ASSIGNMENT- 4

Poorvi Raut- 20009560

Discovering Knowledge in Data: An Introduction to Data Mining, Daniel T. Larose, John Wiley (2004)

Chapter 7, Page 146, #7, 8, and 10

The example is the same as the one in the lecture 8 slides.

Noted that the learning rate = 0.1, although there might be a typo in the textbook/lecture slides saying the learning rate = 0.01.

Questions:

1:) Adjust the weights *W0B, W1B, W2B,* and *W3B* from the example of back-propagation in the text (P137)?

Given table:

$x_0 = 1.0$	$W_{0A} = 0.5$	$W_{0B} = 0.7$	$W_{0Z} = 0.5$
$x_1 = 0.4$	$W_{1A} = 0.6$	$W_{1B} = 0.9$	$W_{AZ} = 0.9$
$x_2 = 0.2$	$W_{2A} = 0.8$	$W_{2B} = 0.8$	$W_{BZ} = 0.9$
$x_3 = 0.7$	$W_{3A} = 0.6$	$W_{3B} = 0.4$	

Wij (NEW)= Wij(CURRENT)+ ΔWij where,

ΔWij = ηδjxij

η = learning rate

xij = ith input to node j.

δj = shows the responsibility for a particular error belonging to node

j.

First pass through network yielded *output* = 0.8750

Target Value = 0.8

Learning Rate = 0.1

Prediction error: Target- First pass output = 0.8-0.8750 =-0.075

δZ= Output(z) * (1-Output(z)) (actual(z)-Output(z)) = -0.0082

δB= Output(b) * (1-Output(b)) (actual(b)-Output(b)) = 0.8176(1 − 0.8176)(0.9)(−0.0082) = −0.0011.

η = learning rate =0.1

Computing weights:
ΔW0B = ηδBX0 = 0.1(-0.0011) (1.0) = -0.00011

W0B (new) = W0B(current) + ΔW0B = 0.7-0.00011 = **0.69989**

ΔW1B = ηδBX1 = 0.1(-0.0011) (0.4) = -0.000044

W1B (new) = W1B(current) + ΔW1B = 0.9-0.000044 = **0.899956**

ΔW2B = ηδBX2= 0.1(-0.0011) (0.2) = -0.000022

W2B (new) = W2B(current) + ΔW2B = 0.8-0.000022 = **0.799978**

ΔW3B = ηδBX3 = 0.1 (-0.0011) (0.7) = -0.000077

W3B (new) = W3B(current) + ΔW3B = 0.4-0.000077 = **0.399923**

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

netA = Σ (i Wi Axi A) = W0A(1) + W1Ax1A + W2Ax2A + W3Ax3A = 0.5 + 0.6(0.4) + 0.8(0.2) + 0.6(0.7) = 1.32

y(A) = 1/ 1 + e ^(−x) = 1/(1 + e^(−1.32)) = 0.7892.

Similary

netB = Σ (i Wi B xi B) = W0B (1) + W1B x1B + W2B x2B + W3B x3B = 0.7 + 0.9(0.4) + 0.8(0.2) + 0.4(0.7) = 1.5

Then y(B) = 1 / (1 + e^ (−1.5)) = 0.8176

Node Z then combines these outputs from nodes A and B.

netZ = Σ (i Wi Z xi Z)= W0Z (1) + WAZ xAZ + WB Z xB Z = 0.5 + 0.9(0.7892) + 0.9(0.8176) = 1.9461

y(z)= 1 / (1 + e^ (−1.9461)) = 0.8750

Assume Target value =0.8

Prediction error: Target– Output = 0.8-0.8750 =**-0.075**

Thus, the prediction error is smaller for the adjusted weights result.

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

Answer: The learning rate is hyperparameter that affects the accuracy and convergence of algorithm. Below are listed some benefits and drawbacks of using smaller or larger learning rate in algorithm.

Benefits of using smaller learning rates:

- Smaller learning rate can leas to better optimization process and prevent overshooting of minimum loss function.
- Using smaller learning rates can help the algorithm converge to slower to global minimum loss.

Drawbacks of using smaller learning rates:

- Smaller learning rates require more training epochs i.e., requires more time to train due to smaller changes made to the weights in each update.

Benefits of using large learning rates:

- Large learning rate can result in faster convergence and less training time, lesser epochs and faster updates.
- In some cases, using a larger learning rate can help the algorithm escape from a local minimum and find a better global minimum.

Drawbacks of using larger learning rates:

- Larger learning rates can result in the algorithm overshooting the minimum of the loss function and diverging.