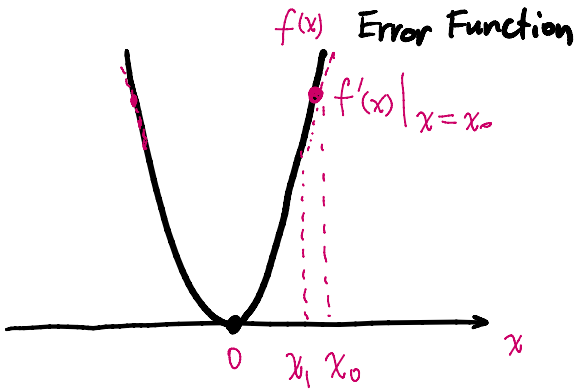
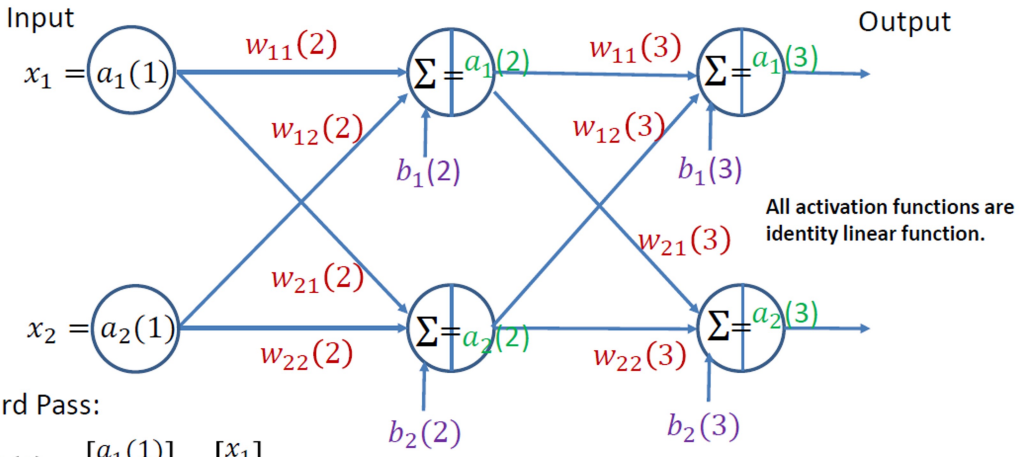
Lecture 25

Gradient Descent Algorithm :



$$x_{k+1} = x_k - \alpha \cdot f'(x)\big|_{x=x_k}$$

A Network with one Hidden Layer



Forward Pass:

$$\boldsymbol{A}(1) = \begin{bmatrix} a_1(1) \\ a_2(1) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{A}(2) = \begin{bmatrix} a_1(2) \\ a_2(2) \end{bmatrix} = \boldsymbol{W}(2)\boldsymbol{A}(1) + \boldsymbol{b}(2) = \begin{bmatrix} w_{11}(2) & w_{12}(2) \\ w_{21}(2) & w_{22}(2) \end{bmatrix} \begin{bmatrix} a_1(1) \\ a_2(1) \end{bmatrix} + \begin{bmatrix} b_1(2) \\ b_2(2) \end{bmatrix}$$

$$\boldsymbol{A}(3) = \begin{bmatrix} a_1(3) \\ a_2(3) \end{bmatrix} = \boldsymbol{W}(3)\boldsymbol{A}(2) + \boldsymbol{b}(3) = \begin{bmatrix} w_{11}(3) & w_{12}(3) \\ w_{21}(3) & w_{22}(3) \end{bmatrix} \begin{bmatrix} a_1(2) \\ a_2(2) \end{bmatrix} + \begin{bmatrix} b_1(3) \\ b_2(3) \end{bmatrix}$$

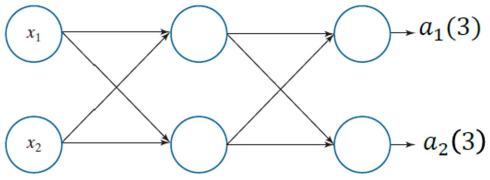
Loss Function for a Multilayer Neural Network

- Given a set of training patterns and a multilayer feedforward neural network architecture, we want to find the network parameters. that minimize an error (also called cost or objective) function.
- Our interest is in classification performance, so we define the error function for a neural network as the average of the differences between desired and actual responses.
- The activation values of neuron j in the output layer is $a_j(L)$. We define the error of that neuron as
  $E_j = \frac{1}{2}\left(r_j - a_j(L)\right)^2$, for $j = 1,2,\dots,n_L$.
- The output error with respect to a single $\mathbf{x}$ is the sum of the errors of all output neurons with respect to that vector (using the Euclidean vector norm):

$$E = \sum_{j=1}^{n_L} E_j = \frac{1}{2}\sum_{j=1}^{n_L}\left(r_j - a_j(L)\right)^2 = \frac{1}{2}\|\mathbf{r} - \mathbf{a}(L)\|^2$$

- The *total network output error* over all training patterns is defined as the sum of the errors of the individual patterns.

The output error with respect to a single $\mathbf{x}$



Desired response: $\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$

$$\boldsymbol{A}(3) = \begin{bmatrix} a_1(3) \\ a_2(3) \end{bmatrix} = \boldsymbol{W}(3)\boldsymbol{A}(2) + \boldsymbol{b}(3)$$

Error: $E = \frac{1}{2}\|\mathbf{r} - \boldsymbol{A}(3)\|^2 = \frac{1}{2}\{[r_1 - a_1(3)]^2 + [r_2 - a_2(3)]^2\}$

Let the derivatives of the output error with respect to the final output be:

$$\boldsymbol{D}(3) = \frac{\partial E}{\partial \boldsymbol{A}(3)} = \begin{bmatrix} \frac{\partial E}{\partial a_1(3)} \\ \frac{\partial E}{\partial a_2(3)} \end{bmatrix} = -\begin{bmatrix} r_1 - a_1(3) \\ r_2 - a_2(3) \end{bmatrix} = \boldsymbol{A}(3) - \mathbf{r}$$

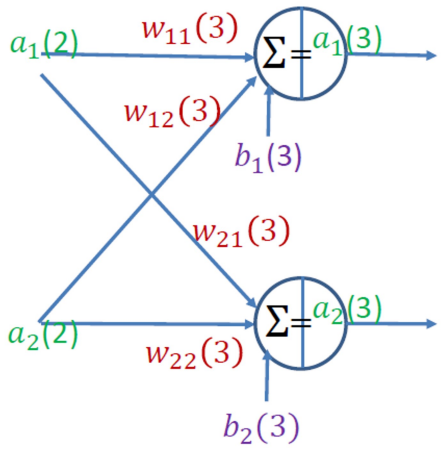
Gradient of the Error with respect to Weights

$$\frac{\partial E}{\partial w_{11}(3)} = \frac{\partial E}{\partial a_1(3)}\frac{\partial a_1(3)}{\partial w_{11}(3)} = \frac{\partial E}{\partial a_1(3)}a_1(2)$$

$$\frac{\partial E}{\partial w_{12}(3)} = \frac{\partial E}{\partial a_1(3)}\frac{\partial a_1(3)}{\partial w_{12}(3)} = \frac{\partial E}{\partial a_1(3)}a_2(2)$$

$$\frac{\partial E}{\partial w_{21}(3)} = \frac{\partial E}{\partial a_2(3)}\frac{\partial a_2(3)}{\partial w_{21}(3)} = \frac{\partial E}{\partial a_2(3)}a_1(2)$$

$$\frac{\partial E}{\partial w_{22}(3)} = \frac{\partial E}{\partial a_2(3)}\frac{\partial a_2(3)}{\partial w_{22}(3)} = \frac{\partial E}{\partial a_2(3)}a_2(2)$$



$$\frac{\partial E}{\partial \boldsymbol{W}(3)} = \begin{bmatrix} \frac{\partial E}{\partial w_{11}(3)} & \frac{\partial E}{\partial w_{12}(3)} \\ \frac{\partial E}{\partial w_{21}(3)} & \frac{\partial E}{\partial w_{22}(3)} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(3)} \\ \frac{\partial E}{\partial a_2(3)} \end{bmatrix} [a_1(2) \quad a_2(2)] = \frac{\partial E}{\partial \boldsymbol{A}(3)}\boldsymbol{A}(2)^{\mathrm{T}} = \boldsymbol{D}(3)\boldsymbol{A}(2)^{\mathrm{T}}$$

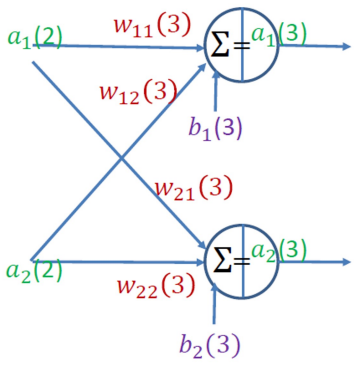
2X2 2X1 1X2

where $\boldsymbol{A}(3) = \begin{bmatrix} a_1(3) \\ a_2(3) \end{bmatrix} = \boldsymbol{W}(3)\boldsymbol{A}(2) + \boldsymbol{b}(3)$

Gradient of the Error with respect to Biases

$$\frac{\partial E}{\partial b_1(3)} = \frac{\partial E}{\partial a_1(3)}\frac{\partial a_1(3)}{\partial b_1(3)} = \frac{\partial E}{\partial a_1(3)}$$

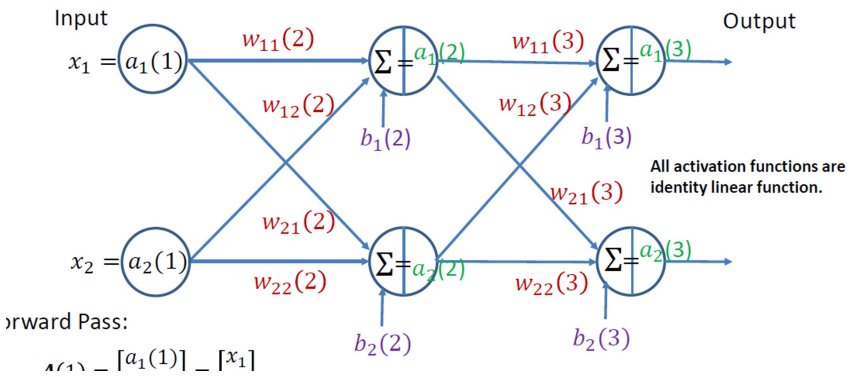
$$\frac{\partial E}{\partial b_2(3)} = \frac{\partial E}{\partial a_2(3)}\frac{\partial a_2(3)}{\partial b_2(3)} = \frac{\partial E}{\partial a_2(3)}$$



$$a_1(3) = w_{11}(3)a_1(2) + w_{12}(3)a_2(2) + b_1(3)$$

$$\frac{\partial E}{\partial b_1(3)} = 1$$

$$\frac{\partial E}{\partial \boldsymbol{b}(3)} = \begin{bmatrix} \frac{\partial E}{\partial b_1(3)} \\ \frac{\partial E}{\partial b_2(3)} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(3)} \\ \frac{\partial E}{\partial a_2(3)} \end{bmatrix} = \frac{\partial E}{\partial \boldsymbol{A}(3)} = \boldsymbol{D}(3)$$



rward Pass:

$A(1) = [a_1(1)] = [x_1]$

Relation between $\boldsymbol{D}(2)$ and $\boldsymbol{D}(3)$

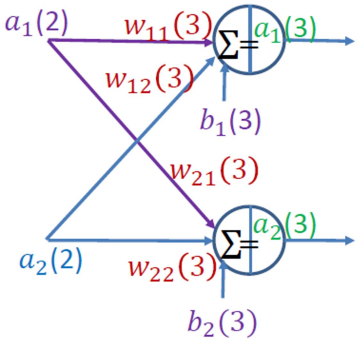
$$\boldsymbol{D}(2) = \frac{\partial E}{\partial \boldsymbol{A}(2)} = \begin{bmatrix} \frac{\partial E}{\partial a_1(2)} \\ \frac{\partial E}{\partial a_2(2)} \end{bmatrix}$$

$$\frac{\partial E}{\partial a_1(2)} = \frac{\partial E}{\partial a_1(3)}\frac{\partial a_1(3)}{\partial a_1(2)} + \frac{\partial E}{\partial a_2(3)}\frac{\partial a_2(3)}{\partial a_1(2)}$$

$$= \frac{\partial E}{\partial a_1(3)} w_{11}(3) + \frac{\partial E}{\partial a_2(3)} w_{21}(3)$$

$$\frac{\partial E}{\partial a_2(2)} = \frac{\partial E}{\partial a_1(3)}\frac{\partial a_1(3)}{\partial a_2(2)} + \frac{\partial E}{\partial a_2(3)}\frac{\partial a_2(3)}{\partial a_2(2)}$$

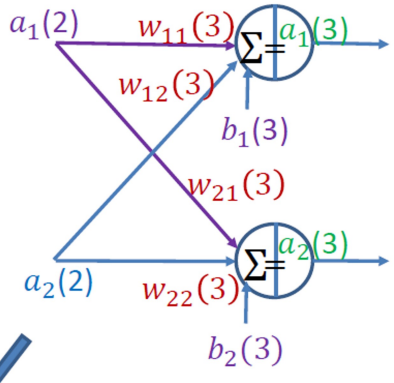
$$= \frac{\partial E}{\partial a_1(3)} w_{12}(3) + \frac{\partial E}{\partial a_2(3)} w_{22}(3)$$



$$\text{Thus } \boldsymbol{D}(2) = \begin{bmatrix} \frac{\partial E}{\partial a_1(2)} \\ \frac{\partial E}{\partial a_2(2)} \end{bmatrix} = \begin{bmatrix} w_{11}(3) & w_{12}(3) \\ w_{21}(3) & w_{22}(3) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{\partial E}{\partial a_1(3)} \\ \frac{\partial E}{\partial a_2(3)} \end{bmatrix} = \boldsymbol{W}(3)^{\mathrm{T}} \boldsymbol{D}(3)$$

$$\begin{bmatrix} W_{11}(3) & W_{21}(3) \\ W_{12}(3) & W_{22}(3) \end{bmatrix}$$

Backpropagation of $\boldsymbol{D}(3)$



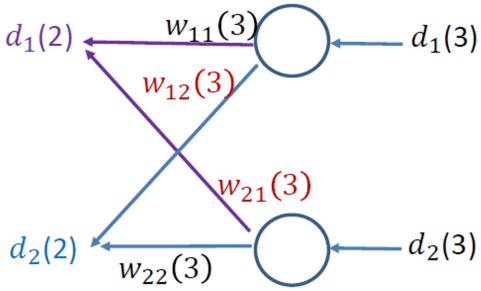
To calculate $\boldsymbol{D}(2)$, we back propagate $\boldsymbol{D}(3)$ as:

$$\boldsymbol{D}(2) = \begin{bmatrix} d_1(2) \\ d_2(2) \end{bmatrix} = \begin{bmatrix} w_{11}(3) & w_{21}(3) \\ w_{12}(3) & w_{22}(3) \end{bmatrix} \begin{bmatrix} d_1(3) \\ d_2(3) \end{bmatrix}$$

$$= \boldsymbol{W}(3)^{\mathrm{T}} \boldsymbol{D}(3)$$

Note the reversed directions of the arrows, thus the transpose of the weight matrix $\boldsymbol{W}(3)^{\mathrm{T}}$.

Note the reversed directions of the arrows, thus the transpose of the weight matrix $\boldsymbol{W}(3)^{\mathrm{T}}$.



$$\boldsymbol{D}(2) = \begin{bmatrix} d_1(2) \\ d_2(2) \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(2)} \\ \frac{\partial E}{\partial a_2(2)} \end{bmatrix}$$

$$\boldsymbol{D}(3) = \begin{bmatrix} d_1(3) \\ d_2(3) \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(3)} \\ \frac{\partial E}{\partial a_2(3)} \end{bmatrix}$$

Gradient of the Error with respect to Weights (Level Two)

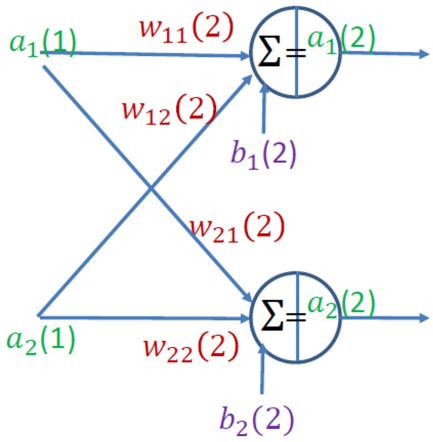
Similar to the previous derivations, for the 2nd layer:

$$\frac{\partial E}{\partial w_{11}(2)} = \frac{\partial E}{\partial a_1(2)}\frac{\partial a_1(2)}{\partial w_{11}(2)} = \frac{\partial E}{\partial a_1(2)}a_1(1)$$

$$\frac{\partial E}{\partial w_{12}(2)} = \frac{\partial E}{\partial a_1(2)}\frac{\partial a_1(2)}{\partial w_{12}(2)} = \frac{\partial E}{\partial a_1(2)}a_2(1)$$

$$\frac{\partial E}{\partial w_{21}(2)} = \frac{\partial E}{\partial a_2(2)}\frac{\partial a_2(2)}{\partial w_{21}(2)} = \frac{\partial E}{\partial a_2(2)}a_1(1)$$

$$\frac{\partial E}{\partial w_{22}(2)} = \frac{\partial E}{\partial a_2(2)}\frac{\partial a_2(2)}{\partial w_{22}(2)} = \frac{\partial E}{\partial a_2(2)}a_2(1)$$



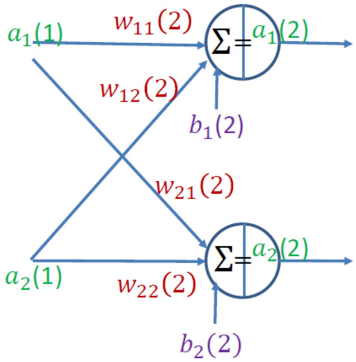
$$\frac{\partial E}{\partial \boldsymbol{W}(2)} = \begin{bmatrix} \frac{\partial E}{\partial w_{11}(2)} & \frac{\partial E}{\partial w_{12}(2)} \\ \frac{\partial E}{\partial w_{21}(2)} & \frac{\partial E}{\partial w_{22}(2)} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(2)} \\ \frac{\partial E}{\partial a_2(2)} \end{bmatrix} [a_1(1) \quad a_2(1)] = \frac{\partial E}{\partial \boldsymbol{A}(2)}\boldsymbol{A}(1)^{\mathrm{T}} = \boldsymbol{D}(2)\boldsymbol{A}(1)^{\mathrm{T}}$$

Where $\boldsymbol{D}(2)$ is obtained by back propagating $\boldsymbol{D}(3)$, and $\boldsymbol{A}(1) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is the input vector.

Gradient with respect to Biases (2nd Layer)

$$\frac{\partial E}{\partial b_1(2)} = \frac{\partial E}{\partial a_1(2)}\frac{\partial a_1(2)}{\partial b_1(2)} = \frac{\partial E}{\partial a_1(2)}$$

$$\frac{\partial E}{\partial b_2(2)} = \frac{\partial E}{\partial a_2(2)}\frac{\partial a_2(2)}{\partial b_2(2)} = \frac{\partial E}{\partial a_2(2)}$$



$$\frac{\partial E}{\partial \boldsymbol{b}(2)} = \begin{bmatrix} \frac{\partial E}{\partial b_1(2)} \\ \frac{\partial E}{\partial b_2(2)} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial a_1(2)} \\ \frac{\partial E}{\partial a_2(2)} \end{bmatrix} = \frac{\partial E}{\partial \boldsymbol{A}(2)} = \boldsymbol{D}(2)$$

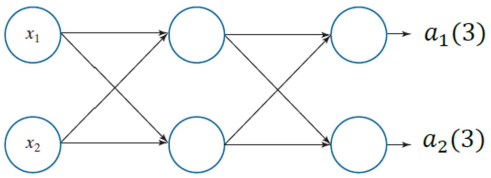
Summary of the Results

$$\boldsymbol{A}(1) = \begin{bmatrix} a_1(1) \\ a_2(1) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{A}(2) = \begin{bmatrix} a_1(2) \\ a_2(2) \end{bmatrix} = \boldsymbol{W}(2)\boldsymbol{A}(1) + \boldsymbol{b}(2)$$

$$\boldsymbol{A}(3) = \begin{bmatrix} a_1(3) \\ a_2(3) \end{bmatrix} = \boldsymbol{W}(3)\boldsymbol{A}(2) + \boldsymbol{b}(3)$$

Forward Pass



Error: $E = \frac{1}{2}\|\mathbf{r} - \boldsymbol{A}(3)\|^2$

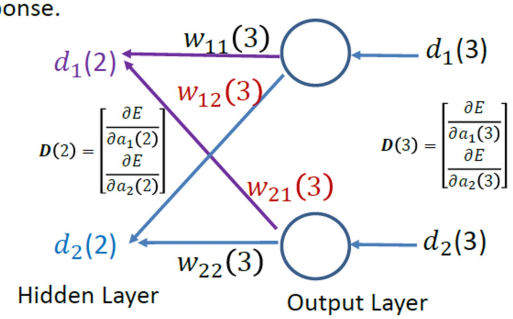
$\boldsymbol{D}(3) = \boldsymbol{A}(3) - \mathbf{r}$, where $\mathbf{r}$ is the desired response.

Backpropagation of error gradient from output to hidden layer:

Backpropagation: $\boldsymbol{D}(2) = \boldsymbol{W}(3)^{\mathrm{T}}\boldsymbol{D}(3)$

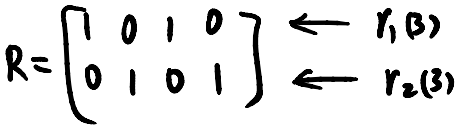
$$\frac{\partial E}{\partial \boldsymbol{W}(3)} = \boldsymbol{D}(3)\boldsymbol{A}(2)^{\mathrm{T}}, \quad \frac{\partial E}{\partial \boldsymbol{b}(3)} = \boldsymbol{D}(3)$$

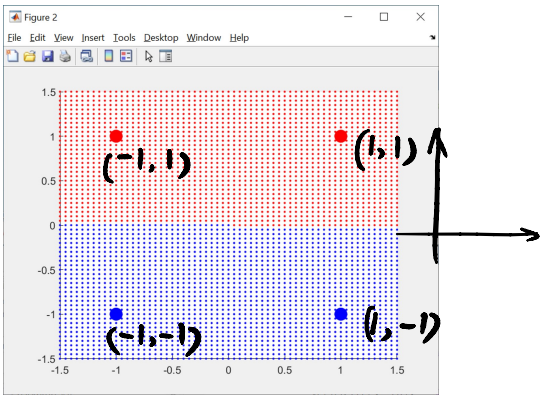
$$\frac{\partial E}{\partial \boldsymbol{W}(2)} = \boldsymbol{D}(2)\boldsymbol{A}(1)^{\mathrm{T}}, \quad \frac{\partial E}{\partial \boldsymbol{b}(2)} = \boldsymbol{D}(2)$$




```
epoch = 0;
while (epoch <= max_iter)
   epoch = epoch + 1;

   for i = 1: 4
      A1 = X(:,i);
      A2 = W2*A1 + b2;
      A3 = W3*A2 + b3;

      D3 = A3 - R(:,i);

      mse(epoch) = 0.5*norm(D3)^2;

      % backpropagation
      D2 = W3'*D3;

      % Update the weights and biases
      W3 = W3 - alpha*D3*A2';
      W2 = W2 - alpha*D2*A1';

      b3 = b3 - alpha*D3;
      b2 = b2 - alpha*D2;
   end

end
mse(epoch)
plot(mse); grid
```

```
% backprop.m
% Explain the backpropagation algorithm using a fully connected neural
% network with one hidden layer.
% However, the activation of each neuron is a linear function, thus the
% network output is a linear combination of the input. Therefore, this
% network cannot handle linearly non-separable cases.
% The weights and biases are updated for each input sample

alpha = 0.1;  % learning rate

% Linearly separable example
% Input data pattern
X = [1 -1 -1 1; 1 -1 1 -1];
% Response
R = [1  0  1 0; 0  1 0  1];
```






```
rng('default');
Std = 0.02;

% Initial weights and biases
W2 = Std*randn(2,2);
b2 = Std*randn(2,1);

W3 = Std*randn(2,2);
b3 = Std*randn(2,1);

max_iter = 100;
mse = zeros(1, max_iter);
```