From my understand of back prop , it is essentially an application of chain rule in calculus. By calculating the derivatives of total error with respective to the weights we can use gradient descent to perform updates on them. The derivations on slide 19-21 also present a derivations to express the update formula from a calculus perspective. Below is my pseudo code.

X[j]: an array of a single example with size(2,1)

Y[j] : desired output for X[j]. of size(1,1)

W1 : weights from input layer to hidden layer , array of size (4,2)

b1 : bias for hidden layer of size (4,1). One for each neuron.

Z1 : the weighted sum in the hidden layer of size (4,1). Z1 = W1.dot(X1)+b1

A1 : the activated weighted sum for hidden layer of size(4,1). A1 = sigmoid(Z1)

W2 : weights from hidden layer to output layer of size (1,4)

b2 : bias for output neuron of size (1,1). Since there is only one output neuron.

Z2 : weighted sum for output layer . Z2 = W2.dot(A2) + b2 of size (1,1)

A2 : activated weighted sum for output layer of size(1,1)

/\*epochs\*/

For i in range(1000):

Cost = 0

/\* total of four training examples\*/

For j in range(4):

/\* forward pass\*/

Z1 = W1.dot(X[j]) + b1

A1 = sigmoid(Z1)

Z2 = W2.dot(A1)+b2

A2 = sigmoid(Z2) /\* A2 is the actual output generated from neuron network\*/

Cost += (Y[j]-A2)^2

/\* backward pass\*/

dZ2 = (A2-Y[j])\*A2\*(1-A2) /\* A2\*(1-A2) is the sigmoid derivative of A2\*/

dW2= dZ2.dot(A1.T) /\* A1.T is the transpose of A1. The operation is dot product of 2 arrays with size(1,1) and (1,4) respectively . The resulting dW2 is size(1,4) which match the size of W2

db2 = dZ2

dZ1 = W2.T .dot(dZ2)\*A1\*(1-A1) /\*size check : (4,1).dot(1,1) = (4,1)\*/

dW1 = dZ1 .dot(X[j].T) /\* X[j].T has size (1,2) , (4,1).dot(1,2) = (4,2) which match the size of W1

db1 = dZ1

/\* update weights using sgd\*/

W1 = W1 – learning rate\*dW1

b1 = b1 – learning rate\*db1

W2 = W2 – learning rate\*dW2

b2 = b2 – learning rate\*db2

cost = cost \*0.5 /\* total error = 0.5 \*sum(Y[j]-A2) for all training examples