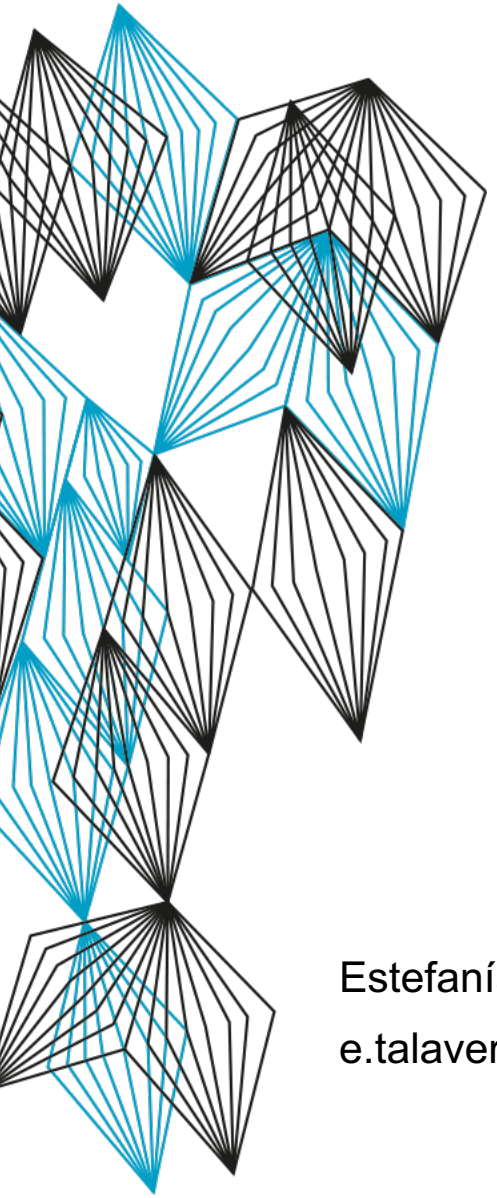


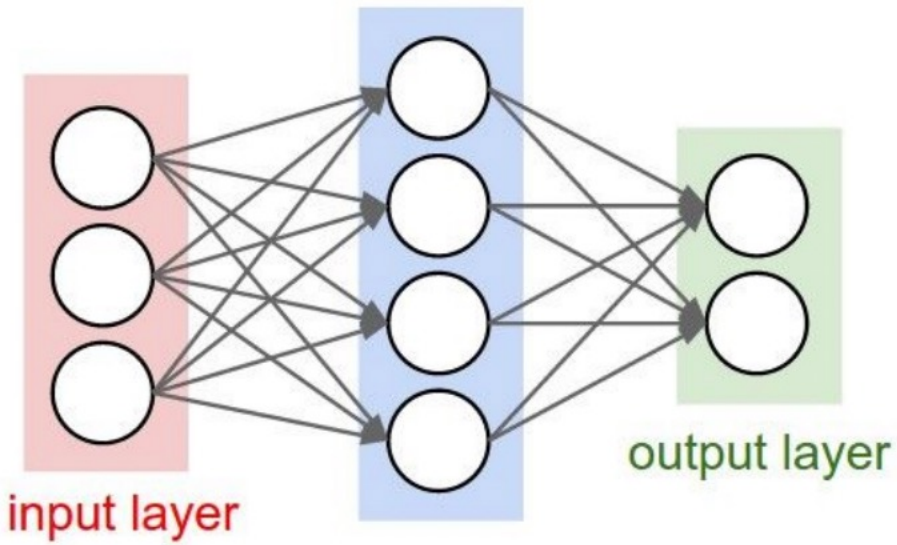
ARTIFICIAL INTELLIGENCE & CYBER SECURITY

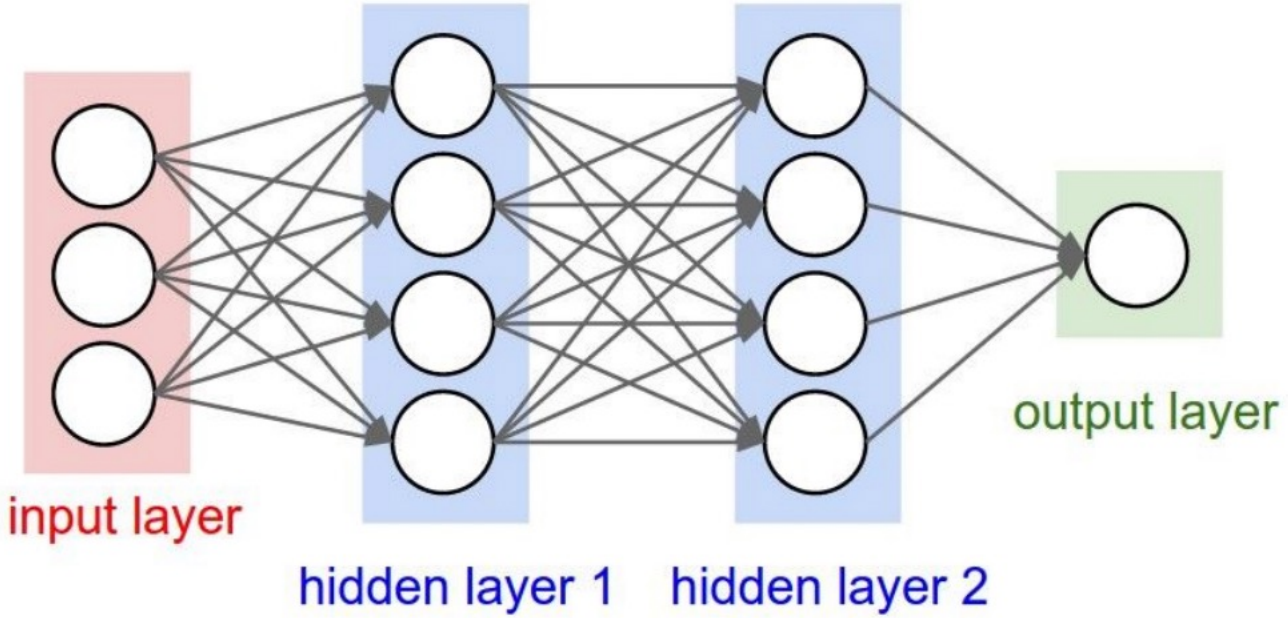
NEURAL NETWORKS

Feed Forward and Error Backpropagation

Estefanía Talavera Martínez
e.talaveramartinez@utwente.nl

Neural Networks Architectures





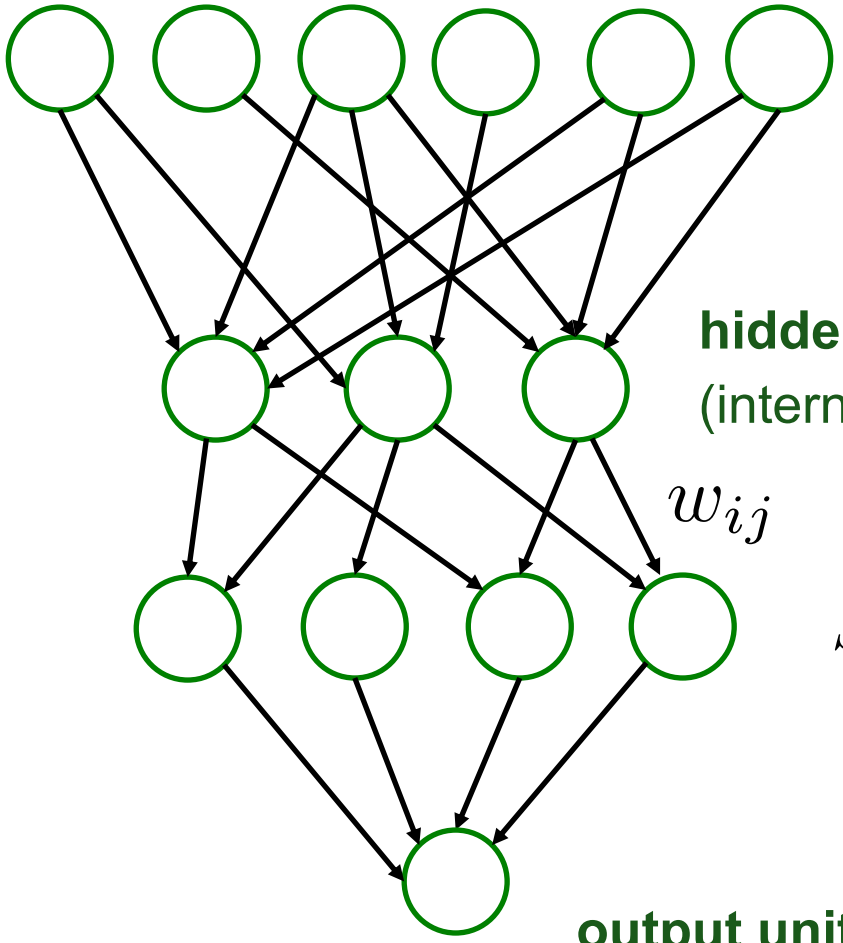
"2-layer Neural Net", or
"1-hidden-layer Neural Net"

"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

Feed-forward networks

input layer (external stimulus)



layered architecture
(here: 6-3-4-1)

directed connections
(here: only to next layer)

hidden units
(internal representation)

w_{ij}

$$S_i = g\left(\sum_j w_{ij} S_j\right)$$

↑ previous layer only

output unit(s)
(function of input vector)

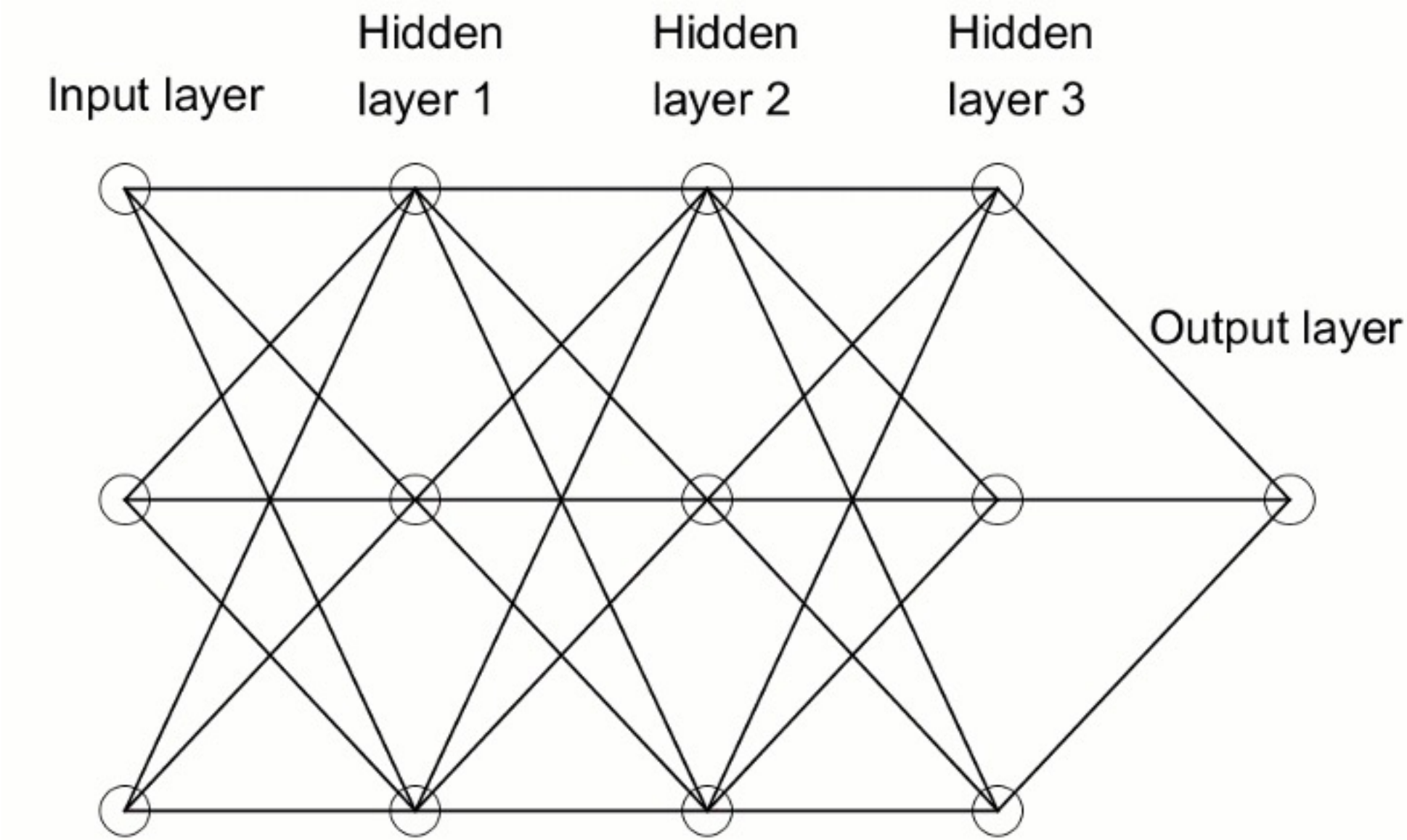
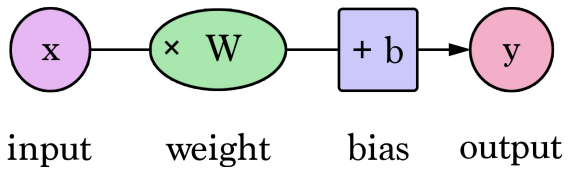


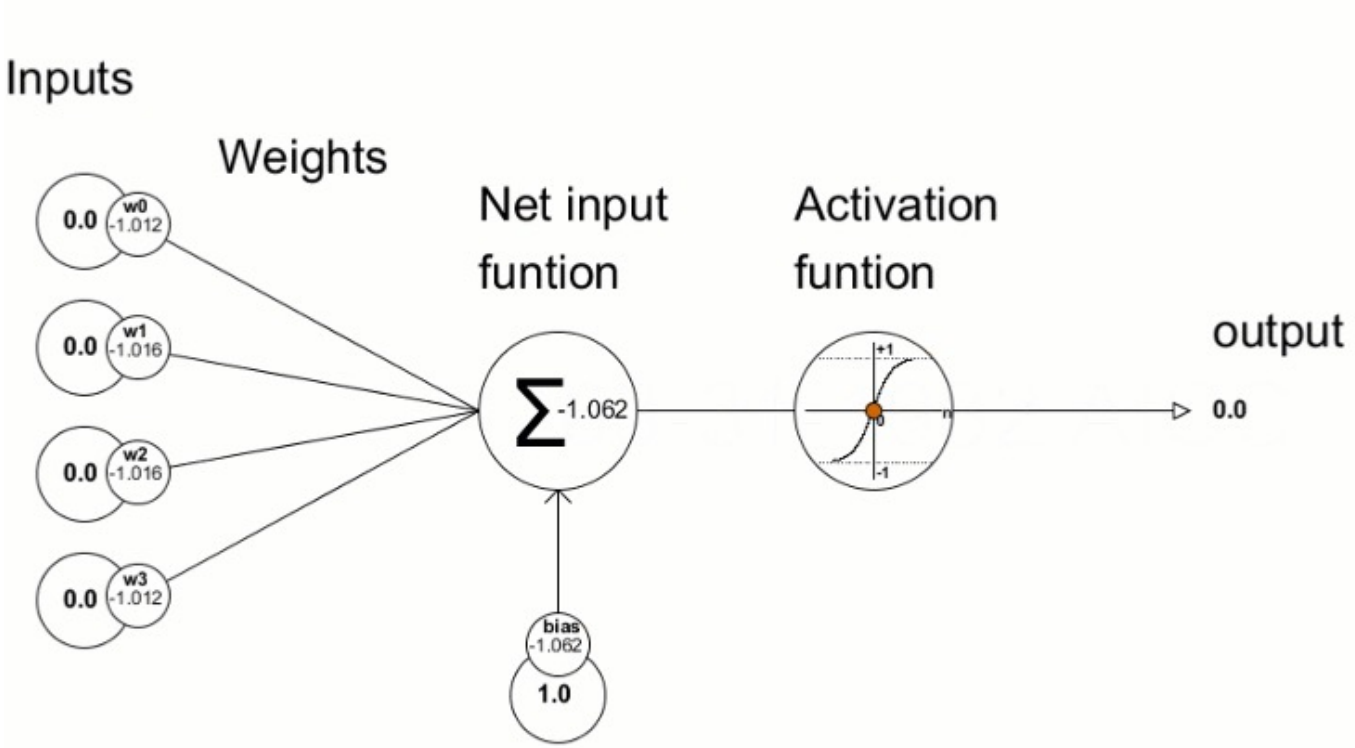
Figure: https://www.mql5.com/en/blogs/post/724245

Bias / Bias units / Bias nodes

What is bias?
Bias per neuron basis
The values are updated during training
Role similar to threshold – determines the activation function of the network

How is bias implemented?
g(w*x + b)



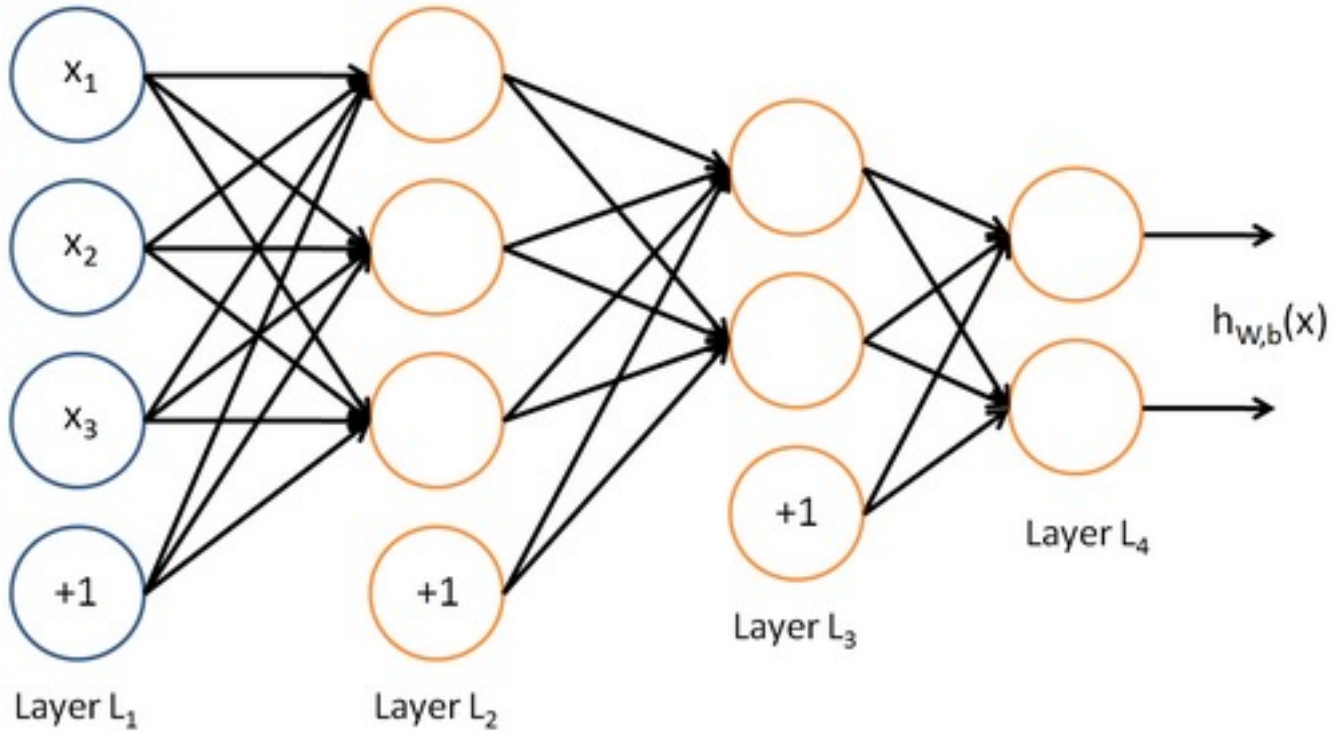


https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network

Figure: https://www.mql5.com/en/blogs/post/724245

Bias / Bias units / Bias nodes

How is bias implemented?
g(w*x + b)

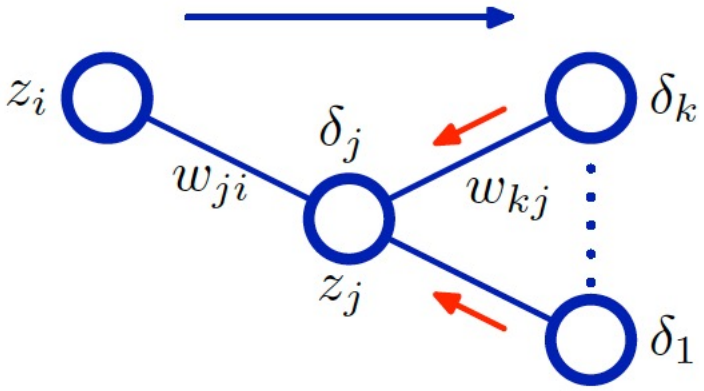


Backpropagation

Backward: apply the chain rule to compute the gradient of the loss function with respect to the inputs

Backpropagation = recursive application of the chain rule along a computational graph to compute the gradients of all inputs/parameters/intermediates

ERROR BACKPROPAGATION



Error Backpropagation

1. Feed forward Forward propagate an input vector x_n to find the activations for the hidden units.

2. Evaluate δ_k for all output units – calculate error.

3. Back propagate error Backpropagate the δ_k to obtain δj for all hidden units in all hidden layers.

4. Use the chain rule to find the derivatives with respect to the weights.