Mult-layer perceptron

Back Propagation

Steps of a single epoch

For each pattern

- Forward prop
  - Calculate net_j and o_j for all neurons (except input layer and bias neurons)
  - Calculate specific error (for single pattern)
- Back prop
  - Calculate δ_j for all neurons (except input layer and bias neurons)
  - Calculate $\Delta w_{i,j}$ for all variable weights including bias weights
  - $w_{i,j} \coloneqq w_{i,j} + \Delta w_{i,j}$

After end of epoch

- Calculate total error = sum of specific errors
- Check stopping condition
- Run another epoch if stopping condition is False

Notes

Assuming layer K is before layer H, and either layer L is after layer H, or layer H is the output layer

- $net_h = \sum_{k \in K} w_{k,h}\, o_k$
- $o_h = f_{act}(net_h)$ Default is $\frac{1}{1+e^{-net}}$
- $\delta_h =$
  - $f'_{act}(net_h) \cdot \sum_{l \in L} \delta_l\, w_{h,l}$ if h is hidden neuron
    - Default is: $o_h(1-o_h) \sum_{l \in L} \delta_l\, w_{h,l}$ for sigmoid activation function
  - $f'_{act}(net_h) \cdot -\frac{\partial Err_p}{\partial y_h}$ if h is output neuron
    - Default is: $y_h(1-y_h)(t_h - y_h)$ for sigmoid activation function and for $Err_p = \frac{1}{2}\sum_{h \in H}(t_h - y_h)^2$

  Warning! if different activation function or different error function is used, you must calculate the derivatives $f'_{act}(net_h) \cdot -\frac{\partial Err_p}{\partial y_h}$

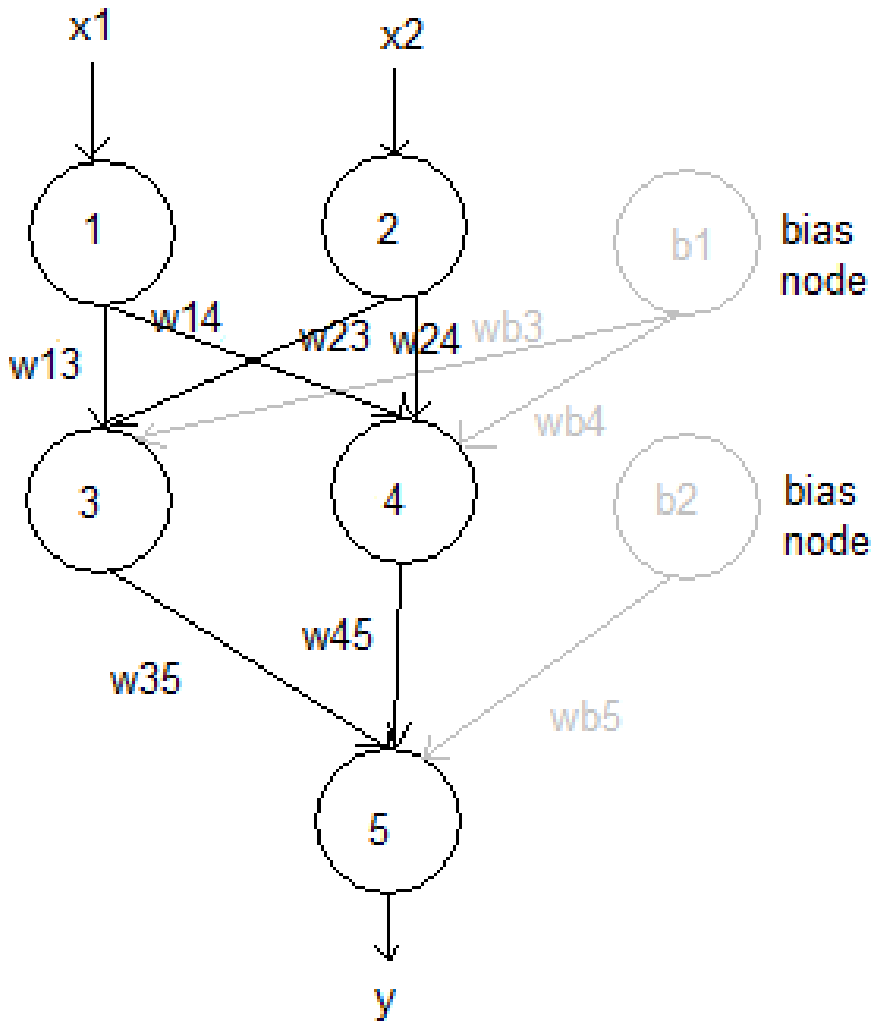
Notes

- $\Delta w_{i,j} = \eta\ o_i\ \delta_j$
- $w_{i,j} := w_{i,j} + \eta\ o_i\ \delta_j$
  - If i is input neuron: $o_i = x_i$
  - If i is bias neuron: $o_i = 1$

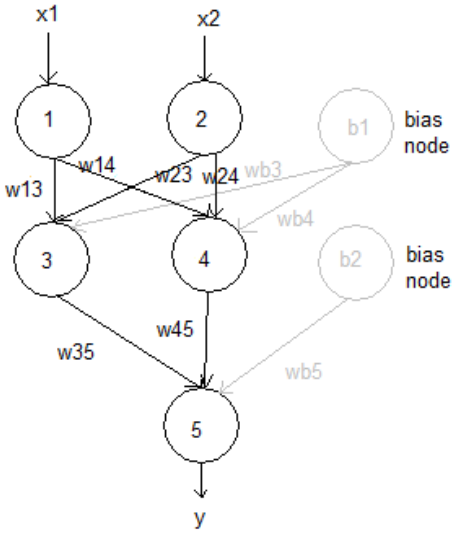
Example

x1	x2	t
0	0	0
0	1	1
1	0	1
1	1	0

Assume learning rate = 0.3



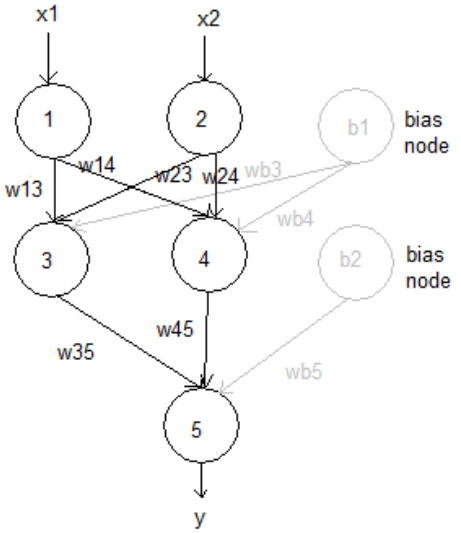
Epoch: 1
Pattern: 1: x1 = 0, x2 = 0, t = 0



- Initial weights:

w13 = 0.3	w23 = -0.1	wb3 = 0.2
w14 = -0.2	w24 = 0.2	wb4 = -0.3
w35 = 0.4	w45 = -0.2	wb5 = 0.4

- Forward prop:
  - net3 = w13 * x1 + w23 * x2 + wb3 = 0.3 * 0 – 0.1 * 0 + 0.2 = 0.2
  - o3 = 1/(1 + e^-net3) = 1/(1 + e^-0.2) = 0.5498
  - net4 = w14 * x1 + w24 * x2 + wb4 = -0.2 * 0 + 0.2 * 0 – 0.3 = -0.3
  - o4 = 1/(1 + e^-net4) = 1/(1 + e^0.3) = 0.4256
  - net5 = w35 * o3 + w45 * o4 + wb5 = 0.4 * 0.5498 – 0.2 * 0.4256 + 0.4 = 0.5348
  - y = 1/(1+e^-net5) = 1/(1+e^-0.5348) = 0.6306
- Calculating error:
  - Err_p1 = 0.5 * (0 – 0.6306)^2 = 0.1988



Epoch: 1
Pattern: 1: x1 = 0, x2 = 0, t = 0

Back Prop:

1) Finding delta

$\delta 5 = y*(1 - y)*(t - y) = 0.6306 * (1 - 0.6306) * (0 - 0.6306) = - 0.1469$

$\delta 3 = o3(1 – o3)* \delta 5 * w35 = 0.5498 * (1 - 0.5498) * - 0.1469 * 0.4 = - 0.0145$

$\delta 4 = o4(1 – o4)* \delta 5 * w45 = 0.4256 * (1 - 0.4256) * - 0.1469 * -0.2 = 0.0072$

2) Finding new weights

$w35 := w35 + \eta * o3 * \delta 5 = 0.4 + 0.3 * 0.5498 * - 0.1469 = 0.3758$

$w45 := w45 + \eta * o4 * \delta 5 = -0.2 + 0.3 * 0.4256 * - 0.1469 = -0.2188$

$wb5 := wb5 + \eta * 1 * \delta 5 = 0.4 + 0.3 * 1 * - 0.1469 = 0.3559$

$w14 := w14 + \eta * x1 * \delta 4 = - 0.2 + 0.3 * 0 * 0.0072 = - 0.2$

$w24 := w24 + \eta * x2 * \delta 4 = 0.2 + 0.3 * 0 * 0.0072 = 0.2$

$wb4 := wb4 + \eta * 1 * \delta 4 = - 0.3 + 0.3 * 1 * 0.0072 = - 0.2978$

$w13 := w13 + \eta * x1 * \delta 3 = 0.3 + 0.3 * 0 * - 0.0145 = 0.3$

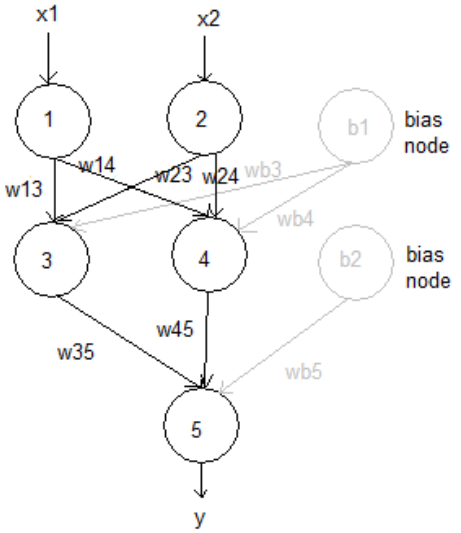
$w23 := w23 + \eta * x2 * \delta 3 = - 0.1 + 0.3 * 0 * - 0.0145 = - 0.1$

$wb3 := wb3 + \eta * 1 * \delta 3 = 0.2 + 0.3 * 1 * - 0.0145 = 0.1957$

Epoch: 1
Pattern: 2: x1 = 0, x2 = 1, t = 1

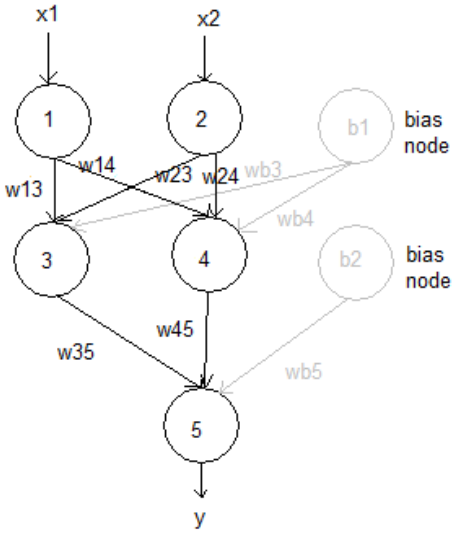
- weights:

w13 = 0.3	w23 = - 0.1	wb3 = 0.1957
w14 = - 0.2	w24 = 0.2	wb4 = - 0.2978
w35 = 0.3758	w45 = - 0.2188	wb5 = 0.3559



- Forward prop:
  - net3 = w13 * x1 + w23 * x2 + wb3 = …
  - o3 = 1/(1 + e^-net3) = …
  - net4 = w14 * x1 + w24 * x2 + wb4 = …
  - o4 = 1/(1 + e^-net4) = …
  - net5 = w35 * o3 + w45 * o4 + wb5 = …
  - y = 1/(1+e^-net5) = …
- Calculating error:
  - Err_p2 = 0.5 * (t – y)^2 = …

Epoch: 1
Pattern: 2: x1 = 0, x2 = 1, t = 1



Back Prop:

1) Finding delta

$\delta 5 = y*(1 - y)*(t - y) = \ldots$

$\delta 3 = o3(1 - o3)* \delta 5 * w35 = \ldots$

$\delta 4 = o4(1 - o4)* \delta 5 * w45 = \ldots$

2) Finding new weights

$w35 := w35 + \eta * o3 * \delta 5 = \ldots$

$w45 := w45 + \eta * o4 * \delta 5 = \ldots$

$wb5 := wb5 + \eta * 1 * \delta 5 = \ldots$

$w14 := w14 + \eta * x1 * \delta 4 = \ldots$

$w24 := w24 + \eta * x2 * \delta 4 = \ldots$

$wb4 := wb4 + \eta * 1 * \delta 4 = \ldots$

$w13 := w13 + \eta * x1 * \delta 3 = \ldots$

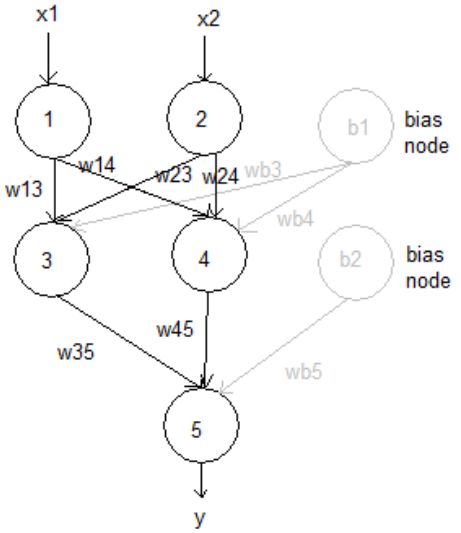
$w23 := w23 + \eta * x2 * \delta 3 = \ldots$

$wb3 := wb3 + \eta * 1 * \delta 3 = \ldots$

Epoch: 1
Pattern: 3: x1 = 1, x2 = 0, t = 1

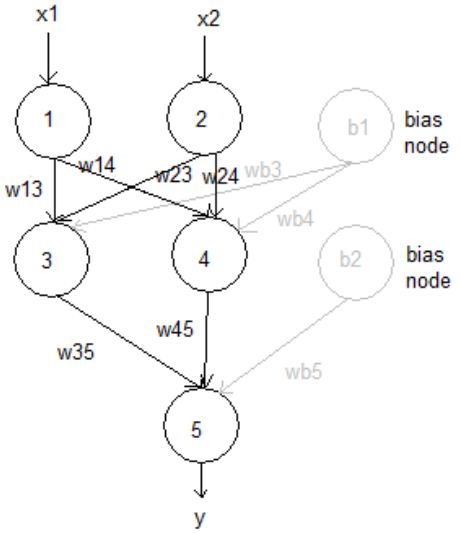
- weights:

w13 = ?	w23 = ?	wb3 = ?
w14 = ?	w24 = ?	wb4 = ?
w35 = ?	w45 = ?	wb5 = ?



- Forward prop:
  - net3 = w13 * x1 + w23 * x2 + wb3 = …
  - o3 = 1/(1 + e^-net3) = …
  - net4 = w14 * x1 + w24 * x2 + wb4 = …
  - o4 = 1/(1 + e^-net4) = …
  - net5 = w35 * o3 + w45 * o4 + wb5 = …
  - y = 1/(1+e^-net5) = …
- Calculating error:
  - Err_p3 = 0.5 * (t – y)^2 = …

Epoch: 1
Pattern: 3: x1 = 1, x2 = 0, t = 1



Back Prop:

1) Finding delta

$\delta 5 = y*(1 - y)*(t - y) = \ldots$

$\delta 3 = o3(1 - o3)* \delta 5 * w35 = \ldots$

$\delta 4 = o4(1 - o4)* \delta 5 * w45 = \ldots$

2) Finding new weights

$w35 := w35 + \eta * o3 * \delta 5 = \ldots$

$w45 := w45 + \eta * o4 * \delta 5 = \ldots$

$wb5 := wb5 + \eta * 1 * \delta 5 = \ldots$

$w14 := w14 + \eta * x1 * \delta 4 = \ldots$

$w24 := w24 + \eta * x2 * \delta 4 = \ldots$

$wb4 := wb4 + \eta * 1 * \delta 4 = \ldots$

$w13 := w13 + \eta * x1 * \delta 3 = \ldots$

$w23 := w23 + \eta * x2 * \delta 3 = \ldots$

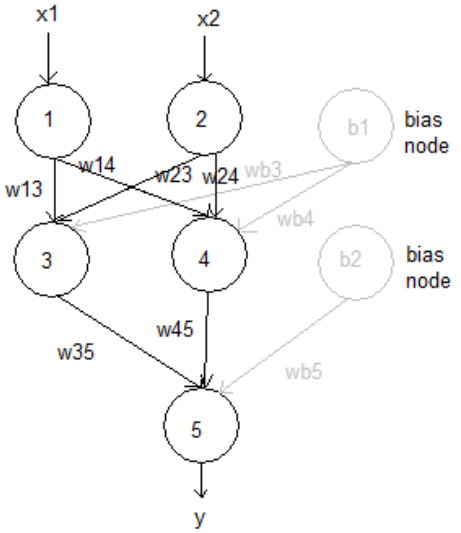
$wb3 := wb3 + \eta * 1 * \delta 3 = \ldots$

Epoch: 1
Pattern: 4: x1 = 1, x2 = 1, t = 0

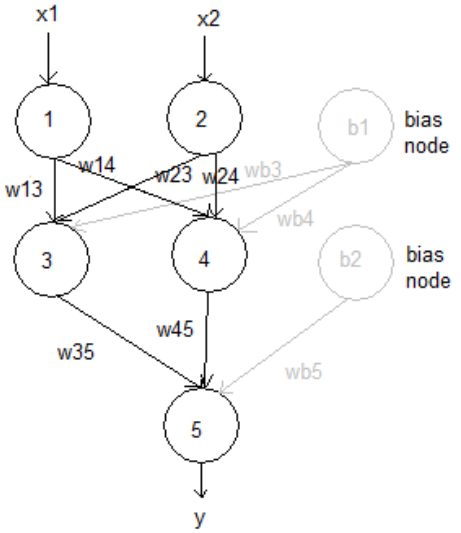
- weights:

w13 = ?	w23 = ?	wb3 = ?
w14 = ?	w24 = ?	wb4 = ?
w35 = ?	w45 = ?	wb5 = ?

- Forward prop:
  - net3 = w13 * x1 + w23 * x2 + wb3 = …
  - o3 = 1/(1 + e^-net3) = …
  - net4 = w14 * x1 + w24 * x2 + wb4 = …
  - o4 = 1/(1 + e^-net4) = …
  - net5 = w35 * o3 + w45 * o4 + wb5 = …
  - y = 1/(1+e^-net5) = …
- Calculating error:
  - Err_p4 = 0.5 * (t – y)^2 = …



Epoch: 1
Pattern: 4: x1 = 1, x2 = 1, t = 0



Back Prop:

1) Finding delta

$\delta 5 = y*(1 - y)*(t - y) = \ldots$

$\delta 3 = o3(1 - o3)* \delta 5 * w35 = \ldots$

$\delta 4 = o4(1 - o4)* \delta 5 * w45 = \ldots$

2) Finding new weights

$w35 := w35 + \eta * o3 * \delta 5 = \ldots$

$w45 := w45 + \eta * o4 * \delta 5 = \ldots$

$wb5 := wb5 + \eta * 1 * \delta 5 = \ldots$

$w14 := w14 + \eta * x1 * \delta 4 = \ldots$

$w24 := w24 + \eta * x2 * \delta 4 = \ldots$

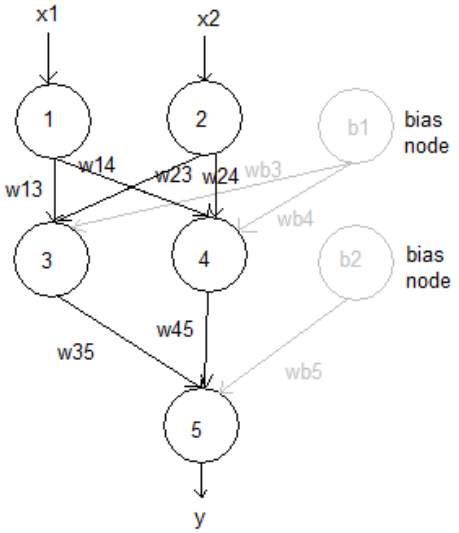
$wb4 := wb4 + \eta * 1 * \delta 4 = \ldots$

$w13 := w13 + \eta * x1 * \delta 3 = \ldots$

$w23 := w23 + \eta * x2 * \delta 3 = \ldots$

$wb3 := wb3 + \eta * 1 * \delta 3 = \ldots$

End of Epoch 1



Total error = Err_p1 + Err_p2 + Err_p3 + Err_p4 = 0.1988 + …

If Total error <= tolerance (If given): Then stop training

If epoch number = max number of epochs (if given): Then stop training

Otherwise, run another epoch using last weights