**MIS 637 Assignment no 4**

1. Adjust the weights *W*0*B, W*1*B, W*2*B,* and *W*3*B*from the example of back-propagation in the text?

**Answer:**

**Data Inputs and Initial Values for Neural Network Weights:**

X0 = 1.0; X1 = 0.4; X2 = 0.2; X3 = 0.7

W0A = 0.5; W1A = 0.6; W2A = 0.8; W3A = 0.6

W0B = 0.7; W1B = 0.9; W2B = 0.8; W3B = 0.4

W0Z = 0.5; WAZ = 0.9; WBZ = 0.9

***δZ = outputZ (1 − outputZ )(actualZ − outputZ )***

First pass output= 0.8750

Actual target value = 0.8

Learning rate = 0.1

Prediction error = 0.8 - 0.8750 = -0.075

*δZ* = 0.875(1 − 0.875)(0.8 − 0.875) = −0.0082

Output B = 0.8176

**1. W0B**

δB = 0.8176(1 −0.8176) (0.9) (−0.0082) = −0.0011

∆W0B = n\* δB \*X0 = 0.1 (-0.0011) (1) = -0.00011

W0B.new = W0B.current + ∆W0B = 0.7 – 0.00011 = 0.69989

**2. W1B**

δB = 0.8176(1 −0.8176) (0.9) (−0.0082) = −0.0011

∆W1B = n \*δB\* X1 = 0.1 (-0.0011) (0.4) = -0.000044

W1B.new = W1B.current + ∆ W1B = 0.9 - 0.000044 = 0.899956

**3. W2B**

δB = 0.8176(1 −0.8176) (0.9) (−0.0082) = −0.0011

∆W2B = n\* δB\* X2 = 0.1 (-0.0011) (0.2) = -0.000022

W2B.new = W2B.current + ∆W2B = 0.8 - 0.000022 = 0.799978

**4. W3B**

δB = 0.8176(1 −0.8176) (0.9) (−0.0082) = −0.0011

∆W3B = n\* δB\* X3 = 0.1 (-0.0011) (0.7) = -0.000077

W3B.new = W3B.current + ∆W3B = 0.4 – 0.000077 = 0.399923

1. Refer to the previous problem. Show that the adjusted weights result in a smaller prediction error?

**Answer:**

**Data Inputs and Initial Values for Neural Network Weights:**

X0 = 1.0; X1 = 0.4; X2 = 0.2; X3 = 0.7

W0A = 0.5; W1A = 0.6; W2A = 0.8; W3A = 0.6

W0B = 0.7; W1B = 0.9; W2B = 0.8; W3B = 0.4

W0Z = 0.5; WAZ = 0.9; WBZ = 0.9

**Input layer Hidden layer Output layer**

**X1 W1A W0A**

**W1B**

**W2A**

**X2 W2B**

**W0Z**

**W3A**

**X3 W3B W0B**

For a given node j, linear combination of node inputs and the connection weights called net j is given by:

Net j = ∑ Wij Xij = W0j X0j + W1j X1J + …….+WIJ XIJ

3)  Describe the benefits and drawbacks of using large or small values for the learning rate?

**Answer:**

* If the learning rate is set too low, training will progress very slowly as it makes very tiny updates to the weights in the network.
* If the learning rate is set too high, it can cause undesirable divergent behaviour in the loss function.
* Generally, a large learning rate allows the model to learn faster, at the cost of arriving on a sub-optimal final set of weights.
* A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer to train.
* At extremes, a learning rate that is too large will result in weight updates that will be too large and the performance of the model (such as its loss on the training dataset) will oscillate over training epochs. Oscillating performance is said to be caused by weights that diverge (are divergent).
* A learning rate that is too small may never converge or may get stuck on a suboptimal solution.