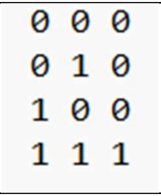
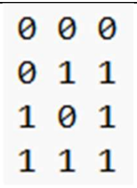
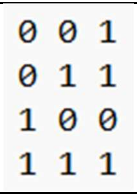
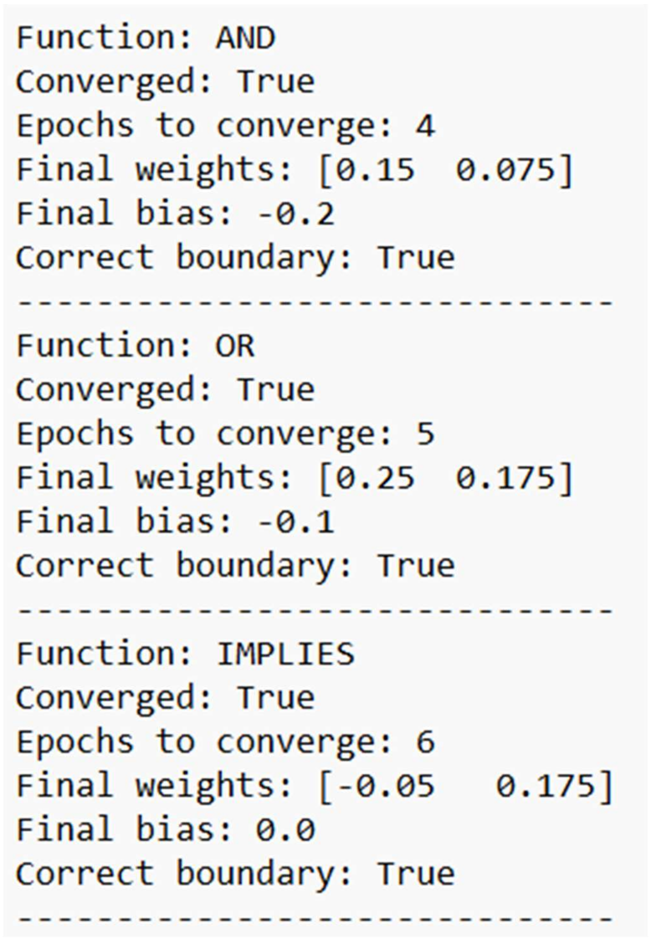
Assignment 2 – ECE 457B

Answer 1

Given below are the text input files (.txt) with the input combinations to the Boolean functions.

 0 0 0 0 1 0 1 0 0 1 1 1	 0 0 0 0 1 1 1 0 1 1 1 1	 0 0 1 0 1 1 1 0 0 1 1 1
AND Boolean inputs	OR Boolean inputs	IMPLIES Boolean inputs

The ch02.py perceptron was used as a template and modified with the specified initial weight w = [0.25, -0.125], learning rate η = 0.1 and bias b = 0. All three Boolean functions were able to achieve convergence within 10 epochs. Given below are the results.

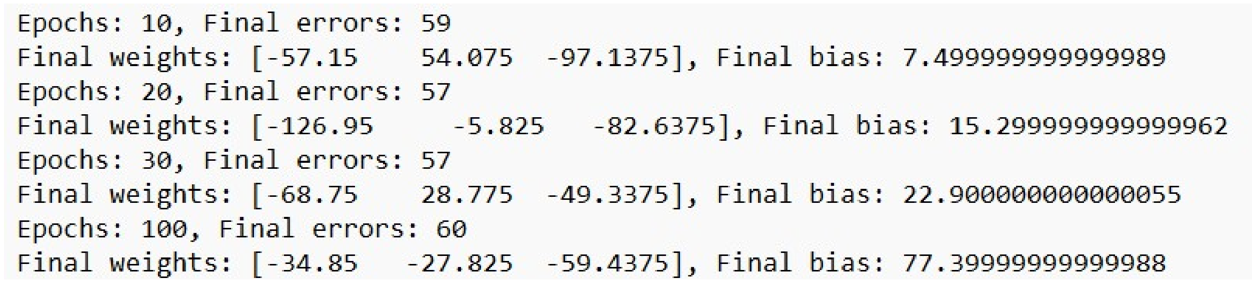


```
Function: AND
Converged: True
Epochs to converge: 4
Final weights: [0.15  0.075]
Final bias: -0.2
Correct boundary: True
------------------------------
Function: OR
Converged: True
Epochs to converge: 5
Final weights: [0.25  0.175]
Final bias: -0.1
Correct boundary: True
------------------------------
Function: IMPLIES
Converged: True
Epochs to converge: 6
Final weights: [-0.05   0.175]
Final bias: 0.0
Correct boundary: True
------------------------------
```

Figure 1: Results of perceptron learning on Boolean functions.

Answer 2

(a) On using the perceptron learning on rectangle.data set with initial weights w = [0.25, -0.125, 0.0625], learning rate η = 0.1 and bias b = 0, the following results are obtained.



```
Epochs: 10, Final errors: 59
Final weights: [-57.15    54.075  -97.1375], Final bias: 7.499999999999989
Epochs: 20, Final errors: 57
Final weights: [-126.95     -5.825   -82.6375], Final bias: 15.299999999999962
Epochs: 30, Final errors: 57
Final weights: [-68.75    28.775  -49.3375], Final bias: 22.900000000000055
Epochs: 100, Final errors: 60
Final weights: [-34.85   -27.825  -59.4375], Final bias: 77.39999999999988
```

Figure 2: Output of the rectangle data perceptron model.

Putting the above results into table format for better comparison.

Table 1: Comparison between the error rates across different epochs

Epochs	Final Weights	Final Bias	Error Rate
10	[-57.15,54.08, -97.14]	7.50	59
20	[-126.95, -5.82, -82.64]	15.30	57
30	[-68.75, 28.775, -49.3375]	22.90	57
100	[-34.85, -27.825, -59.4375]	77.40	60

From the above data the following observations can be made: -

1. The error rate for the perceptron learning model remains between the high 50-60 range.
2. The weight and bias vary across epochs however the learning model does not successful converge to a low error solution.
3. This leads to the conclusion that the rectangle data points are not linearly separable with a single perceptron.

The following graph created shows the high error rates across epochs.

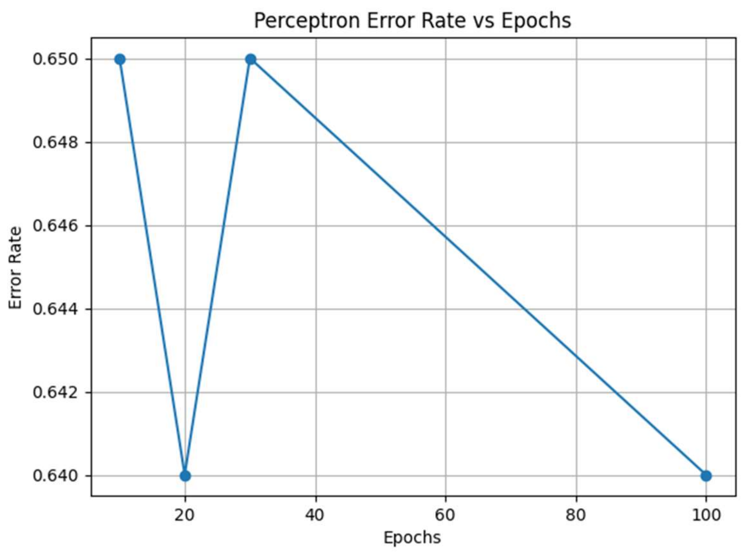


Figure 3: Error rates vs epochs graph.

(b) Following the instructions detailed in the question the following output was obtained.

```
Error proportions per group: [0.6, 1.0, 0.6, 0.6, 0.4, 0.8, 0.4, 0.6]
Groups with error ≤ 0.2: 0
Meets requirement (≥6 groups)? False
```

Figure 4: Using test groups.

According to the question an error proportion success was achieved if 1-δ of the 8 groups are at most ϵ. This translates to at least 6 of the 8 groups having an error of less than or equal to 0.2. This requirement is not met with not a single group having an error less than 0.2 which further shows that a single perceptron is unable to accurately learn the 3D rectangle points. The graph below shows the error proportion of each test group with the 0.2 threshold marked by the red dotted line.

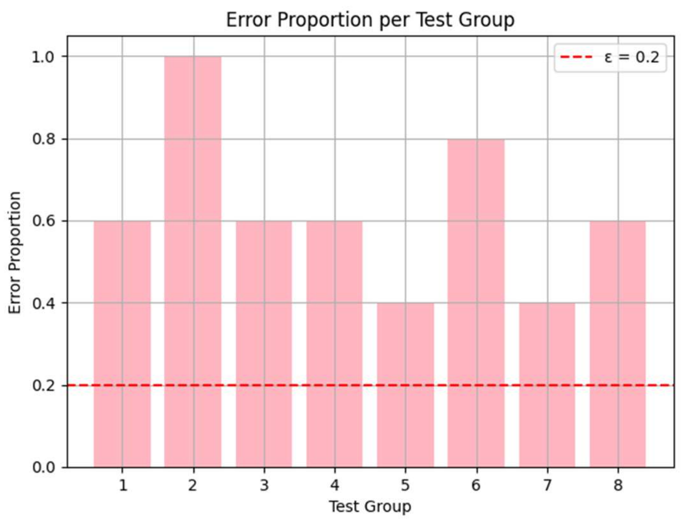
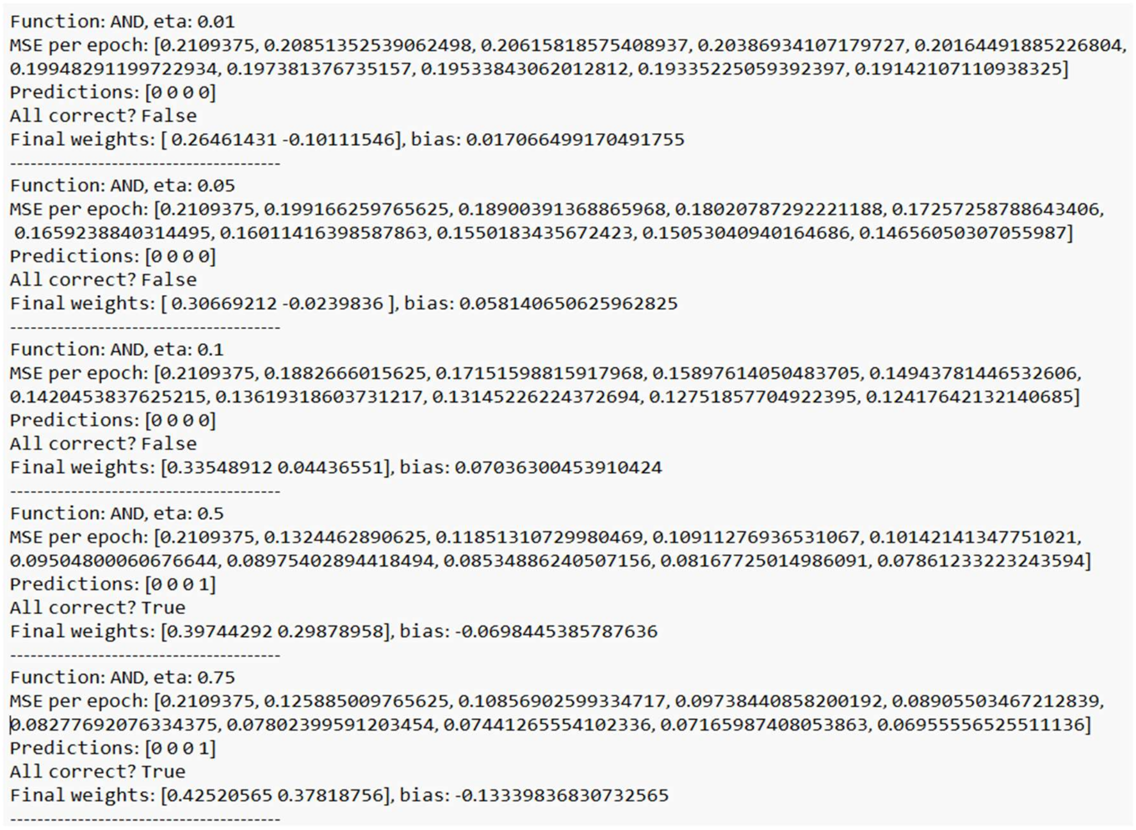


Figure 5: Graph showing the number of test groups above the error tolerance threshold.

Answer 3

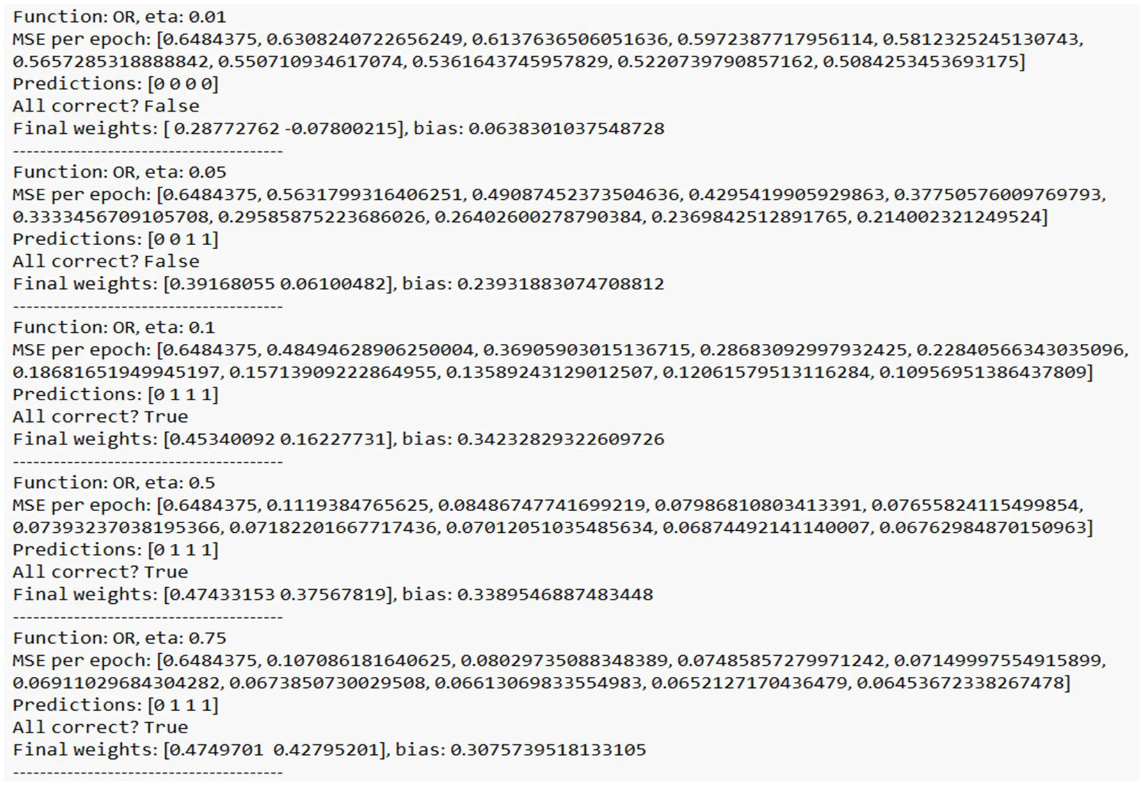
The output using different learning rates for AND is as follows: -



```
Function: AND, eta: 0.01
MSE per epoch: [0.2109375, 0.20851352539062498, 0.20615818575408937, 0.20386934107179727, 0.20164491885226804,
0.19948291199722934, 0.197381376735157, 0.19533843062012812, 0.19335225059392397, 0.19142107110938325]
Predictions: [0 0 0 0]
All correct? False
Final weights: [ 0.26461431 -0.10111546], bias: 0.017066499170491755
---------------------------------------
Function: AND, eta: 0.05
MSE per epoch: [0.2109375, 0.199166259765625, 0.18900391368865968, 0.18020787292221188, 0.17257258788643406,
 0.1659238840314495, 0.16011416398587863, 0.1550183435672423, 0.15053040940164686, 0.14656050307055987]
Predictions: [0 0 0 0]
All correct? False
Final weights: [ 0.30669212 -0.0239836 ], bias: 0.058140650625962825
---------------------------------------
Function: AND, eta: 0.1
MSE per epoch: [0.2109375, 0.1882666015625, 0.17151598815917968, 0.15897614050483705, 0.14943781446532606,
0.1420453837625215, 0.13619318603731217, 0.13145226224372694, 0.12751857704922395, 0.12417642132140685]
Predictions: [0 0 0 0]
All correct? False
Final weights: [0.33548912 0.04436551], bias: 0.07036300453910424
---------------------------------------
Function: AND, eta: 0.5
MSE per epoch: [0.2109375, 0.1324462890625, 0.11851310729980469, 0.10911276936531067, 0.10142141347751021,
0.09504800060676644, 0.08975402894418494, 0.08534886240507156, 0.08167725014986091, 0.07861233223243594]
Predictions: [0 0 0 1]
All correct? True
Final weights: [0.39744292 0.29878958], bias: -0.0698445385787636
---------------------------------------
Function: AND, eta: 0.75
MSE per epoch: [0.2109375, 0.125885009765625, 0.10856902599334717, 0.09738440858200192, 0.08905503467212839,
0.08277692076334375, 0.07802399591203454, 0.07441265554102336, 0.07165987408053863, 0.06955556525511136]
Predictions: [0 0 0 1]
All correct? True
Final weights: [0.42520565 0.37818756], bias: -0.13339836830732565
---------------------------------------
```

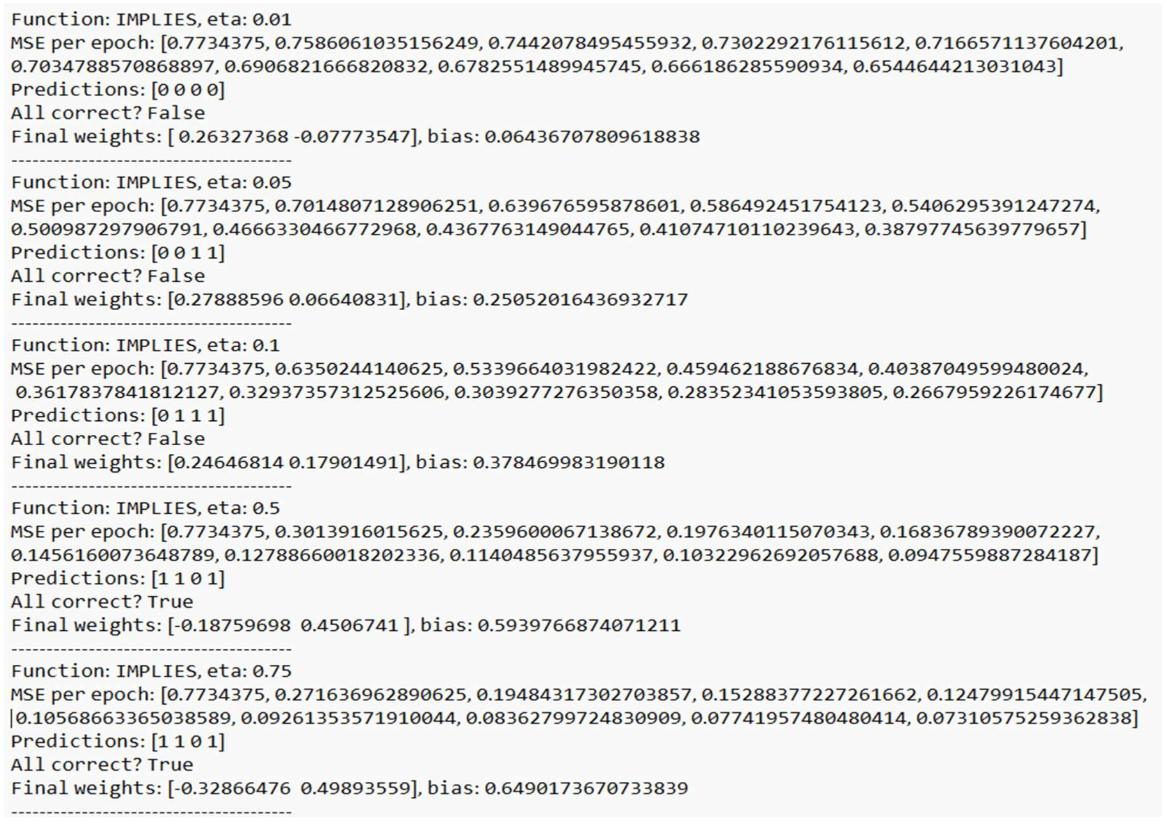
Figure 6: AND operator learning using AdalineGD.

Similarly, below is the output for OR and IMPLIES.



```
Function: OR, eta: 0.01
MSE per epoch: [0.6484375, 0.6308240722656249, 0.6137636506051636, 0.5972387717956114, 0.5812325245130743,
0.5657285318888842, 0.550710934617074, 0.5361643745957829, 0.5220739790857162, 0.5084253453693175]
Predictions: [0 0 0 0]
All correct? False
Final weights: [ 0.28772762 -0.07800215], bias: 0.0638301037548728
----------------------------------------
Function: OR, eta: 0.05
MSE per epoch: [0.6484375, 0.5631799316406251, 0.49087452373504636, 0.4295419905929863, 0.37750576009769793,
0.3333456709105708, 0.29585875223686026, 0.26402600278790384, 0.2369842512891765, 0.214002321249524]
Predictions: [0 0 1 1]
All correct? False
Final weights: [0.39168055 0.06100482], bias: 0.23931883074708812
----------------------------------------
Function: OR, eta: 0.1
MSE per epoch: [0.6484375, 0.48494628906250004, 0.36905903015136715, 0.28683092997932425, 0.22840566343035096,
0.18681651949945197, 0.15713909222864955, 0.13589243129012507, 0.12061579513116284, 0.10956951386437809]
Predictions: [0 1 1 1]
All correct? True
Final weights: [0.45340092 0.16227731], bias: 0.34232829322609726
----------------------------------------
Function: OR, eta: 0.5
MSE per epoch: [0.6484375, 0.1119384765625, 0.08486747741699219, 0.07986810803413391, 0.07655824115499854,
0.07393237038195366, 0.07182201667717436, 0.07012051035485634, 0.06874492141140007, 0.06762984870150963]
Predictions: [0 1 1 1]
All correct? True
Final weights: [0.47433153 0.37567819], bias: 0.3389546887483448
----------------------------------------
Function: OR, eta: 0.75
MSE per epoch: [0.6484375, 0.107086181640625, 0.08029735088348389, 0.07485857279971242, 0.07149997554915899,
0.06911029684304282, 0.0673850730029508, 0.06613069833554983, 0.0652127170436479, 0.06453672338267478]
Predictions: [0 1 1 1]
All correct? True
Final weights: [0.4749701  0.42795201], bias: 0.3075739518133105
----------------------------------------
```

Figure 7: OR operator learning using AdalineGD.



```
Function: IMPLIES, eta: 0.01
MSE per epoch: [0.7734375, 0.7586061035156249, 0.7442078495455932, 0.7302292176115612, 0.7166571137604201,
0.7034788570868897, 0.6906821666820832, 0.6782551489945745, 0.666186285590934, 0.6544644213031043]
Predictions: [0 0 0 0]
All correct? False
Final weights: [ 0.26327368 -0.07773547], bias: 0.06436707809618838
----------------------------------------
Function: IMPLIES, eta: 0.05
MSE per epoch: [0.7734375, 0.7014807128906251, 0.639676595878601, 0.586492451754123, 0.5406295391247274,
0.500987297906791, 0.4666330466772968, 0.4367763149044765, 0.41074710110239643, 0.38797745639779657]
Predictions: [0 0 1 1]
All correct? False
Final weights: [0.27888596 0.06640831], bias: 0.25052016436932717
----------------------------------------
Function: IMPLIES, eta: 0.1
MSE per epoch: [0.7734375, 0.6350244140625, 0.5339664031982422, 0.459462188676834, 0.40387049599480024,
 0.3617837841812127, 0.32937357312525606, 0.3039277276350358, 0.28352341053593805, 0.2667959226174677]
Predictions: [0 1 1 1]
All correct? False
Final weights: [0.24646814 0.17901491], bias: 0.378469983190118
----------------------------------------
Function: IMPLIES, eta: 0.5
MSE per epoch: [0.7734375, 0.3013916015625, 0.2359600067138672, 0.1976340115070343, 0.16836789390072227,
0.1456160073648789, 0.12788660018202336, 0.1140485637955937, 0.10322962692057688, 0.0947559887284187]
Predictions: [1 1 0 1]
All correct? True
Final weights: [-0.18759698  0.4506741 ], bias: 0.5939766874071211
----------------------------------------
Function: IMPLIES, eta: 0.75
MSE per epoch: [0.7734375, 0.271636962890625, 0.19484317302703857, 0.15288377227261662, 0.12479915447147505,
|0.10568663365038589, 0.09261353571910044, 0.08362799724830909, 0.07741957480480414, 0.07310575259362838]
Predictions: [1 1 0 1]
All correct? True
Final weights: [-0.32866476  0.49893559], bias: 0.6490173670733839
----------------------------------------
```

Figure 8: IMPLIES operator learning using AdalineGD.

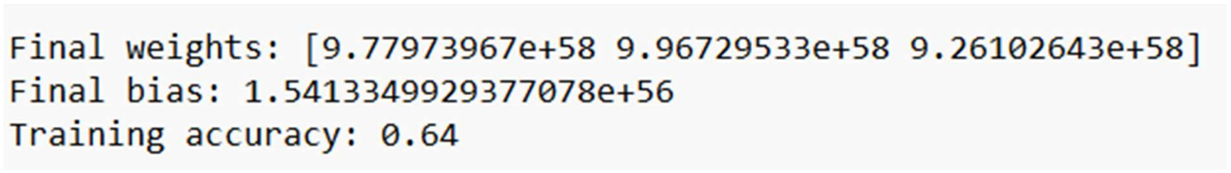
The table below summarizes the results of the various learning rates so that accurate conclusions can be made.

Learning Rate (η)	MSE Converges to 0	Final w, b Correct
0.01	Slow	Yes
0.05	Yes	Yes
0.1	Yes	Yes
0.5	No, it oscillates	No
0.75	No, does not converge	No

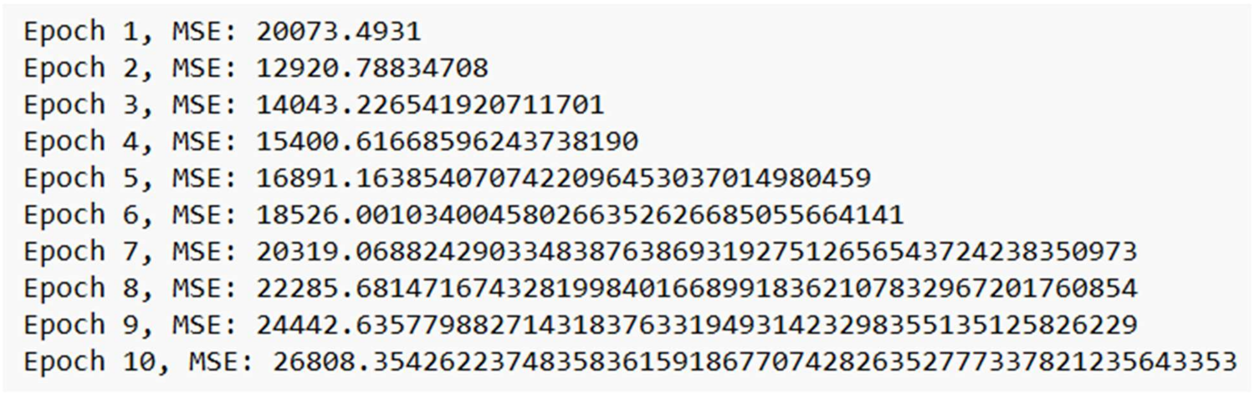
Through this comparison data the following conclusions can be drawn. Learning rate of η = 0.1 is generally the best. The smaller learning rates take too long to converge while the larger ones do not converge or diverge. Thus only the moderate learning rates of 0.05 to 0.1 yield reliable weights and bias for all the Boolean operator inputs.

Answer 4

The output is given below,



```
Final weights: [9.77973967e+58 9.96729533e+58 9.26102643e+58]
Final bias: 1.5413349929377078e+56
Training accuracy: 0.64
```



```
Epoch 1, MSE: 20073.4931
Epoch 2, MSE: 12920.78834708
Epoch 3, MSE: 14043.226541920711701
Epoch 4, MSE: 15400.61668596243738190
Epoch 5, MSE: 16891.1638540707422096453037014980459
Epoch 6, MSE: 18526.001034004580266352626685055664141
Epoch 7, MSE: 20319.06882429033483876386931927512656543724238350973
Epoch 8, MSE: 22285.68147167432819984016689918362107832967201760854
Epoch 9, MSE: 24442.63577988271431837633194931423298355135125826229
Epoch 10, MSE: 26808.3542622374835836159186770742826352777337821235643353
```

Figure 9: Rectangle data using AdalineGD.

The data shows that the MSE value decreases over time but never really reaches 0, this implies the model is learning but is unable to perfectly fit the data. The final bias and weight are not a perfect separation, and the overall accuracy remains pretty low. This outcome is sort of expected since the previous learning model with perceptron was unable to linearly separate the 3D rectangle.data which hints that AdalineGD which is a linear model may also be unable separate the data.