Assignment 2-Machine Learning

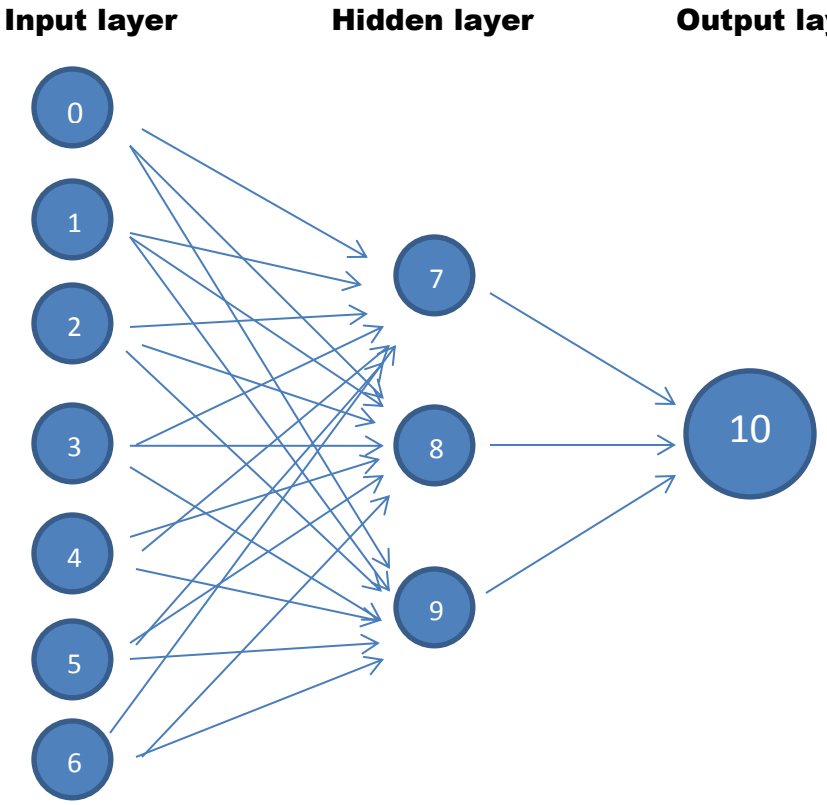
Professor: Dr. Yuan

Author: Kaikai Bian

[TRAINING MULTI-LAYER NEURAL NETWORKS]

Predict CPU performances based on 3-layer neural network.

In this assignment, I used multilayer neural network to predict the relative CPU performance based on its cycle time, memory size and so on. Each data point has 7 features and 1 true output. I separated the data points into 5 folds and trained the program with the first 4 folds to obtain relative weights, and then evaluated the performance with the last fold. Through this method, I tried different thresholds to train the neural network, such as 0.1, 0.01 and 0.001, and observed a number of intriguing phenomena. The multilayer neural network is presented as follows. It has 3 layers: input layer, hidden layer, and output layer. The input layer contains 7 nodes, each representing a specific feature. The hidden layer includes 3 hidden units, and output layer has only 1 node.



In the hidden layer and output layer, each unit contains a sigmoid function, g(z)=1/(1+e^(-z)). In the first layer, we feed the individual features of each data point into each node, and then in the second layer, each hidden unit will compute their weighted sum by the sigmoid function. Lastly, this result will be sent into the output unit and computed by sigmoid function. We will compare this final result with the normalized true output, obtaining the error value. Repeating this process for every data point, we will get a total error value. Our goal is to minimize this value as small as possible. In this procedure, we use some threshold to control the minimization.

Then we use the error values to back-propagate/adjust the weights between nodes in different layers. The program outputs the concluding result and plots chart to represent the result.

1. **Experiment with threshold 0.1, learning rate 0.5:**

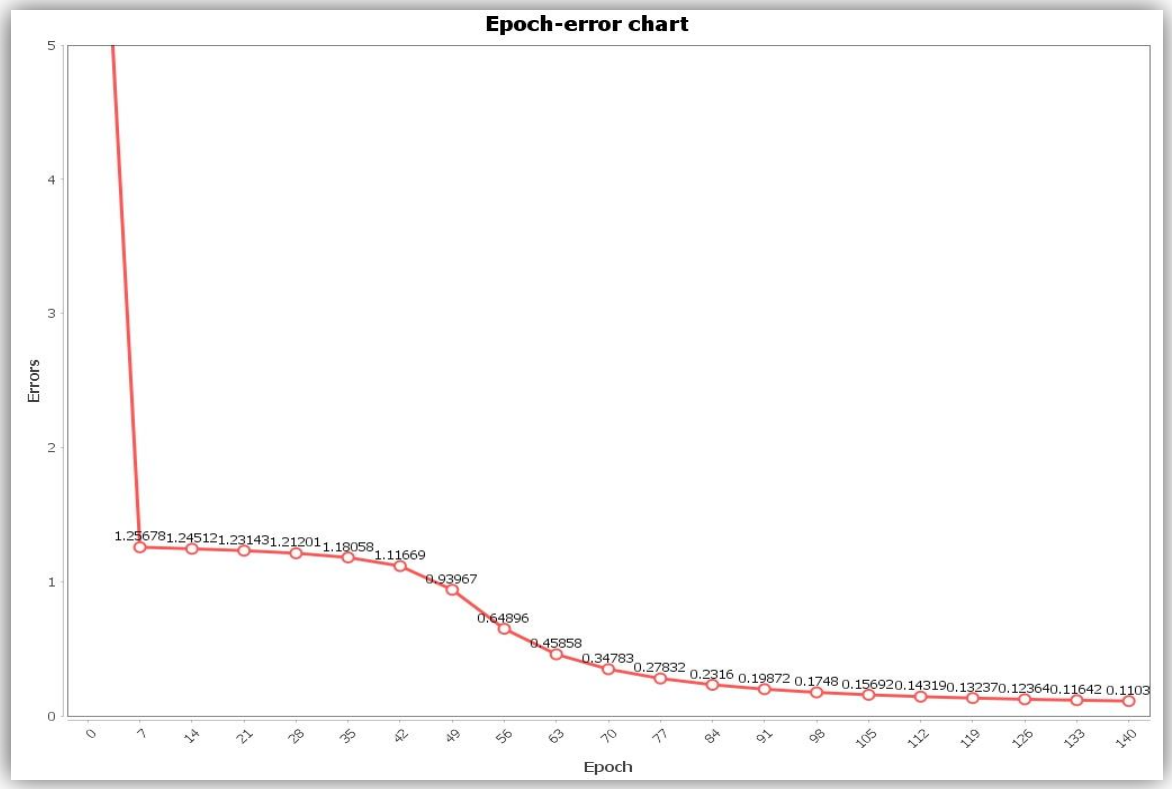
A) Predicted outputs:

0: 0.16234564303278823
1: 0.022087269930450636
2: 0.02358218717374349
3: 0.03933067646198036
4: 0.01891060722941621
5: 0.020186524014297613
6: 0.029666310920322646
7: 0.02647087852746494
8: 0.025408910939865904
9: 0.027826924728803347
10: 0.03360928290557056
11: 0.04058364756426368
12: 0.01831378456288683
13: 0.01891445913866375
14: 0.03744909023006027
15: 0.0430554721872489
16: 0.04577663464699074
17: 0.09206057234828785
18: 0.06864974202145964
19: 0.08089179030253793
20: 0.14473901863619076
21: 0.16678424537217193
22: 0.23810034128805393
23: 0.31681453855831526
24: 0.04712293820039068
25: 0.12368109756920194
26: 0.1701812254385226
27: 0.2475445184339292
28: 0.2761610145301684
29: 0.41653210850985334
30: 0.45405775563015693
31: 0.023727495156105425
32: 0.02418911045682533
33: 0.02384935213294908
34: 0.024334076853072817
35: 0.03536637271310129
36: 0.045982525228261566
37: 0.03818684841542592
38: 0.011841047277936735
39: 0.00930105665013697

True outputs:

0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618

These two charts are the predicted outputs obtained by the program and true outputs from original data set. After comparing the two sets of outputs, we can realize that predicted outputs have been approximating to the true outputs.



From this chart, we observe that the total errors decrease extremely fast at the beginning, and then slow down its speed of decreasing, but during some epochs, the decreasing speeds up and then turns to be smooth. Hence, the curve presented seems to be in a sigmoid shape.

The weights of each edge from the input layer to the hidden layer, and then from hidden layer to the output layer are listed in the following:

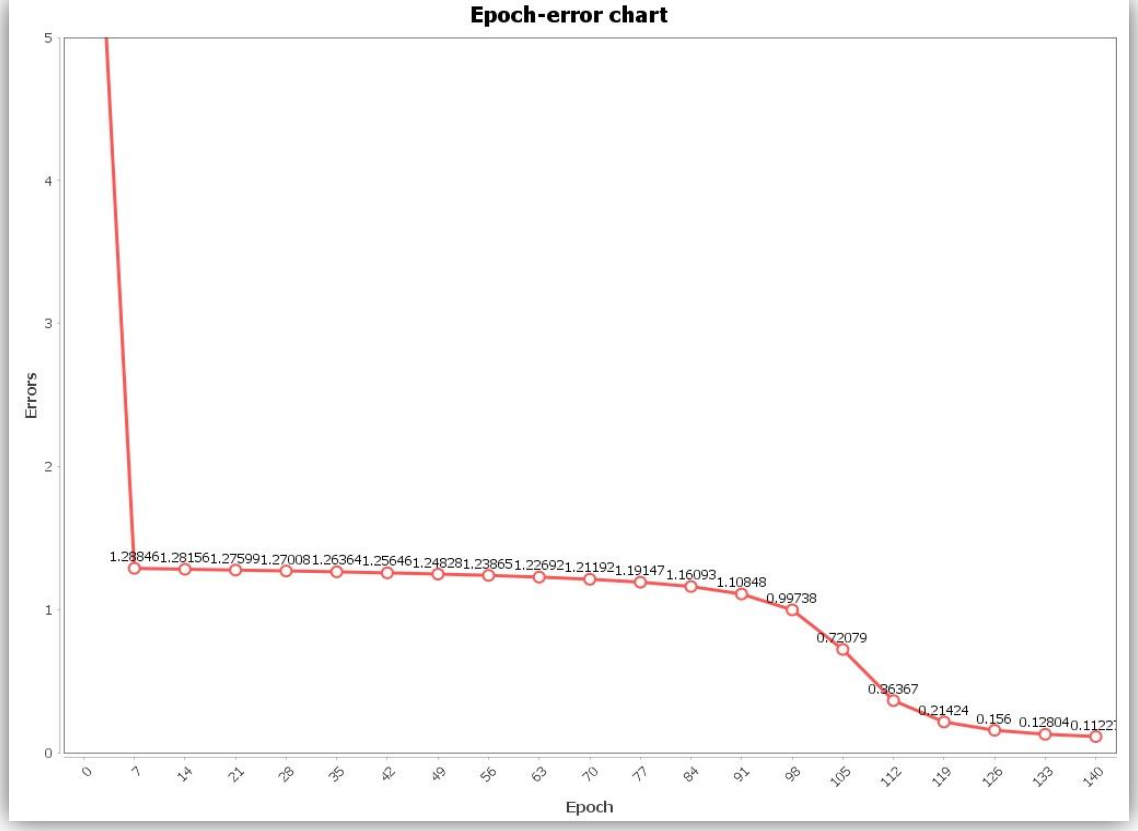
3.068895181437563, 0.41849014363338355, 1.7457612901525108, -1.072885931992936, 1.34887688508634, 1.1545942304154517, -1.0928673614121542, 0.801740915855917, -0.45240121842682973, -0.5290747413746102, 1.0407893571603077, 1.4157968506602367, -0.3096582871373117, 1.0395087775174083, 0.13848683319738264, 0.2656114541158491, 0.394078601052223, 0.20171276779306802, -0.33297267492940114, 1.8092649076053386, 0.15717146774125512, -7.480389056034305, 0.8664460685601818, 0.1595695590521706.

2. Experiment with threshold 0.1, learning rate 0.5:

The predicted outputs approximate to the true outputs.

B) Predicted outputs:
0: 0.128170444469044
1: 0.025002750025844237
2: 0.02728140002221223
3: 0.04532657890897104
4: 0.022780640507001435
5: 0.023132879631694987
6: 0.04145450999107947
7: 0.029680910031756456
8: 0.026267668302573544
9: 0.030087464776955914
10: 0.038655601077157646
11: 0.04753656357308558
12: 0.020905066834287705
13: 0.02181023374114911
14: 0.035960943316589604
15: 0.03846415753451661
16: 0.045303853667868876
17: 0.09629766318105458
18: 0.06539090148741837
19: 0.07702604949984207
20: 0.12033806131553741
21: 0.1611331461233649
22: 0.23685558443364757
23: 0.2789144624030052
24: 0.043996640937949645
25: 0.14002835595653187
26: 0.12465371255121402
27: 0.09947613392371228
28: 0.12197532055781397
29: 0.45306831552369936
30: 0.5089664552746153
31: 0.027001811359842677
32: 0.02743216928488291
33: 0.027295053167212213
34: 0.027780043267545504
35: 0.039794537605506726
36: 0.04847570608731723
37: 0.04062539434818969
38: 0.019313235830591604
39: 0.01407693064996361

True outputs:
0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618



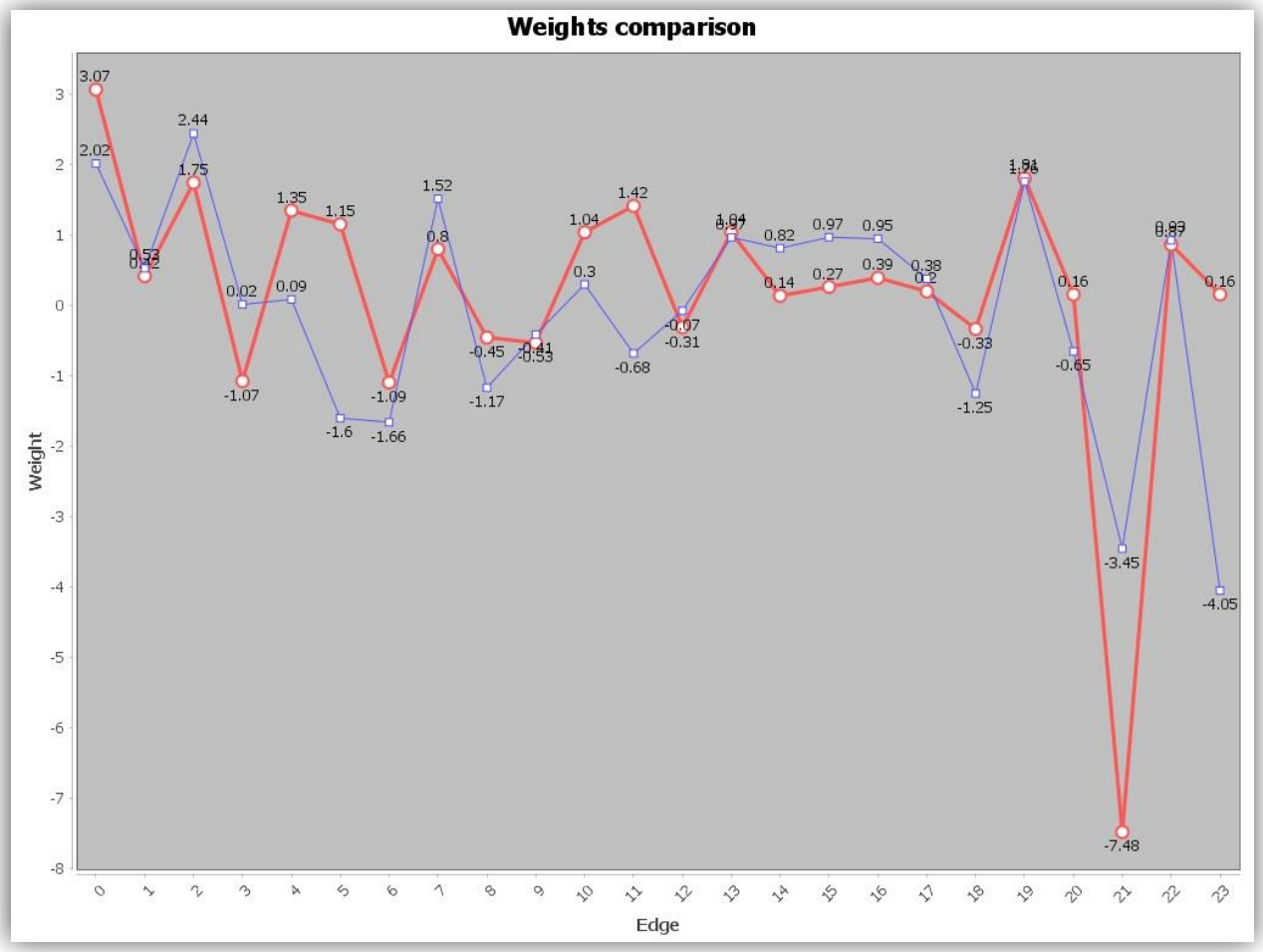
In this chart, we find that the turnaround appears later than the first experiment.

The weights of edges are listed as follows:

2.018319404590006, 0.5285060509673671, 2.4429370174031297, 0.01575942410892352, 0.08716352836794658, -1.5998170002863852, -1.657367330213325, 1.5177646198131947, -1.1673063142855102, -0.4102739936505453, 0.30106480126776175, -0.6803318962773288, -0.07389039895083285, 0.9697464834795433, 0.8153226379793883, 0.9713073617633685, 0.9489746144386406, 0.3816388957190291, -1.250662341191915, 1.764016187050276, -0.6501337314940963, -3.453059579607419, 0.9257277536388722, -4.048235846779631.

This set of weights differs from the first set of weights; hence there does not exist a "standard" weight the network approaches to.

In the following, I plot all the weights of both the first and second experiment.



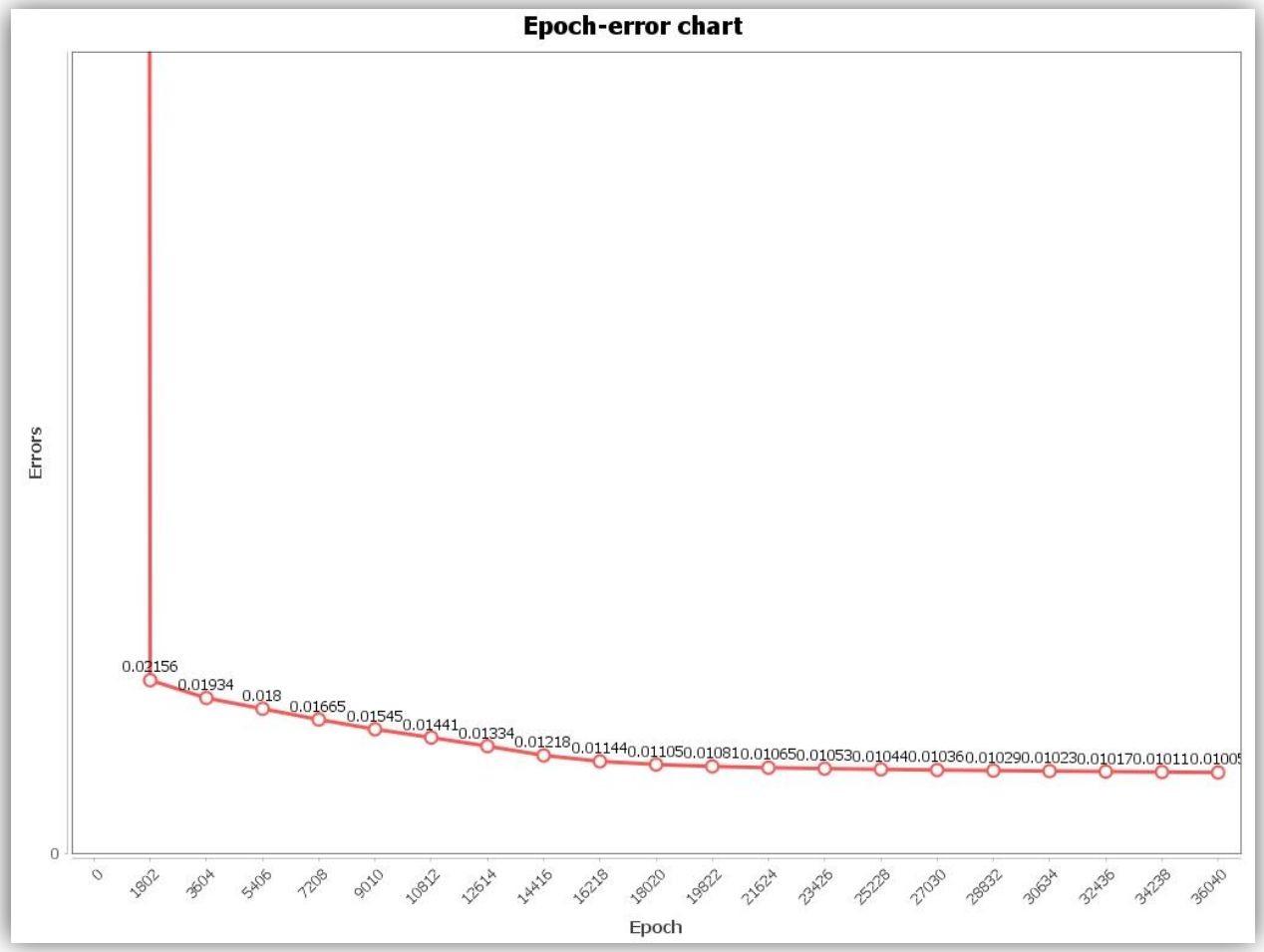
We can observe that the weights of edges with threshold of 0.1 are approximate and show similar behaviors in different edges.

3. Experiment with threshold 0.01, learning rate 0.5:

The predicted outputs approximate to the true outputs.

Predicted outputs:
0: 0.14353130878431275
1: 0.021523093045630992
2: 0.02438180862352515
3: 0.04642538649130571
4: 0.023575889531713017
5: 0.024738491644665284
6: 0.04846356568606686
7: 0.024865032959642284
8: 0.022772351994301207
9: 0.02663933057496433
10: 0.03550475312911368
11: 0.04073706577386507
12: 0.020304228760979905
13: 0.02169576556512996
14: 0.028014819316731082
15: 0.03472714232810496
16: 0.03857314378838864
17: 0.07224099930001567
18: 0.05061057414283853
19: 0.059052412776346465
20: 0.10649926204401547
21: 0.1287123364794501
22: 0.21271941148951617
23: 0.32009179433908563
24: 0.043760306702655516
25: 0.14574842391833231
26: 0.22258747672731327
27: 0.36592897608538966
28: 0.3932997582786706
29: 0.8091616046150388
30: 0.8505117397924159
31: 0.023817554597919097
32: 0.024002874494572533
33: 0.0239133463299115
34: 0.024115509480021786
35: 0.03242322163237453
36: 0.039394781746132077
37: 0.03426881115624963
38: 0.031964467286332676
39: 0.021074672216177728

True outputs:
0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618



In this chart, I minimize the range of the y-axis to 0-0.1. We can observe that the errors diminish tremendously fast at the beginning epochs and then the curve turns to be comparatively smooth. It does not show any sigmoid shape in this chart.

The following are the weights of individual edges:

0.9677050696130661, 0.34790988909321685, 0.46857161549405485, -0.06608713501686682, 1.6478435166989351, 0.4490712319405222, -1.0019606073023417, -2.469915093528045, -1.9972627899498956, -0.25800363430126455, -3.5988521136534346, -1.7705595514858359, 1.8111099698058224, 0.6056643182130607, 0.41968147428302033, 0.0026008244148482264, 2.5681691036445256, 0.9234045447394361, 0.9201591509449241, 2.029920094124157, 0.7806903782624406, 3.0916246231657736, 7.052500279518403, -17.937293724639815.

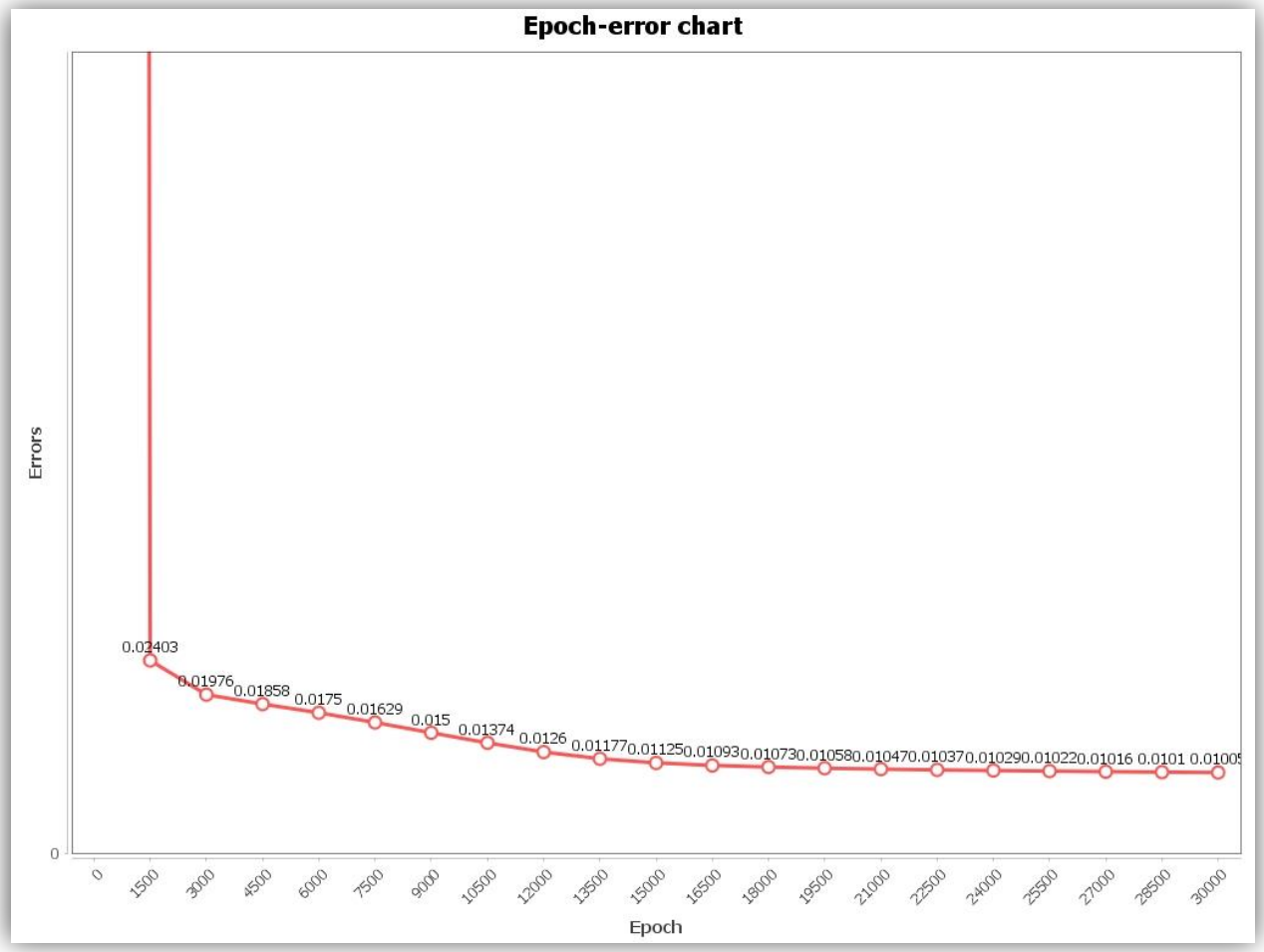
4. Experiment with threshold 0.01, learning rate 0.5:

The predicted outputs approximate to the true outputs.

Since the threshold has been limited to 0.01, the predicted outputs approximate the true outputs much better than the experiments with threshold of 0.1.

Predicted outputs:
0: 0.14328500702185534
1: 0.02159878164365376
2: 0.024422403063100274
3: 0.04583846907198754
4: 0.023580602522410135
5: 0.024785944709536104
6: 0.04796945235426951
7: 0.02489189593310087
8: 0.02287396943753777
9: 0.026691791897841606
10: 0.03541679229161842
11: 0.04064047867851734
12: 0.020349513629587316
13: 0.021724870704166354
14: 0.028174469460148968
15: 0.034908049310852625
16: 0.038654942042921074
17: 0.07209310910775552
18: 0.050795510887247035
19: 0.059194983513276615
20: 0.10721926921829575
21: 0.12816656300088436
22: 0.21254164923320903
23: 0.3243113257374739
24: 0.04383526495631184
25: 0.14554673657783823
26: 0.22414155423927123
27: 0.33129114571823337
28: 0.35195454533996595
29: 0.8048410326547644
30: 0.8407174407942802
31: 0.023826465505207853
32: 0.02402380174422875
33: 0.023922736114794334
34: 0.02413695609054437
35: 0.03239378262368602
36: 0.03923206137589717
37: 0.03434758735833371
38: 0.031365061427975634
39: 0.021020230081033108

True outputs:
0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618



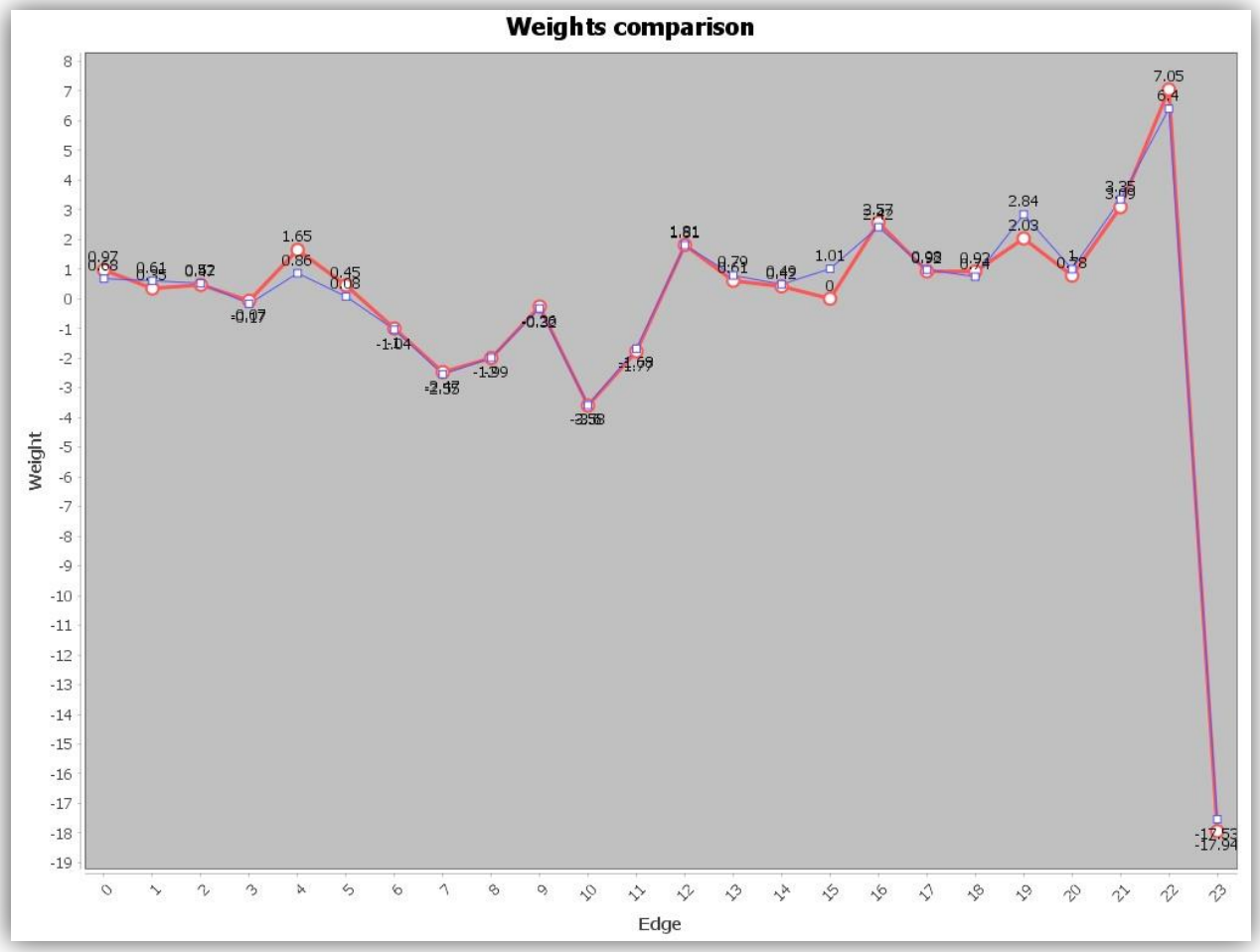
This chart shows us a similar curve with the last experiment. Decrease excessively fast at the beginning and then suddenly turn to be flat. This is because the more epochs the program run, the smaller error values it can deduct.

The weights of edges are as follows:

0.678690983373701, 0.6068863039134603, 0.524646535931502, -0.17063370526707775, 0.8628670472529243, 0.0759031696323422, -1.040609826070455, -2.5474263826866292, -1.9865953556753397, -0.3240081004553215, -3.5830423534353257, -1.6806000864403148, 1.806765987961375, 0.7876696424628783, 0.4899835643204287, 1.0098485289171895, 2.4174115922643193, 0.9828101024967129, 0.7431927715844727, 2.8409254065287715, 0.9983520891489454, 3.3501928486074566, 6.400930375056133, -17.533588571777887.

Comparing the two weights, observe that the last 3 weights, which represent the edges between the hidden layer and output layer, are approximate.

I also made a chart to present the weights to observe whether I can find some differences. With threshold of 0.01, the two sets of weights are so matching.



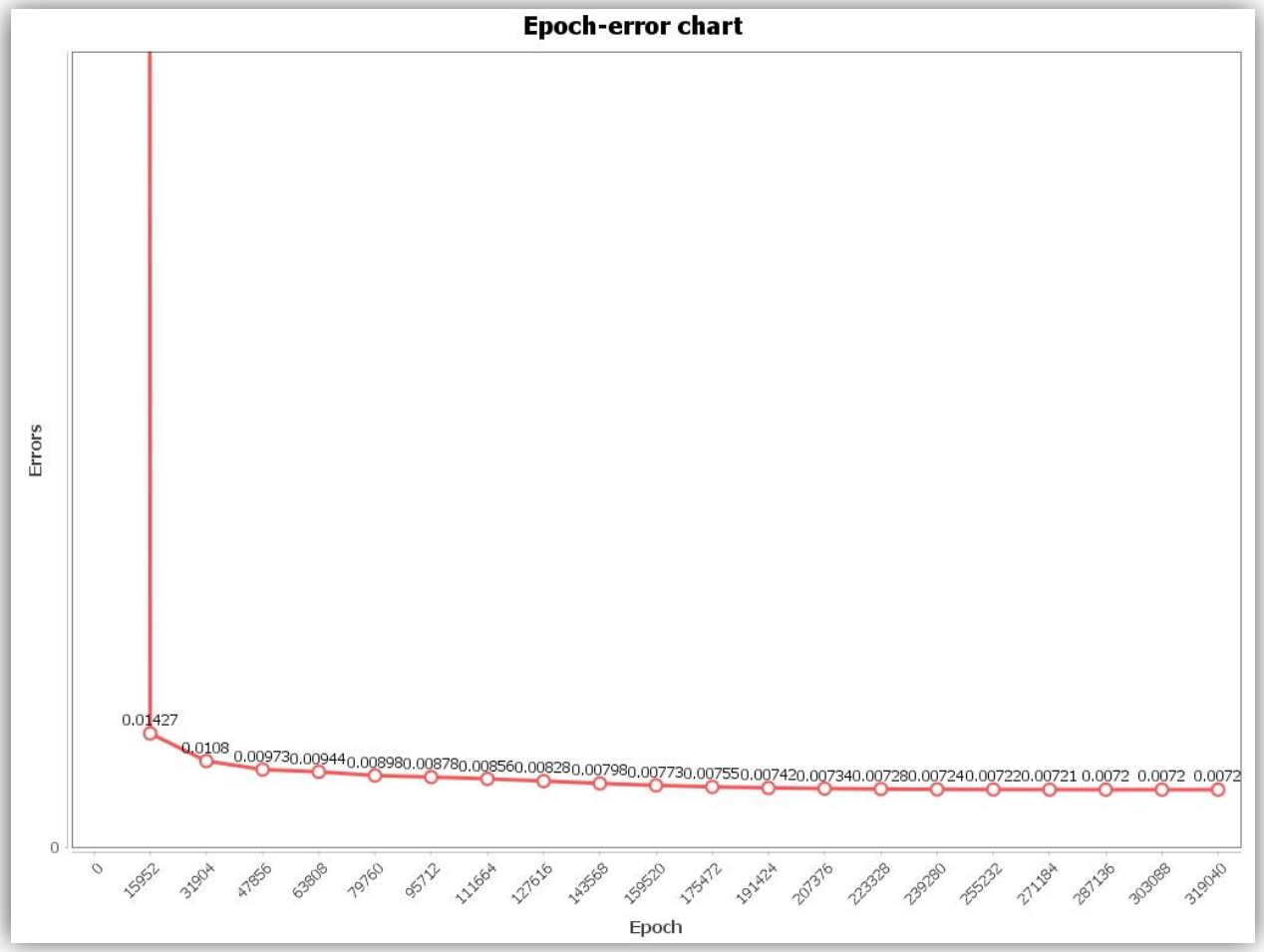
5. Experiment with threshold 0.001, learning rate 0.5:

This set of predicted outputs approximate the true outputs much better than the previous ones.

Predicted outputs:
0: 0.15395832155958536
1: 0.019907276478122814
2: 0.022438362541108776
3: 0.05033698973477141
4: 0.021151608566556894
5: 0.022340702937165263
6: 0.045299991751305356
7: 0.023037500700144133
8: 0.021275444624160245
9: 0.02453501395624639
10: 0.032794023779363655
11: 0.03833823090128563
12: 0.019010946258596725
13: 0.020159236637897027
14: 0.028084348760532847
15: 0.03585226914705448
16: 0.037991561213797174
17: 0.07253123533465182
18: 0.052862546351832795
19: 0.06061703023682995
20: 0.11117821666039664
21: 0.13511917421048128
22: 0.201552745606094
23: 0.2977114518283195
24: 0.04296664967348548
25: 0.15083016404147942
26: 0.2156450379716417
27: 0.33050782559310177
28: 0.34410280696204487
29: 0.2500059880769084
30: 0.037521400681617835
31: 0.02218334934334096
32: 0.022420271163714214
33: 0.022141216140547912
34: 0.02236847908120996
35: 0.030250652712183225
36: 0.03944648596839876
37: 0.03226384832493776
38: 0.029350323805796878
39: 0.01782590125208908

True outputs:
0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618

The program ended while it had not reached the threshold but reached the minimum error values. Becauase the weights were initialized randomly, it can lead to half-way ending like this. Nevertheless, observe that the predicted values still approximate the true much better than the previous ones.



The chart of threshold of 0.001 just resembles the counterpart of threshold of 0.01 with slight difference. It turns to be smoother and there seems to be no deduction in the end.

The weights of edges are listed in the following:

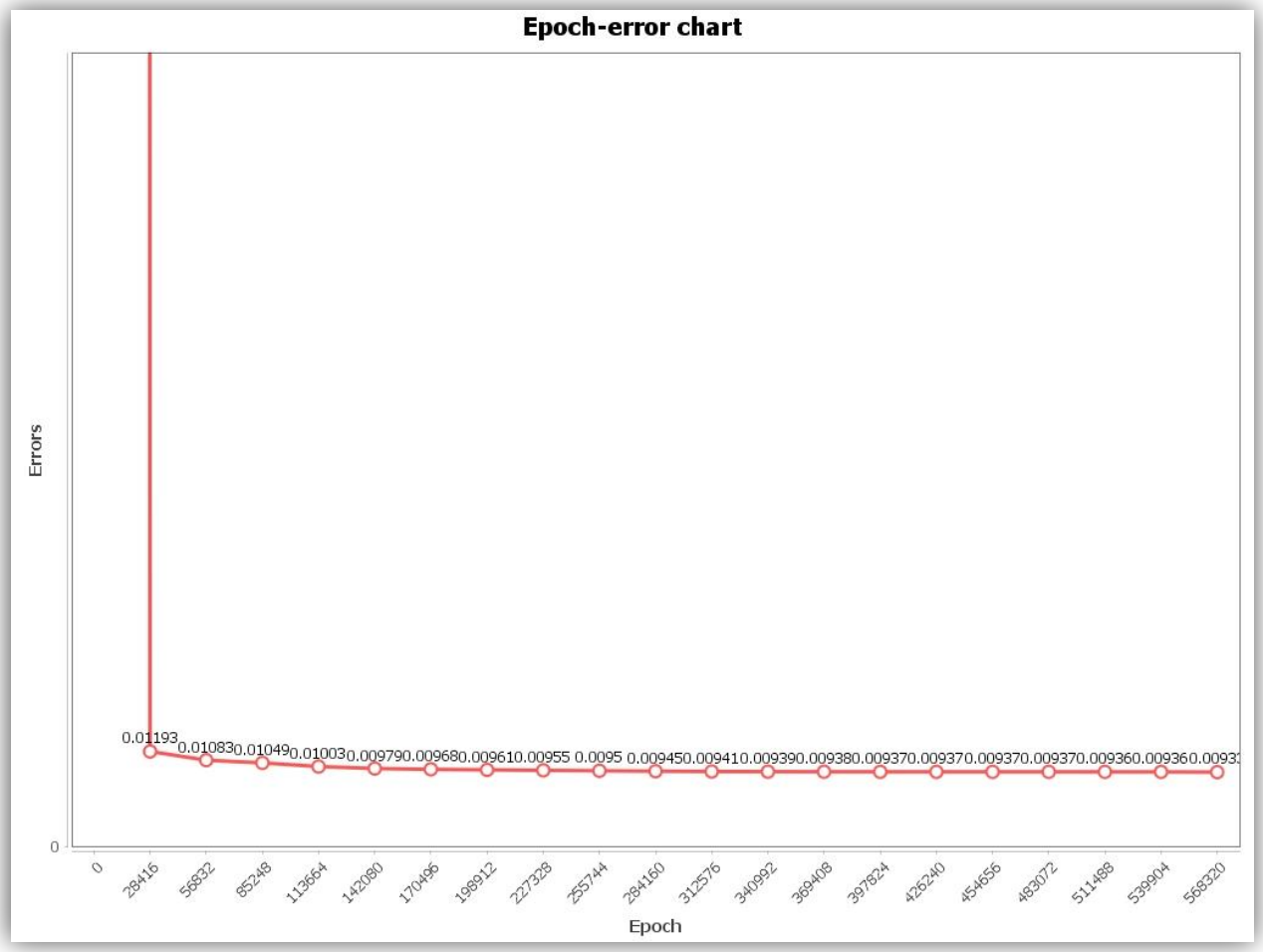
-2.303588050168534, -1.071318092847127, -1.6415469951438375, -3.5606258113467306, -1.6078861146336052, -0.4329588215459802, -3.45837174868928, -2.407584457364578, -1.699773589498611, -1.2001421049479393, -1.9917597691144382, -4.0500739008088456, -2.0873464162398805, -1.3360145383117643, -1.3880415715060503, -0.5455611948926716, 0.7390912328431472, 3.026047231181294, 9.532180610998084, 4.648891693869785, 4.358121493013824, 8.050091877820153, -22.78140558046731, 6.929937367880352.

6. Experiment with threshold 0.001, learning rate 0.5:

This set of predicted outputs approximate the true outputs much better than the previous ones, even though it stopped at the 596754 epochs since it reached the minimum value of total errors.

Predicted outputs:
0: 0.18398331713591112
1: 0.020492677187749064
2: 0.023059338592774023
3: 0.05238540126251654
4: 0.02191568158787993
5: 0.022677564672001563
6: 0.04517199974629256
7: 0.02334337451561981
8: 0.021360635271119824
9: 0.02529751131800165
10: 0.033578994297624835
11: 0.03940284756586978
12: 0.019727424730594207
13: 0.020919190681247885
14: 0.02811111880031169
15: 0.03498109905562144
16: 0.03797324187854435
17: 0.0731335408173717
18: 0.050728326222084515
19: 0.060477346370635725
20: 0.11239708543345224
21: 0.12854770104064112
22: 0.21159097256048837
23: 0.34825876497976493
24: 0.04336811002184157
25: 0.14283470790646297
26: 0.1767762271376241
27: 0.2532312209240168
28: 0.2600245264828188
29: 0.9234776332593072
30: 0.962066796744213
31: 0.02249974421191948
32: 0.02278380201051014
33: 0.02246514417084856
34: 0.0227467151890991
35: 0.03041828473841803
36: 0.040022129661610464
37: 0.03271587638676319
38: 0.03564317409050222
39: 0.0196536451577331

True outputs:
0: 0.13662631203770062
1: 0.01945276395282272
2: 0.02250389878858121
3: 0.04454584970970098
4: 0.021781463680551498
5: 0.022662477741453357
6: 0.04729431846916678
7: 0.02352479724694016
8: 0.02127297236775973
9: 0.025398840472375277
10: 0.03467837031390077
11: 0.03898736443420334
12: 0.01832986857389347
13: 0.019816384163121516
14: 0.026417655521321134
15: 0.03359885825191587
16: 0.03772498441677026
17: 0.07311093511407776
18: 0.04945973893039902
19: 0.057979351311830095
20: 0.10126539646652896
21: 0.13327392638286895
22: 0.22162372464718408
23: 0.3072949358961183
24: 0.04336135802633999
25: 0.13671973356584255
26: 0.18479356052680876
27: 0.25659147057674553
28: 0.30228017118856665
29: 0.4280598712041204
30: 0.4456225340467613
31: 0.022447086428511166
32: 0.02251471573240699
33: 0.02251801877789944
34: 0.02259811149003198
35: 0.031909218853160864
36: 0.03872149939706565
37: 0.033530786351742066
38: 0.03244085114339997
39: 0.01864469711959618



The chart of threshold of 0.001 just resembles the counterpart of threshold of 0.01 with slight difference. It turns to be smoother and there seems to be no deduction in the end.

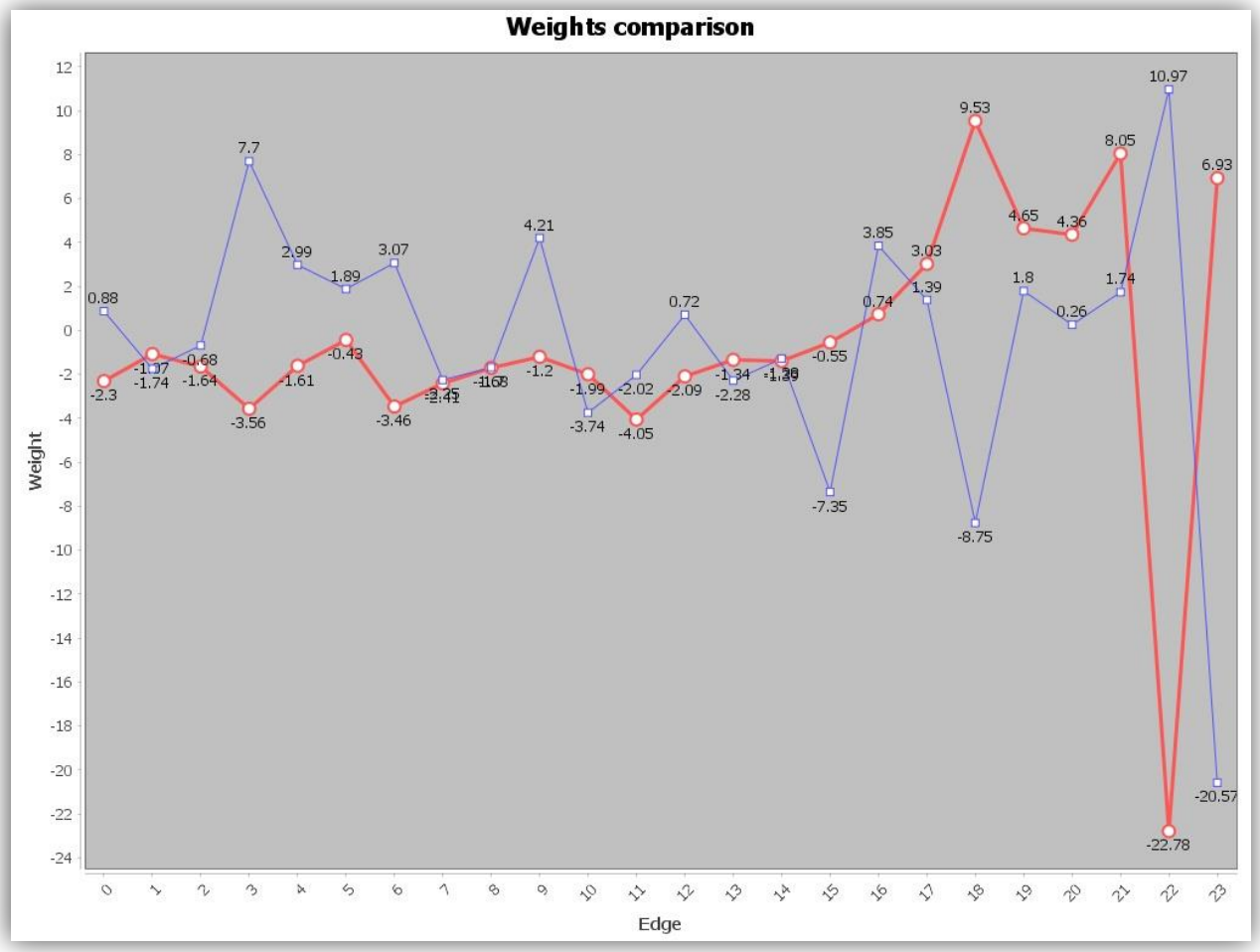
The weights of edges are shown here:
0.8782608557089355, -1.743380771995503, -0.680876621048059, 7.69725590114782, 2.985210976970883, 1.8877990193962224, 3.072688792182264, -2.2497246050484336, -1.6812939656435737, 4.206084210803893, -3.74262647427343, -2.0186935124067555, 0.7154381142372433, -2.279765485828273, -1.2827617249981593, -7.348833171263777, 3.853123086128208, 1.3911331502343023, -8.75192315911962, 1.7963697672327623, 0.2637185657998494, 1.7376931053168096, 10.96651069192475, -20.572450299699547.

Observe that the weights are still random and does not resemble its counterpart of another trial with same threshold.

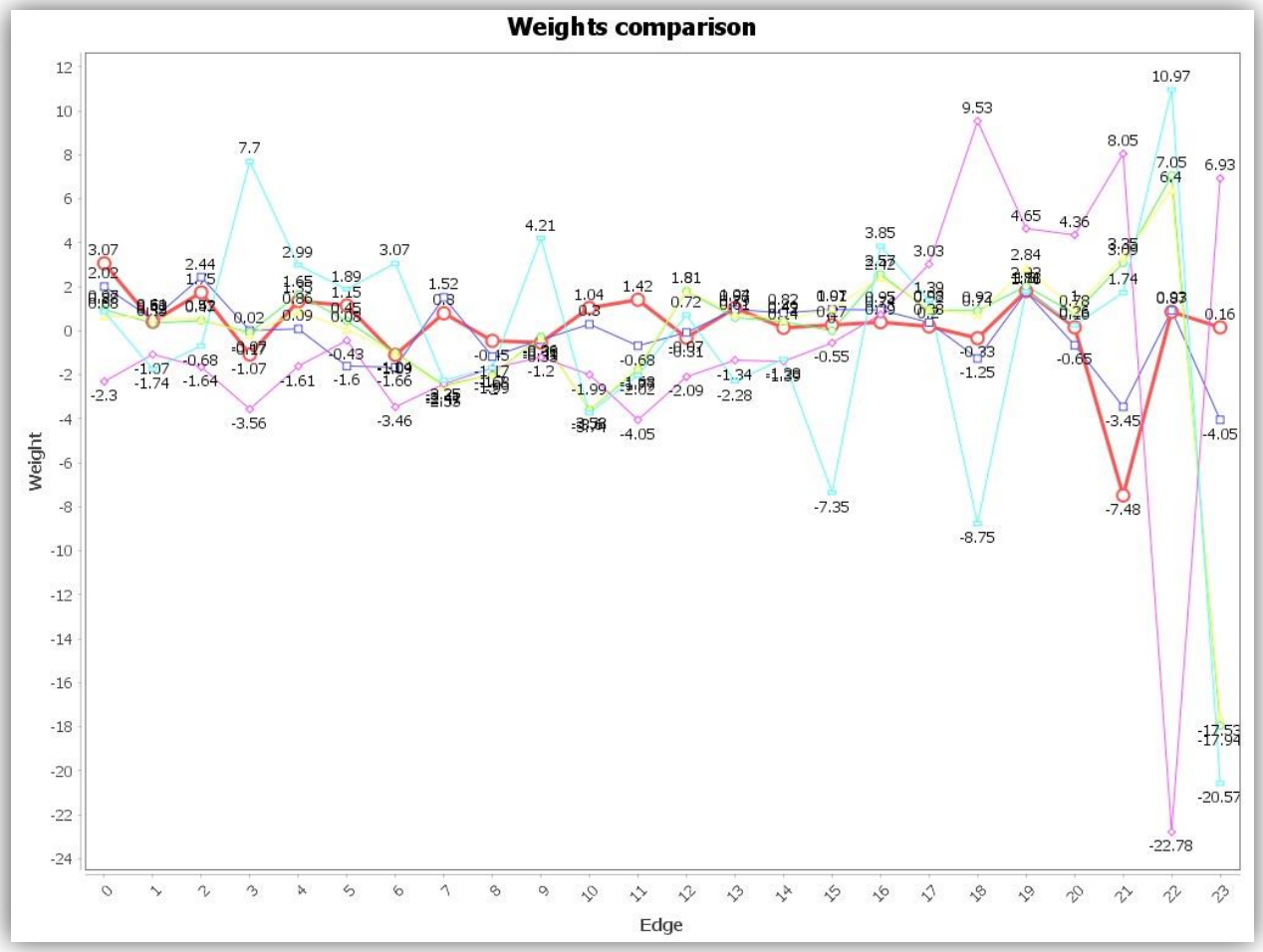
In a nutshell, we could find that when the threshold approaches to being smaller, the curve tends to be smoother and the significant deduction only occurs in the beginning epochs. Although it takes enormous amount of time to compute for the threshold of 0.001 or it ends when approximating

the minimum error, it still provides much accurate results when compared with larger thresholds such as 0.1 and 0.01.

For this pair of experiments, the charts do not show any matching between the two weights, but we still can find they share the same tendency in the chart when the input values go into higher layer of the neural networks.



Lastly, I gather all the weights of experiments with different thresholds and plot them into one chart. As what we observe, the edges between the input layer and hidden layer have similar weights but the edges between hidden layer and output layer can vary significantly.


Weights comparison
Weight
Edge