{\rightarrow}$ = (0.24 , 0.17 , 0.65)

O3 Error = O3 * (1 – O3) * (**DO**$^{\rightarrow}$ - O3)
= 0.579 * (1 – 0.579) * (0.24 – 0.579)
= **- 0.082**

Rate of Changes ΔW03 = ų * O3 Error * O0
= 0.2 * (-0.082) * 0.168
= **- 0.002**

Momentum Rate of Changes in this case is assumed to be always zero, thus (α * Δ (t – 1)) = 0.

W03 New = W03 Old + ΔW03 + (α * Δ (t – 1))
= 0.11 - 0.002 + 0

Updated Weight for O3 : = **0.10**

O4 Error = O4 * (1 – O4) * (**DO**$^{\rightarrow}$ - O4)
= 0.313 * (1 – 0.313) * (0.17 – 0.313)
= **- 0.030**

ΔW04 = ų * O4 Error * O1
= 0.2 * (-0.030) * 0.747
= - 0.004

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

W04 New = W04 Old + ΔW04 + (α * Δ(t – 1))
= -0.67 - 0.004 + 0

O4's Updated Weight = **-0.67**

O5 Error = O5 * (1 – O5) * (**DO**$^{\rightarrow}$ - O5)
= 0.248 * (1 – 0.248) * (0.65 – 0.248)
= **0.074**

ΔW05 = ɥ * O5 Error * O2
= 0.2 * 0.074 * 0.630
= 0.009

W05 New = W05 Old + ΔW05 + (α * Δ(t – 1))
= - 0.75 + 0.009 + 0

O5's Updated Weight = **- 0.74**

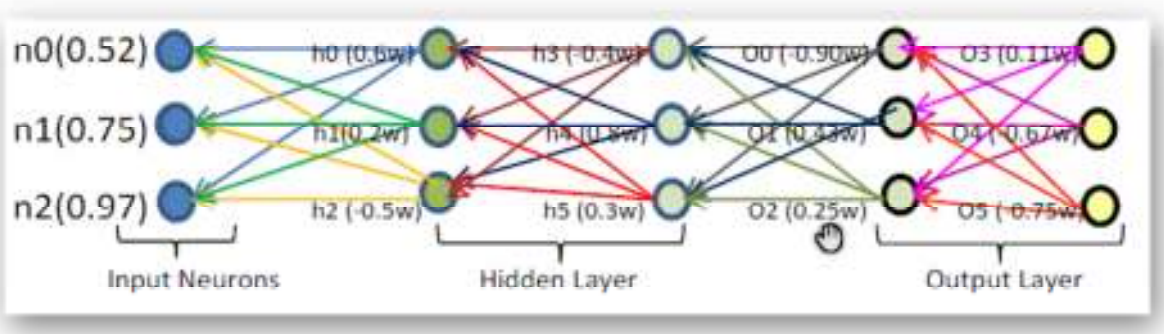
So, the New Updated Weights Set for the Last End Output Layer is { 0.10 , -0.67 , -0.74 }.

WO01 = { 0.10 , -0.67 , -0.74 }

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Calculating Errors in the 1st (First) Output Layer (NO00)

Neurons in the First (1st) Output Layer are O0, O1 and O2 respectively.



O0 Error = O3 Error * WO3 New
= -0.082 * 0.10
O0 Error = **-0.0082**

ΔWO0 = ų * O0 Error * h3
= 0.2 * (-0.0082) * 0.400
= - 0.0006

WO0 New = WO0 Old + ΔWO0 + (α * Δ (t – 1))
= -0.90 - 0.0006 + 0
WO0 New = **-0.90**

O1 Error = O4 Error * W04 New
= -0.030 * (-0.67)
= **0.020**

ΔW01 = ų * O1 Error * h4
= 0.2 * 0.020 * 0.846
= 0.0033

W01 New = WO1 Old + ΔW01 + (α * Δ (t – 1))
= 0.43 + 0.0033 + 0
W01 New = **0.43**

O2 Error = O5 Error * WO5 New
= 0.074 * (-0.74)
= **- 0.054**

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

ΔWO2 = μ * O2 Error * h5
= 0.2 * (-0.054) * 0.695
= **- 0.007**

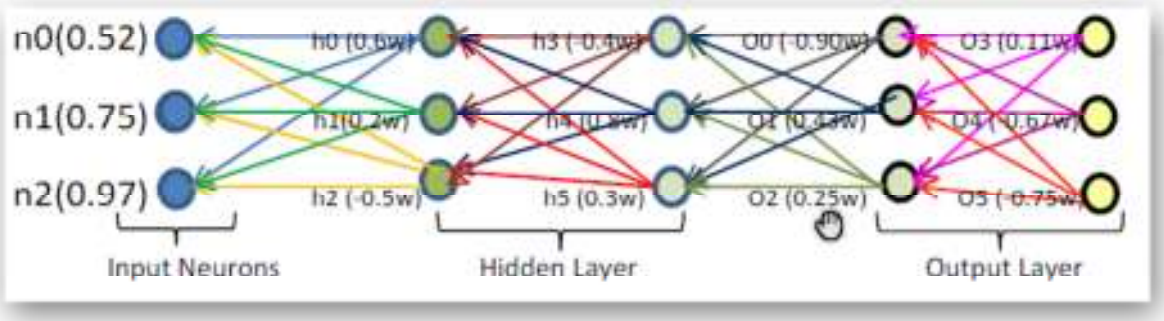
WO2 New = WO2 Old + ΔWO2 + (α * Δ (t – 1))
= 0.25 - 0.007 + 0
WO2 New = **0.24**

New Updated Weights Set for the 1st Layer Ouput is WO00 = {- 0.90 , 0.43 , 0.24}

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Calculating Errors in the Last End Hidden Layer (NH01)

Neurons laid in this layer are h3, h4 and h5 respectively.



h3 Error = O0 Error * WO0 New
= - 0.0082 * - 0.90
= 0.0073

ΔWH3 = μ * h3 Error * h0
= 0.2 * 0.0073 * 0.824
= 0.0012

WH3 New = WH3 Old + ΔWH3 + (α * Δ (t – 1))
= -0.4 + 0.0012 + 0
= **-0.39**

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

h4 Error = O1 Error * WO1 New
= 0.020 * 0.43
= 0.0086

ΔWH4 = ɥ * h4 Error * h1
= 0.2 * 0.0086 * 0.656
= 0.0011

WH4 New = WH4 Old + ΔWH4 + (α * Δ (t – 1))
= 0.8 + 0.0011 + 0
= **0.8**

---

h5 Error = O2 Error * W02 New
= -0.054 * 0.24
= -0.012

ΔWH5 = ɥ * h5 Error * h2
= 0.2 * -0.012 * 0.284
= -0.0006

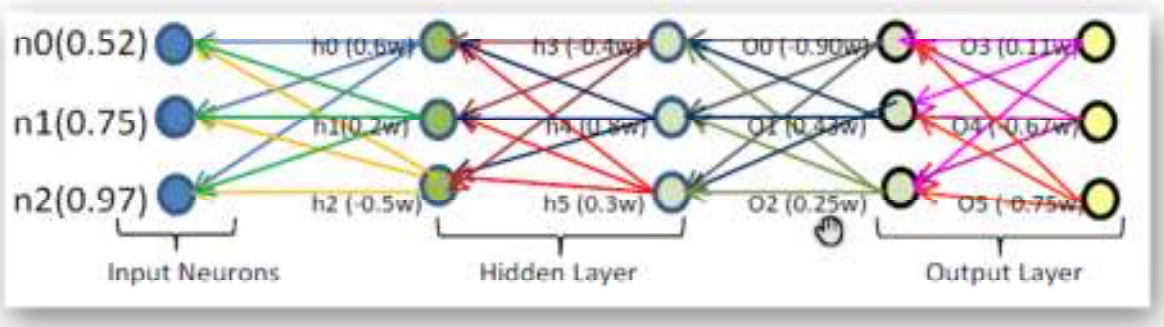
WH5 New = WH5 Old + ΔWH5
= 0.3 - 0.0006
= **0.29**

So, the new adjusted weights set for this end hidden layer (NH01) is WH01 = { -0.39 , 0.8 , 0.29 }

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Calculating Errors in the First (1^{st}) Hidden Layer (NH00)

Neurons in this layer are h0 , h1 and h2 respectively. The determined Learning Rate (ɥ) for this layer to the input layer neurons is set to be 0.15.



h0 Error = h3 Error * WH3 New
= 0.0073 * -0.39
= -0.0028

ΔWH0 = μ * h0 Error * n0
= 0.15 * -0.0028 * 0.52
= -0.00021

WH0 New = WH0 Old + ΔWH0 + 0
= 0.6 - 0.00021 + 0
= **0.59**

---

h1 Error = h4 Error * WH4 New
= 0.0086 * 0.8
= 0.0068

ΔWH1 = μ * h1 Error * n1
= 0.15 * 0.0068 * 0.75
= 0.00076

WH1 New = WH1 Old + ΔWH1 + 0
= 0.2 + 0.00076 + 0
= **0.2**

---

h2 Error = h5 Error * WH5 New
= -0.012 * 0.29
= -0.0034

ΔWH2 = μ * h2 Error * n2
= 0.15 * -0.0034 * 0.97
= -0.00049

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic

WH2 New = WH2 Old + ΔWH2 + 0
= -0.5 - 0.00049 + 0
= **-0.5**

So, the new adjusted weights set for this hidden layer NH00 is:
WH00 = { 0.59 , 0.2 , -0.5 }

Sources of Reference: C++ Fuzzy Network by V.Rao
Neural Networks and Back Propagation Algorithm by M.CilimKovic