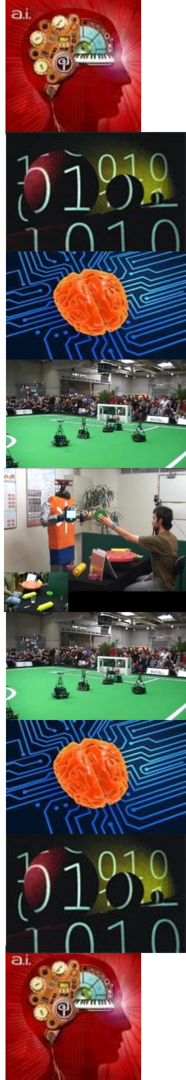
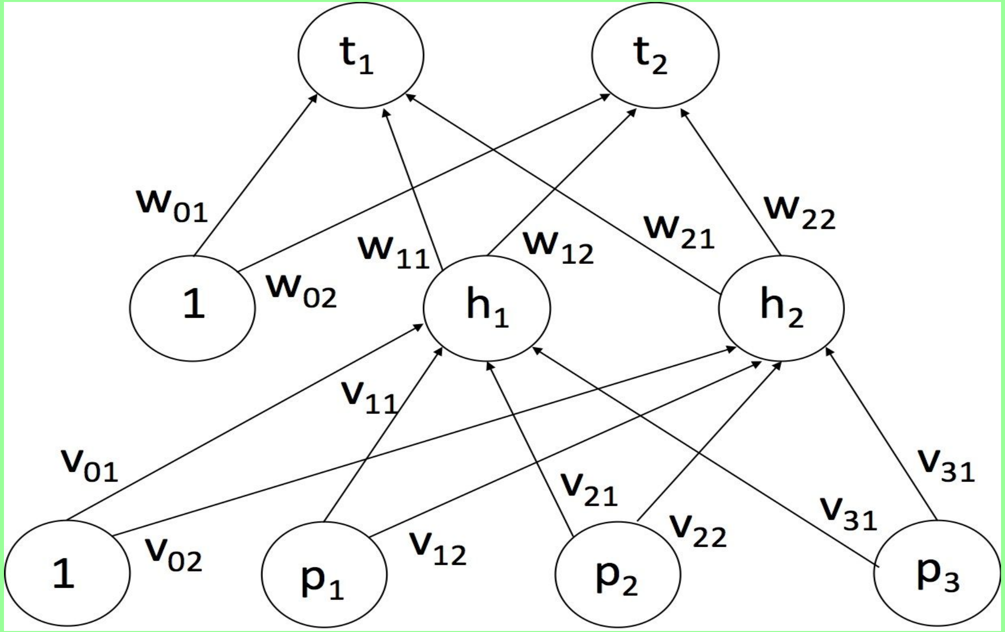
Backpropagation

Introduction

- Variations of the backpropagation algorithm
- Architecture:
  - Input layer
  - Hidden layers
  - Output layer
  - Bias
- Activation functions
  - Example binary sigmoid
  - Differentiable



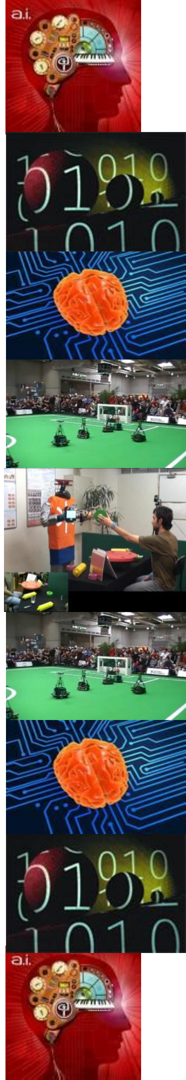
Multilayer Architecture Example



Backpropagation Learning Algorithm

Algorithm 2 Backpropagation learning algorithm

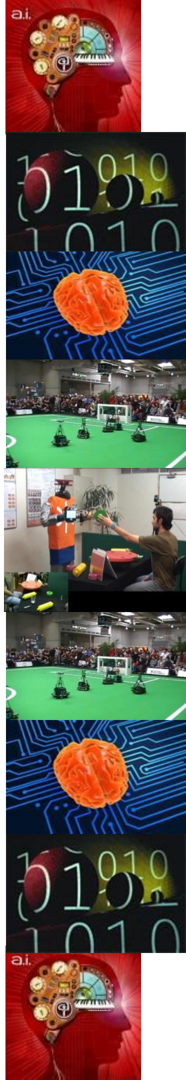
```
Set the weights and bias to zero or small random values.
while The stopping condition is not met do
    Perform feedforward learning
    Backpropagation of error
    Update weights
end while
```



Activation Functions

- Hidden layer

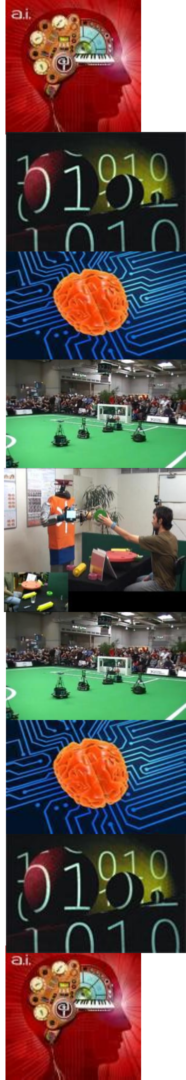
- Output layer



Feedforward Learning

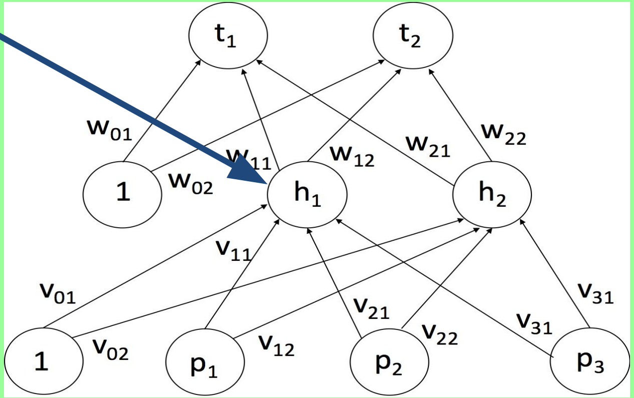
Algorithm 3 Perform feedforward learning

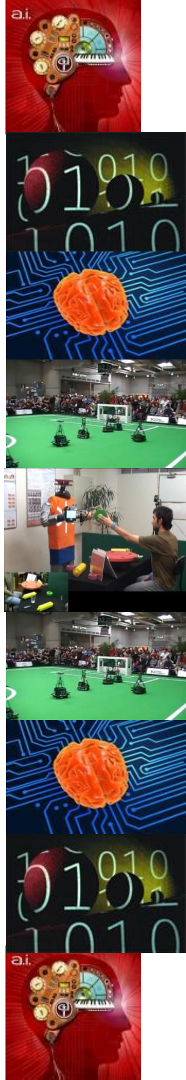
1: Calculate n_1 for each node in the hidden layer
2: Calculate the activation $f(n1)$ for each node in the hidden layer
3: Calculate n_2 for each node in the output layer
4: Calculate the activation $f(n2)$ for each node in the output layer



FeedForward Hidden Layer

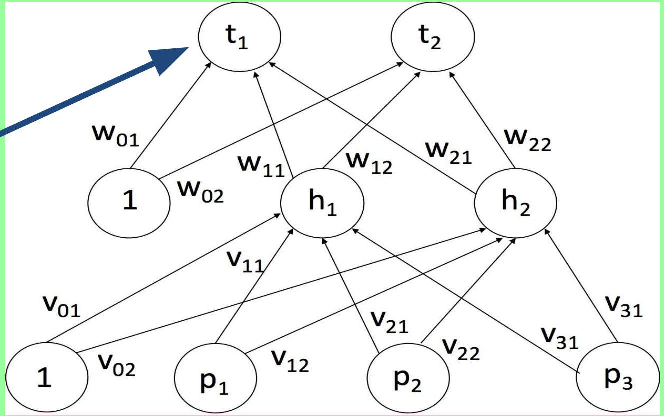
$$n_{1j} = v_{0j} + \sum_{l=1}^{n} v_{lj} * p_l$$

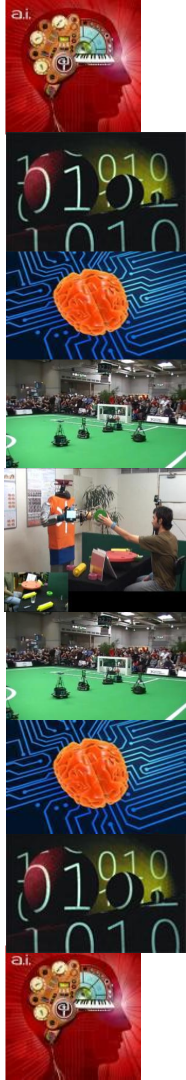




Feedforward Output Layer

$$n_{2m} = w_{0m} + \sum_{i=1}^{j} w_{im} * f(n_{1i})$$

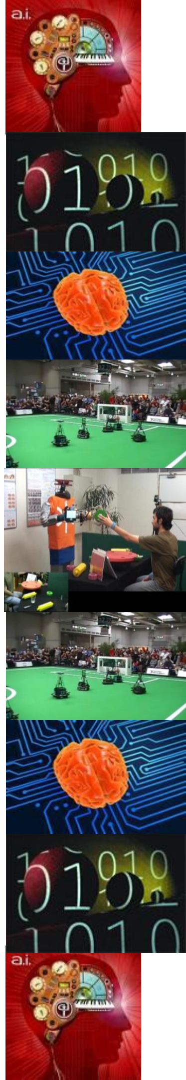




Backpropagation Error

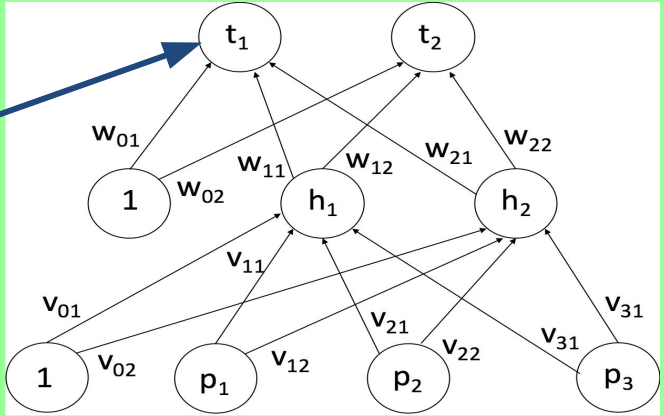
Algorithm 4 Backpropagation of error

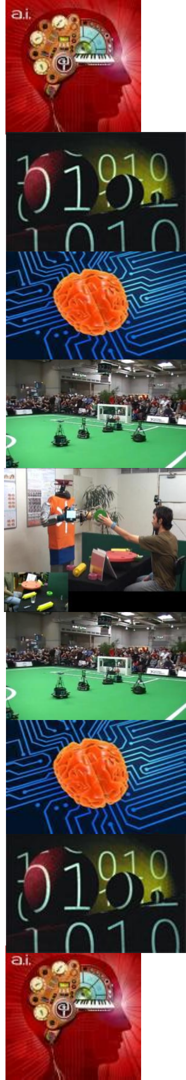
```
Calculate the error information term for each node in the output layer
Calculate the weight correction term for each node in the output layer
Calculate the bias correction term for each node in the output layer
Calculate the sum of delta inputs for each node in the hidden layer
Calculate the error information term for each node in the hidden layer
Calculate the weight error term for each node in the hidden layer
Calculate the bias error term for each node in the hidden layer
```



Backpropagation- Output Layer Error

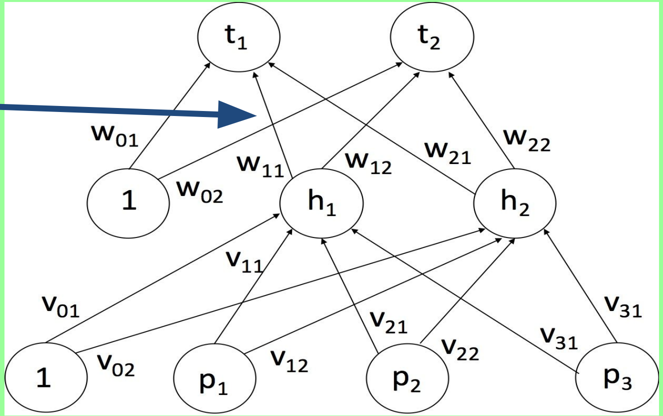
$$\delta_k = (t_k - f(n_{2k}))f'(n_{2k})$$



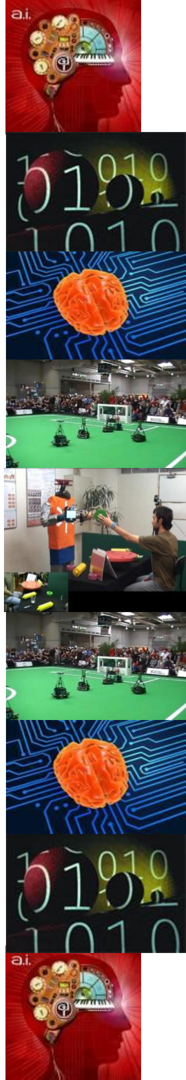


Backpropagation- Weight Correction

$$\Delta w_{ik} = \alpha \delta_k f(n_{1i})$$

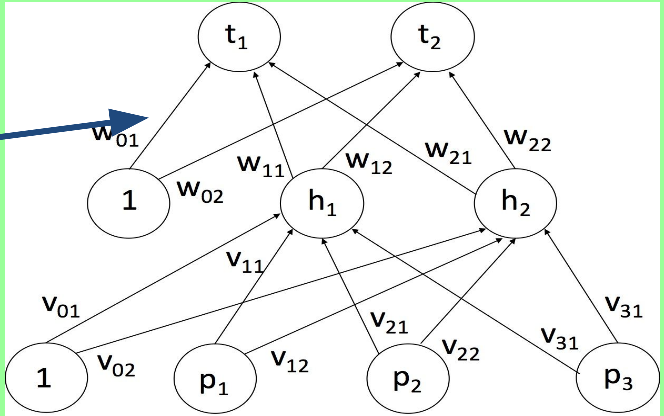


$$w_{ik}(new) = w_{ik}(old) + \Delta w_{ik}$$

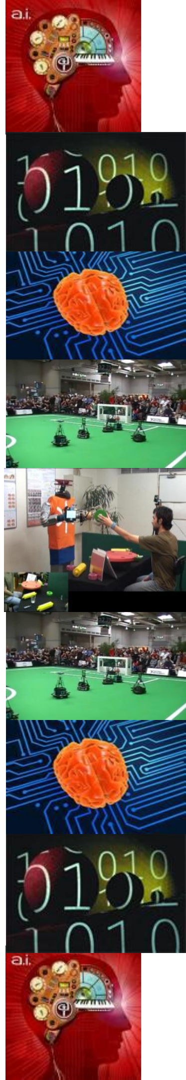


Backpropagation- Output Layer Bias

$$\Delta w_{0k} = \alpha \delta_k$$

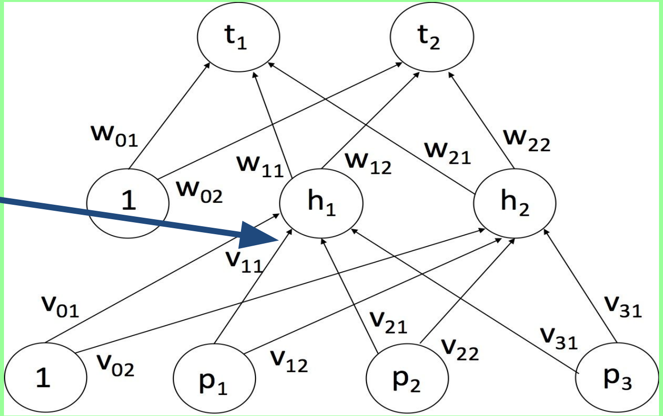


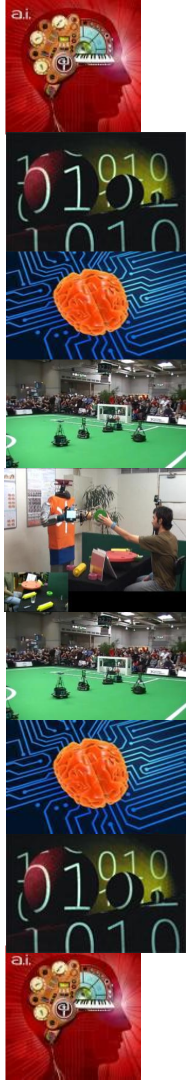
$$w_{ik}(new) = w_{ik}(old) + \Delta w_{ik}$$



Backpropagation- Hidden Layer Sum

$$\delta n_i = \sum_{k=1}^{m} \delta_k w_{ik}$$

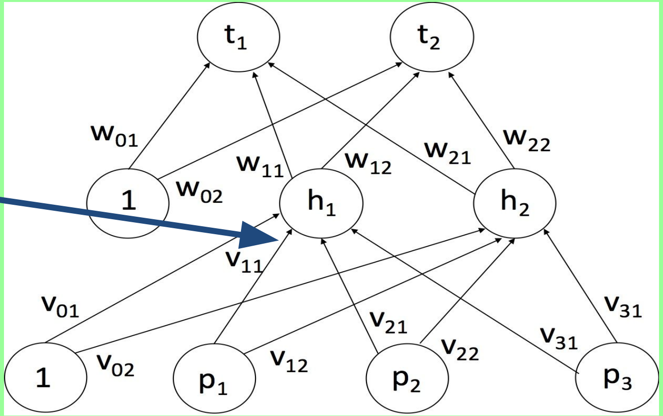


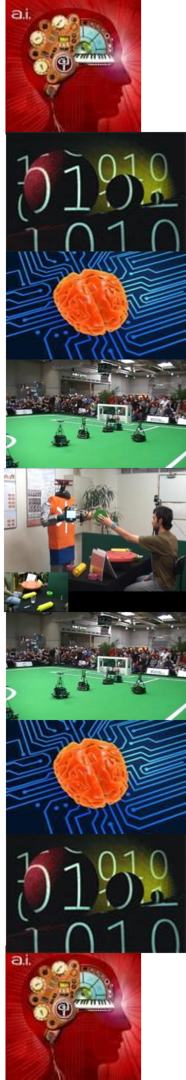


Backpropagation- Hidden Layer

$$\delta n_i = \sum_{k=1}^{m} \delta_k w_{ik}$$

$$\delta_i = \delta n_i f'(n_{1i})$$



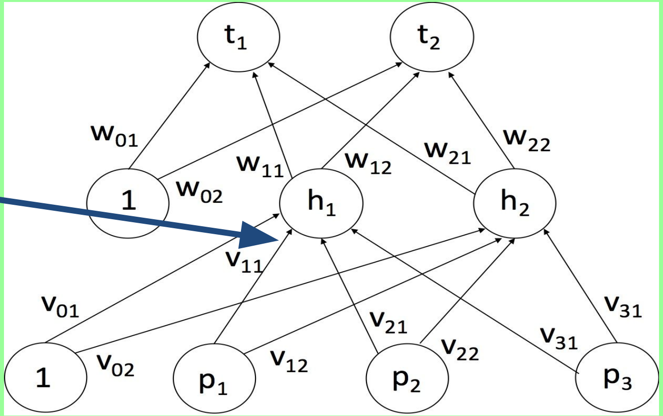


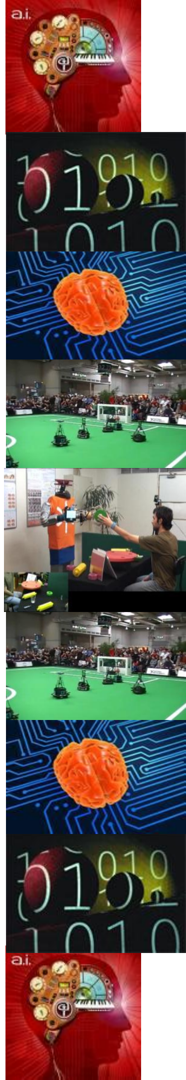
Backpropagation- Hidden Layer

$$\delta n_i = \sum_{k=1}^{m} \delta_k w_{ik}$$

$$\delta_i = \delta n_i f'(n_{1i})$$

$$\Delta v_{li} = \alpha \delta_i p_l$$





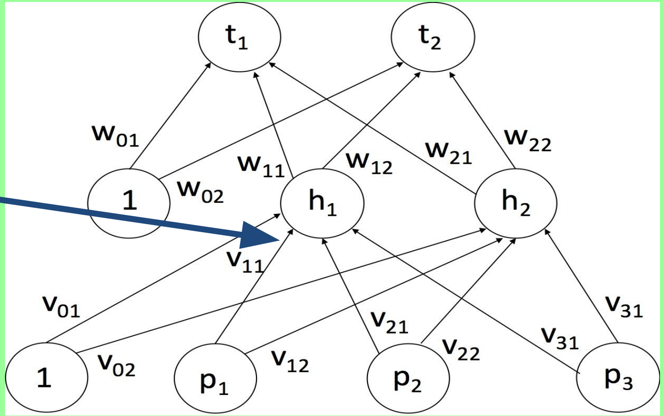
Backpropagation- Hidden Layer Sum

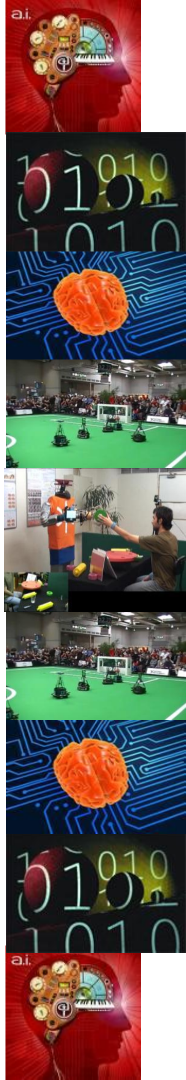
$$\delta n_i = \sum_{k=1}^{m} \delta_k w_{ik}$$

$$\delta_i = \delta n_i f'(n_{1i})$$

$$\Delta v_{li} = \alpha \delta_i p_l$$

$$v_{li}(new) = v_{li}(old) + \Delta v_{li}$$

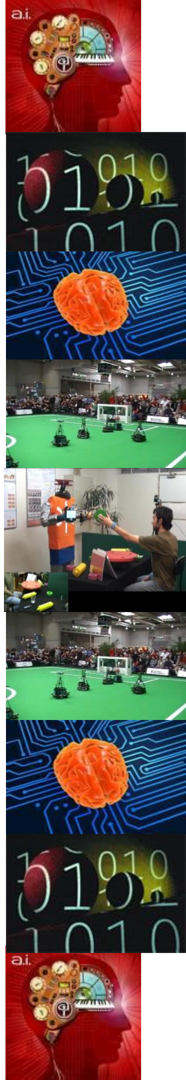




Update Weights and Bias

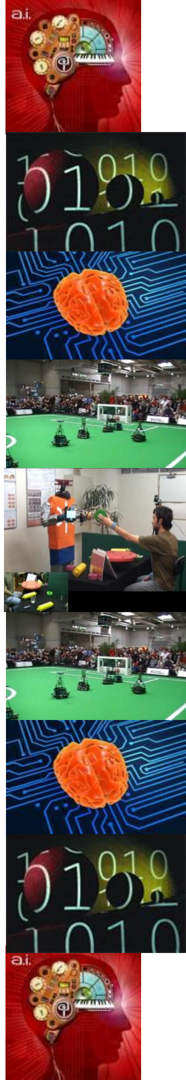
Algorithm 5 Backpropagation: Updating weights

1: Update the weights and the bias for the output layer
2: Update the weights and the bias for the hidden layer

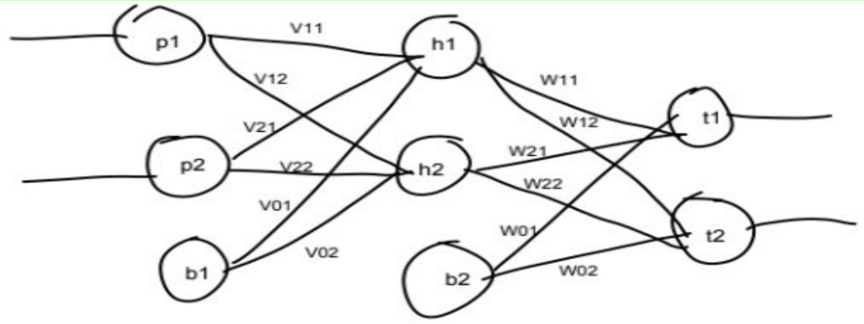


Testing the Neural Network

- Convergence (no error or increase)
- Test set
- Weighted sum of inputs
- Activation function



Example



Example of updating w_{11} in the output layer and V_{11} input layer

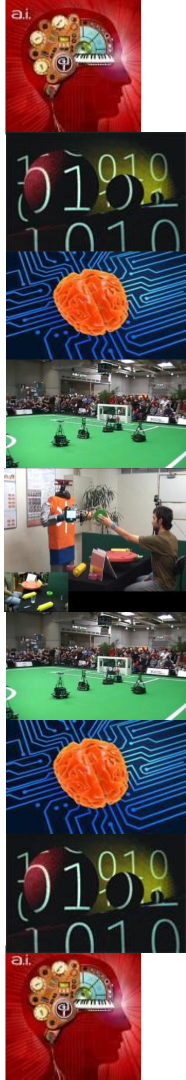
Assume we have following values

$p1=0.35$, $p2= 0.4$, $v_{11}= 0.2$, $v_{12}= 0.3$, $v_{21}= -0.3$, $v_{22}= 0.8$, $v_{o1}= 0.4$, $v_{02}= 0.7$

$w_{11}=0.7$, $w_{12}=0.25$, $w_{21}=0.25$, $w_{22}=0.6$, $w_{01}=0.1$, $w_{02}=0.3$,

Expected results $t_1 = 0.5$, $t_2 = 0.7$

Learning rate 0.8 , activation function $= 1/(1 + e^{-n})$



Example

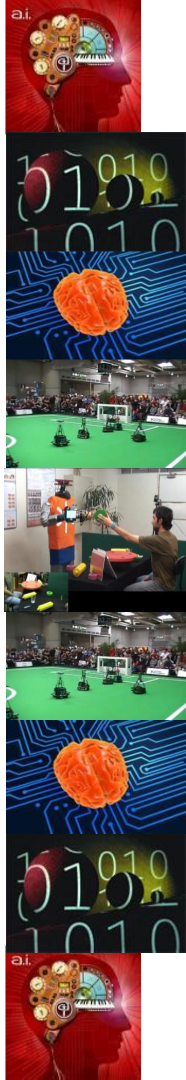
<u>**Step 1 Forward (learning) pass**</u>

$h_1 = p1(v_{11}) + p2(v_{21}) + \text{bias} = (0.35)(0.2) + 0.4(-0.3) + 1(0.4) = 0.35 = 1/(1+ e^{-(0.35)}) = 0.587$

$h_2 = p1(v_{12}) + p2(v_{22}) + \text{bias} = (0.35)(0.3) + 0.4(0.8) + 1(0.7) = 1.125 = 1/(1+ e^{-(1.125)}) = 0.755$

$t_1 = h1(w_{11}) + h2(w_{21}) + \text{bias} = (0.587)(0.7) + 0.755(0.25) + 1(0.1) = 0.7 = 1/(1+ e^{-(0.7)}) = \mathbf{0.67}$

$t_2 = h1(w_{12}) + h2(w_{22}) + \text{bias} = (0.587)(0.25) + 0.755(0.6) + 1(0.3) = 0.8997 = 1/(1+ e^{-(0.8997)}) = \mathbf{0.71}$



Example

<u>Step 2 Backpropagation</u>

Error for each output node given by

$\delta_k = (t_k - f\,(n_{2k}\,))f'(n_{2k}\,)$ { where f'(n) = f (n)(1 − f (n)) = derivative pg 15 notes eq 5} t_k - expected output

$\delta_1 = (0.5 - 0.67)((0.67)(1-0.67)) = -0.0376$ —- error for output node 1

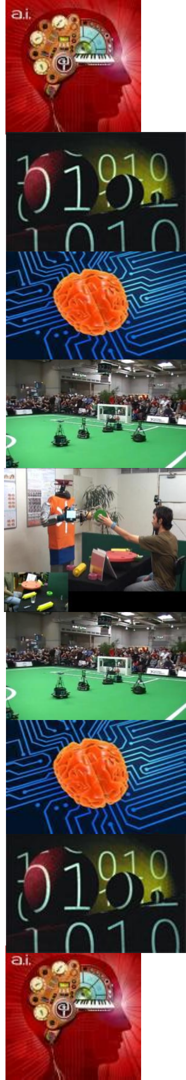
$\delta_2 = (0.7 - 0.71)((0.71)(1-0.71)) = -0.0020$ —- error for output node 2

Calculation for weight w_{11} update

$$\Delta w_{ik} = \alpha \delta_k f(n_{1i})$$

$\Delta w_{11} = 0.8 \text{x} -0.0376 \text{ x } (0.587) = -0.0177$

$W_{11} = w_{11} + \Delta w_{11} = 0.7 - 0.0177 = 0.682$



Example

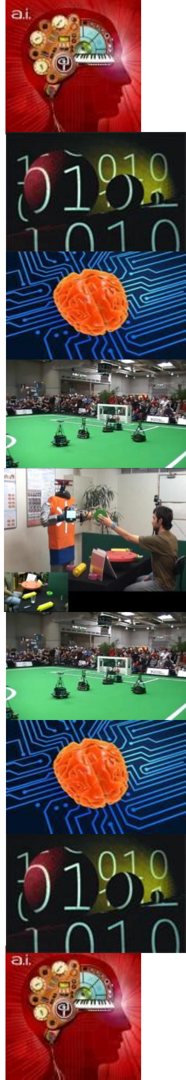
Hidden Layer Error Calculation

$$\delta n_i = \sum_{k=1}^{m} \delta_k w_{ik}$$

$\boldsymbol{\delta_{ni}}$ = (error from output node1 x weight to output node 1) + (error from output node2 x weight to output node2)

$\boldsymbol{\delta_{ni}}$ = (w_{11} * δ_1) + (w_{12} * δ_2) = (0.7*-0.0376) + (0.25*-0.0020)=-0.02682

(note use old value of w_{11})



Example

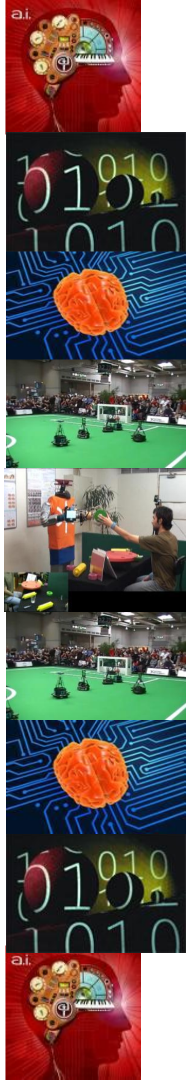
Where f'(n) = f (n)(1 − f (n))

δ_h - hidden layer node 1 error = -0.02682 x (0.67)(1-0.67) = -0.0059

$$\Delta v_{li} = \alpha \delta_i p_l$$

Δv_{11}= 0.8 x -0.0059 x 0.35 = -0.00166

$V_{11} = v_{11} + \Delta v_{11}$ = 0.2 -0.00166 = 0.198



Questions

Next Lecture - Hopfield Neural Network