-0.00436307015621

0.00276729330181

-0.00423322956597

-0.0119653342333

-0.00276925786322

0.00565059059599

-0.00261450133

-0.00330945795109

-0.00274373519003

-0.00604323751733

0.0014044479747

0.00186815214456

-0.00452336394751

-0.00799301353336

-0.0163152087804

-0.0127210096392

-0.0116518570881

-0.0148325409576

-0.00673313215899

-0.0107848991041

-0.00318139121409

-0.0104999390431

-0.00380575966194

-0.00560045604781

0.0050180588146

-0.0181425483613

0.00484775986085

-0.00627683145551

-0.0140359123688

0.00377520931568

-0.0123438397203

-0.00539893799712

-5.39334956716e-05

-0.00315133445495

-0.00840473682772

0.00280836965799

-0.0146559791743

-0.00158304426914

-0.00314108077636

-0.00056180773997

-0.00550852507643

0.00252096150353

0.00737908250807

0.022779683883

0.00300088636746

0.0312034437145

0.0283642646048

0.0147428996958

-0.000171126189213

0.020338971028

0.0196784293309

0.0337556760451

0.0335349242941

0.0126629501752

0.0173737985844

0.0338480208425

0.0213875703823

0.0228757495896

0.0184960839351

0.0210725602601

0.0161004918045

0.0139302286937

0.0279924894128

0.0152912595151

0.00758884707479

0.0349940059274

0.00186197522463

0.00888726857896

0.027415785388

0.0307015768556

0.0237867979467

0.0247092694562

0.0168587686575

0.0128851107162

0.0278485025061

0.00195956431569

-0.00223015776148

0.0286576825991

0.00959647879604

0.00945727461743

-0.0190108166253

0.00264292499471

-0.00166002533284

-0.0175280338577

-0.00588638172751

-0.00647522594023

0.00484040590597

-0.00530231825282

-0.00491056173663

-0.0200330227397

-0.0139543566997

-0.00362541550276

-0.0018539133511

-0.0175087890404

-0.0109802021383

-0.00511772397026

-0.0070342701664

-0.00650306875783

-0.00588284809818

-0.0169944448584

0.0031195508621

-0.00640997744625

-0.00976903841192

-0.0130775872812

-0.000650824351597

-0.0125535209039

-0.010032097132

-0.00635547192737

-0.0146037596993

-0.0159788093844

-0.00936680345195

-0.00865932342907

-0.00467089416383

0.00506440693428

-0.0183421574966

-0.0144015634356

-0.0106671193453

0.0385342911098

0.00113605477652

0.0320188037747

0.0930960018

-1.07413525848

0.071105941586

-0.0418231048415

0.00172599376078

-0.0297184333922

-0.081705947891

1.07135568222

-0.0709963971805

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 230, in Train

perf\_validate = self.Performance(validation\_images)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 180, in Classify

input = self.Convert(image)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 285, in Convert

out.values.append(pixel)

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.27272727 0.300 1.000

2 Performance: 0.27272727 0.300 1.000

3 Performance: 0.27272727 0.300 1.000

4 Performance: 0.27272727 0.300 1.000

5 Performance: 0.27272727 0.300 1.000

6 Performance: 0.27272727 0.300 1.000

7 Performance: 0.27272727 0.300 1.000

8 Performance: 0.27272727 0.300 1.000

9 Performance: 0.27272727 0.300 1.000

10 Performance: 0.27272727 0.300 1.000

11 Performance: 0.27272727 0.300 1.000

12 Performance: 0.27272727 0.300 1.000

13 Performance: 0.27272727 0.300 1.000

14 Performance: 0.27272727 0.300 1.000

15 Performance: 0.27272727 0.300 1.000

16 Performance: 0.27272727 0.300 1.000

17 Performance: 0.27272727 0.300 1.000

18 Performance: 0.27272727 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 230, in Train

perf\_validate = self.Performance(validation\_images)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 181, in Classify

self.FeedForwardFn(self.network, input)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 45, in FeedForward

for i in range(0,len(input.values)):

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse.py > nut\_poi\_new1\_t.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > nut\_poi\_new1\_t.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.19666667 0.300 1.000

2 Performance: 0.19666667 0.300 1.000

3 Performance: 0.19666667 0.300 1.000

4 Performance: 0.19666667 0.300 1.000

5 Performance: 0.19666667 0.300 1.000

6 Performance: 0.19666667 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 111, in Backprop

delta[node] = 0

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.17254902 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 137, in Backprop

network.inputs[m].forward\_weights[j].value += learning\_rate\*network.inputs[m].transformed\_value\*delta[network.inputs[m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 41, in main

images = DataReader.GetImages('training\_new.txt', -1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\data\_reader.py", line 31, in GetImages

image.pixels.append([float(r) for r in line.strip().split()])

ValueError: could not convert string to float: number

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.700 0.000

2 Performance: 0.50000000 0.700 0.000

3 Performance: 0.50000000 0.700 0.000

4 Performance: 0.50000000 0.700 0.000

5 Performance: 0.50000000 0.700 0.000

6 Performance: 0.50000000 0.700 0.000

7 Performance: 0.50000000 0.700 0.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 106, in Backprop

FeedForward(network, input)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 51, in FeedForward

node.raw\_value = NeuralNetwork.ComputeRawValue(node)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 127, in ComputeRawValue

total\_weight += node.weights[i].value \* node.inputs[i].transformed\_value

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t simple

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: SimpleNetwork

Input Nodes: 36, Hidden Nodes: 0, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.300 1.000

2 Performance: 0.50000000 0.300 1.000

3 Performance: 0.50000000 0.300 1.000

4 Performance: 0.50000000 0.300 1.000

5 Performance: 0.49500000 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 230, in Train

perf\_validate = self.Performance(validation\_images)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 181, in Classify

self.FeedForwardFn(self.network, input)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 56, in FeedForward

node.raw\_value = NeuralNetwork.ComputeRawValue(node)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 127, in ComputeRawValue

total\_weight += node.weights[i].value \* node.inputs[i].transformed\_value

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.5 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.500000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.700 0.000

2 Performance: 0.50000000 0.700 0.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 137, in Backprop

network.inputs[m].forward\_weights[j].value += learning\_rate\*network.inputs[m].transformed\_value\*delta[network.inputs[m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.5 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.500000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.300 1.000

2 Performance: 0.50000000 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 230, in Train

perf\_validate = self.Performance(validation\_images)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 181, in Classify

self.FeedForwardFn(self.network, input)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 51, in FeedForward

node.raw\_value = NeuralNetwork.ComputeRawValue(node)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 127, in ComputeRawValue

total\_weight += node.weights[i].value \* node.inputs[i].transformed\_value

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.5 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.500000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

[ 0.49822372 0.49555416]

[ 0.49822542 0.49555214]

[ 0.4982265 0.49555454]

[ 0.49822604 0.49555417]

[ 0.49822692 0.49555267]

[ 0.498224 0.49555388]

[ 0.49822566 0.49555529]

[ 0.49822703 0.49555251]

[ 0.49822303 0.49555336]

[ 0.49822516 0.49555481]

[ 0.49822564 0.49555314]

[ 0.49822495 0.49555375]

[ 0.49822456 0.49555264]

[ 0.49822584 0.49555076]

[ 0.49822477 0.49555325]

[ 0.49822649 0.49555323]

[ 0.49822503 0.49555345]

[ 0.49822697 0.49555448]

[ 0.49822629 0.49555322]

[ 0.49822636 0.49555402]

[ 0.49822568 0.49555325]

[ 0.49822503 0.4955509 ]

[ 0.49822667 0.49555163]

[ 0.4982267 0.49555251]

[ 0.49822683 0.49555365]

[ 0.49822543 0.49555352]

[ 0.49822453 0.49555209]

[ 0.49822899 0.49555133]

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 208, in Train

performance\_log.append((self.Performance(images), self.Performance(validation\_images)))

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 182, in Classify

return self.GetNetworkLabel()

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 251, in GetNetworkLabel

print output\_labels

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.5 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.500000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

[0.77, 0.73, 0.81, 0.67, 0.24, 0.32, 0.68, 0.73, 0.82, 0.64, 0.71, 0.73, 0.79, 0.86, 0.75, 0.61, 0.22, 0.16, 0.75, 0.21, 0.25, 0.76, 0.32, 0.38, 0.59, 0.66, 0.73, 0.6, 0.59, 0.56, 0.57, 0.26, 0.1, 0.7

1, 0.34, 0.15]

[0.77, 0.65, 0.75, 0.66, 0.38, 0.39, 0.73, 0.76, 0.78, 0.6, 0.68, 0.68, 0.74, 0.73, 0.76, 0.45, 0.28, 0.22, 0.67, 0.29, 0.38, 0.7, 0.39, 0.48, 0.49, 0.69, 0.71, 0.55, 0.65, 0.66, 0.57, 0.36, 0.18, 0.5

8, 0.3, 0.21]

[0.71, 0.77, 0.76, 0.69, 0.35, 0.34, 0.8, 0.75, 0.74, 0.52, 0.74, 0.73, 0.78, 0.71, 0.75, 0.55, 0.27, 0.22, 0.77, 0.34, 0.36, 0.83, 0.35, 0.47, 0.63, 0.79, 0.65, 0.54, 0.58, 0.59, 0.45, 0.2, 0.16, 0.4

7, 0.35, 0.2]

[0.79, 0.77, 0.74, 0.68, 0.32, 0.28, 0.81, 0.75, 0.74, 0.63, 0.71, 0.71, 0.72, 0.74, 0.76, 0.49, 0.26, 0.17, 0.64, 0.32, 0.33, 0.79, 0.37, 0.41, 0.58, 0.72, 0.72, 0.54, 0.65, 0.58, 0.41, 0.22, 0.15, 0

.55, 0.33, 0.24]

[0.7, 0.65, 0.75, 0.65, 0.34, 0.4, 0.71, 0.68, 0.81, 0.5, 0.72, 0.79, 0.74, 0.75, 0.78, 0.55, 0.27, 0.16, 0.71, 0.36, 0.42, 0.73, 0.4, 0.48, 0.59, 0.76, 0.75, 0.5, 0.67, 0.51, 0.47, 0.21, 0.19, 0.51,

0.31, 0.2]

[0.84, 0.79, 0.77, 0.81, 0.23, 0.26, 0.75, 0.77, 0.87, 0.58, 0.8, 0.79, 0.84, 0.79, 0.79, 0.56, 0.29, 0.1, 0.76, 0.19, 0.2, 0.85, 0.3, 0.34, 0.58, 0.77, 0.8, 0.6, 0.5, 0.58, 0.56, 0.2, 0.12, 0.68, 0.2

6, 0.11]

[0.79, 0.72, 0.65, 0.65, 0.32, 0.34, 0.75, 0.74, 0.73, 0.55, 0.78, 0.73, 0.64, 0.69, 0.73, 0.49, 0.22, 0.17, 0.62, 0.27, 0.31, 0.72, 0.43, 0.41, 0.55, 0.79, 0.74, 0.54, 0.7, 0.6, 0.45, 0.18, 0.11, 0.5

5, 0.36, 0.14]

[0.81, 0.81, 0.83, 0.72, 0.32, 0.38, 0.76, 0.7, 0.76, 0.53, 0.72, 0.7, 0.73, 0.69, 0.75, 0.52, 0.26, 0.13, 0.62, 0.36, 0.32, 0.79, 0.36, 0.44, 0.49, 0.71, 0.7, 0.58, 0.58, 0.48, 0.39, 0.2, 0.21, 0.6,

0.39, 0.2]

[0.81, 0.75, 0.76, 0.65, 0.21, 0.25, 0.78, 0.76, 0.81, 0.56, 0.76, 0.76, 0.82, 0.77, 0.84, 0.59, 0.31, 0.11, 0.73, 0.22, 0.28, 0.8, 0.18, 0.22, 0.52, 0.81, 0.79, 0.61, 0.49, 0.56, 0.54, 0.23, 0.13, 0.

63, 0.36, 0.19]

[0.76, 0.83, 0.75, 0.76, 0.28, 0.32, 0.75, 0.79, 0.82, 0.55, 0.83, 0.76, 0.8, 0.77, 0.8, 0.69, 0.31, 0.13, 0.69, 0.19, 0.26, 0.75, 0.23, 0.36, 0.59, 0.71, 0.74, 0.52, 0.55, 0.56, 0.68, 0.25, 0.15, 0.6

5, 0.31, 0.15]

[0.75, 0.75, 0.76, 0.67, 0.44, 0.39, 0.72, 0.77, 0.75, 0.55, 0.76, 0.75, 0.74, 0.66, 0.71, 0.5, 0.26, 0.15, 0.72, 0.42, 0.38, 0.68, 0.38, 0.41, 0.55, 0.73, 0.69, 0.54, 0.66, 0.64, 0.49, 0.25, 0.24, 0.

59, 0.34, 0.18]

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 203, in Train

inputs.append(self.Convert(image))

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 286, in Convert

print out.values

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > training\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

[0.77, 0.73, 0.81, 0.67, 0.24, 0.32, 0.68, 0.73, 0.82, 0.64, 0.71, 0.73, 0.79, 0.86, 0.75, Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 203, in Train

inputs.append(self.Convert(image))

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 286, in Convert

print out.values

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.700 0.000

2 Performance: 0.50000000 0.300 1.000

3 Performance: 0.50000000 0.300 1.000

4 Performance: 0.50000000 0.300 1.000

5 Performance: 0.50000000 0.300 1.000

6 Performance: 0.50000000 0.300 1.000

7 Performance: 0.50000000 0.300 1.000

8 Performance: 0.50000000 0.300 1.000

9 Performance: 0.16750000 0.300 1.000

10 Performance: 0.04750000 0.300 1.000

11 Performance: 0.01750000 0.300 1.000

12 Performance: 0.01750000 0.300 1.000

13 Performance: 0.02000000 0.300 1.000

14 Performance: 0.01000000 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 140, in Backprop

for j in range(0,len(network.hidden\_nodes[m].forward\_neighbors)):

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 10 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 10.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.700 0.000

2 Performance: 0.50000000 0.700 0.000

3 Performance: 0.50000000 0.300 1.000

4 Performance: 0.50000000 0.300 1.000

5 Performance: 0.50000000 0.300 1.000

6 Performance: 0.50000000 0.300 1.000

7 Performance: 0.50000000 0.300 1.000

8 Performance: 0.50000000 0.300 1.000

9 Performance: 0.50000000 0.300 1.000

10 Performance: 0.50000000 0.300 1.000

11 Performance: 0.50000000 0.300 1.000

12 Performance: 0.50000000 0.300 1.000

13 Performance: 0.50000000 0.300 1.000

14 Performance: 0.50000000 0.300 1.000

15 Performance: 0.50000000 0.300 1.000

16 Performance: 0.50000000 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 129, in Backprop

for j in range(0,len(network.inputs[-m].forward\_neighbors)):

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.3 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.300000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.300 1.000

2 Performance: 0.50000000 0.300 1.000

3 Performance: 0.50000000 0.300 1.000

4 Performance: 0.50000000 0.300 1.000

5 Performance: 0.50000000 0.300 1.000

6 Performance: 0.50000000 0.300 1.000

7 Performance: 0.50000000 0.300 1.000

8 Performance: 0.50000000 0.300 1.000

9 Performance: 0.50000000 0.300 1.000

10 Performance: 0.50000000 0.300 1.000

11 Performance: 0.50000000 0.300 1.000

12 Performance: 0.50000000 0.300 1.000

13 Performance: 0.50000000 0.300 1.000

14 Performance: 0.50000000 0.300 1.000

15 Performance: 0.50000000 0.300 1.000

16 Performance: 0.50000000 0.300 1.000

17 Performance: 0.50000000 0.300 1.000

18 Performance: 0.50000000 0.300 1.000

19 Performance: 0.50000000 0.300 1.000

20 Performance: 0.50000000 0.300 1.000

21 Performance: 0.50000000 0.300 1.000

22 Performance: 0.50000000 0.300 1.000

23 Performance: 0.50000000 0.300 1.000

24 Performance: 0.50000000 0.300 1.000

25 Performance: 0.50000000 0.300 1.000

26 Performance: 0.50000000 0.300 1.000

27 Performance: 0.50000000 0.300 1.000

28 Performance: 0.50000000 0.300 1.000

29 Performance: 0.50000000 0.300 1.000

30 Performance: 0.50000000 0.300 1.000

31 Performance: 0.50000000 0.300 1.000

32 Performance: 0.50000000 0.300 1.000

33 Performance: 0.50000000 0.300 1.000

34 Performance: 0.50000000 0.300 1.000

35 Performance: 0.50000000 0.300 1.000

36 Performance: 0.50000000 0.300 1.000

37 Performance: 0.50000000 0.300 1.000

38 Performance: 0.50000000 0.300 1.000

39 Performance: 0.50000000 0.300 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 137, in Backprop

network.inputs[m].forward\_weights[j].value += learning\_rate\*network.inputs[m].transformed\_value\*delta[network.inputs[m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > validation\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.3 -t hidden

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 47, in main

validation = DataReader.GetImages('validation\_new.txt', -1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\data\_reader.py", line 31, in GetImages

image.pixels.append([float(r) for r in line.strip().split()])

ValueError: could not convert string to float: number

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.3 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.300000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.714 1.000

2 Performance: 0.50000000 0.714 1.000

3 Performance: 0.50000000 0.714 1.000

4 Performance: 0.50000000 0.286 1.000

5 Performance: 0.50000000 0.286 1.000

6 Performance: 0.50000000 0.286 1.000

7 Performance: 0.50000000 0.286 1.000

8 Performance: 0.50000000 0.286 1.000

9 Performance: 0.50000000 0.286 1.000

10 Performance: 0.50000000 0.286 1.000

11 Performance: 0.50000000 0.286 1.000

12 Performance: 0.50000000 0.286 1.000

13 Performance: 0.50000000 0.286 1.000

14 Performance: 0.50000000 0.286 1.000

15 Performance: 0.50000000 0.286 1.000

16 Performance: 0.50000000 0.286 1.000

17 Performance: 0.01500000 0.018 1.000

18 Performance: 0.04500000 0.093 1.000

19 Performance: 0.02000000 0.039 1.000

20 Performance: 0.01500000 0.029 1.000

21 Performance: 0.01250000 0.021 1.000

22 Performance: 0.00750000 0.014 1.000

23 Performance: 0.00500000 0.014 1.000

24 Performance: 0.00500000 0.011 1.000

25 Performance: 0.00500000 0.007 1.000

26 Performance: 0.00500000 0.007 1.000

27 Performance: 0.00250000 0.007 1.000

28 Performance: 0.00250000 0.007 1.000

29 Performance: 0.00250000 0.007 1.000

30 Performance: 0.00000000 0.007 1.000

31 Performance: 0.00000000 0.007 1.000

32 Performance: 0.00000000 0.007 1.000

33 Performance: 0.00000000 0.007 1.000

34 Performance: 0.00000000 0.004 1.000

35 Performance: 0.00000000 0.004 1.000

36 Performance: 0.00000000 0.004 1.000

37 Performance: 0.00000000 0.004 1.000

38 Performance: 0.00000000 0.004 1.000

39 Performance: 0.00000000 0.004 1.000

40 Performance: 0.00000000 0.004 1.000

41 Performance: 0.00000000 0.004 1.000

42 Performance: 0.00000000 0.004 1.000

43 Performance: 0.00000000 0.004 1.000

44 Performance: 0.00000000 0.004 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 130, in Backprop

e\_m += network.inputs[-m].forward\_weights[j].value\*delta[network.inputs[-m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 0.3 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 0.300000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 208, in Train

performance\_log.append((self.Performance(images), self.Performance(validation\_images)))

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 193, in Performance

if self.Classify(image) == label:

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 181, in Classify

self.FeedForwardFn(self.network, input)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 56, in FeedForward

node.raw\_value = NeuralNetwork.ComputeRawValue(node)

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.286 1.000

2 Performance: 0.50000000 0.286 0.000

3 Performance: 0.50000000 0.714 0.000

4 Performance: 0.50000000 0.714 0.000

5 Performance: 0.50000000 0.714 0.000

6 Performance: 0.50000000 0.286 0.000

7 Performance: 0.50000000 0.286 0.000

8 Performance: 0.56250000 0.371 0.000

9 Performance: 0.90000000 0.854 0.000

10 Performance: 0.97250000 0.946 0.000

11 Performance: 0.98250000 0.968 0.000

12 Performance: 0.98500000 0.975 0.000

13 Performance: 0.98500000 0.979 0.000

14 Performance: 0.98500000 0.979 0.000

15 Performance: 0.99500000 0.986 0.000

16 Performance: 0.98500000 0.968 0.000

17 Performance: 0.96000000 0.936 0.000

18 Performance: 0.96750000 0.946 0.000

19 Performance: 0.96000000 0.932 0.000

20 Performance: 0.99000000 0.982 0.000

21 Performance: 0.95500000 0.911 0.000

22 Performance: 0.83750000 0.757 0.000

23 Performance: 0.90500000 0.857 0.000

24 Performance: 0.86000000 0.793 0.000

25 Performance: 1.00000000 0.996 0.000

26 Performance: 1.00000000 0.996 0.000

27 Performance: 1.00000000 0.996 0.000

28 Performance: 0.99500000 0.982 0.000

29 Performance: 0.66250000 0.496 0.000

30 Performance: 0.95750000 0.907 0.000

31 Performance: 0.97750000 0.957 0.000

32 Performance: 0.96250000 0.932 0.000

33 Performance: 0.71250000 0.571 0.000

34 Performance: 0.64250000 0.457 0.000

35 Performance: 0.64250000 0.464 0.000

36 Performance: 0.65750000 0.489 0.000

37 Performance: 0.69000000 0.518 0.000

38 Performance: 0.77500000 0.621 0.000

39 Performance: 0.88250000 0.811 0.000

40 Performance: 0.98750000 0.982 0.000

41 Performance: 0.99750000 0.993 0.000

42 Performance: 1.00000000 0.996 0.000

43 Performance: 1.00000000 1.000 0.000

44 Performance: 1.00000000 0.996 0.000

45 Performance: 1.00000000 0.996 0.000

46 Performance: 1.00000000 0.993 0.000

47 Performance: 1.00000000 1.000 0.000

48 Performance: 1.00000000 1.000 0.000

49 Performance: 1.00000000 1.000 0.000

50 Performance: 1.00000000 1.000 0.000

51 Performance: 1.00000000 1.000 0.000

52 Performance: 1.00000000 1.000 0.000

53 Performance: 1.00000000 1.000 0.000

54 Performance: 1.00000000 1.000 0.000

55 Performance: 1.00000000 1.000 0.000

56 Performance: 1.00000000 1.000 0.000

57 Performance: 1.00000000 1.000 0.000

58 Performance: 1.00000000 1.000 0.000

59 Performance: 1.00000000 1.000 0.000

60 Performance: 1.00000000 1.000 0.000

61 Performance: 1.00000000 1.000 0.000

62 Performance: 1.00000000 1.000 0.000

63 Performance: 1.00000000 1.000 0.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 137, in Backprop

network.inputs[m].forward\_weights[j].value += learning\_rate\*network.inputs[m].transformed\_value\*delta[network.inputs[m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > test\_new.txt

Traceback (most recent call last):

File "parse2.py", line 4, in <module>

infile = open(filename, 'r')

IOError: [Errno 2] No such file or directory: 'nut\_poi\_new6.txt'

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > test\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.286 0.216

2 Performance: 0.50000000 0.286 0.216

3 Performance: 0.50000000 0.714 0.784

4 Performance: 0.50000000 0.714 0.784

5 Performance: 0.50000000 0.714 0.784

6 Performance: 0.50000000 0.714 0.784

7 Performance: 0.50000000 0.286 0.216

8 Performance: 0.54750000 0.361 0.302

9 Performance: 0.90000000 0.850 0.827

10 Performance: 0.97750000 0.954 0.937

11 Performance: 0.98000000 0.957 0.957

12 Performance: 0.98000000 0.957 0.949

13 Performance: 0.99000000 0.982 0.992

14 Performance: 0.96000000 0.943 0.914

15 Performance: 0.99500000 0.989 1.000

16 Performance: 0.98000000 0.961 0.957

17 Performance: 0.83750000 0.757 0.678

18 Performance: 0.95000000 0.896 0.886

19 Performance: 0.96000000 0.943 0.918

20 Performance: 0.70000000 0.536 0.459

21 Performance: 0.99750000 0.993 1.000

22 Performance: 0.54750000 0.357 0.290

23 Performance: 0.97750000 0.954 0.949

24 Performance: 1.00000000 0.996 1.000

25 Performance: 0.52000000 0.329 0.247

26 Performance: 0.97750000 0.957 0.957

27 Performance: 0.99750000 0.993 1.000

28 Performance: 0.98500000 0.975 0.969

29 Performance: 1.00000000 0.993 1.000

30 Performance: 0.98250000 0.964 0.965

31 Performance: 0.99500000 0.986 0.996

32 Performance: 1.00000000 0.993 1.000

33 Performance: 1.00000000 0.996 1.000

34 Performance: 1.00000000 0.996 1.000

35 Performance: 1.00000000 1.000 1.000

36 Performance: 1.00000000 1.000 1.000

37 Performance: 0.98500000 0.975 0.976

38 Performance: 1.00000000 0.996 1.000

39 Performance: 1.00000000 1.000 1.000

40 Performance: 1.00000000 1.000 1.000

41 Performance: 1.00000000 1.000 1.000

42 Performance: 1.00000000 0.996 1.000

43 Performance: 1.00000000 1.000 1.000

44 Performance: 1.00000000 0.996 1.000

45 Performance: 1.00000000 0.996 1.000

46 Performance: 1.00000000 0.996 1.000

47 Performance: 1.00000000 0.996 1.000

48 Performance: 0.65500000 0.504 0.408

49 Performance: 1.00000000 0.996 1.000

50 Performance: 1.00000000 0.996 1.000

51 Performance: 1.00000000 0.996 1.000

52 Performance: 1.00000000 0.996 1.000

53 Performance: 1.00000000 0.996 1.000

54 Performance: 1.00000000 0.996 1.000

55 Performance: 1.00000000 1.000 1.000

56 Performance: 1.00000000 0.996 1.000

57 Performance: 1.00000000 1.000 1.000

58 Performance: 1.00000000 0.996 1.000

59 Performance: 1.00000000 1.000 1.000

60 Performance: 1.00000000 0.996 1.000

61 Performance: 1.00000000 0.996 1.000

Traceback (most recent call last):

File "neural\_net\_main.py", line 84, in <module>

main()

File "neural\_net\_main.py", line 81, in main

network.Train(images, validation, test, rate, epochs)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net.py", line 218, in Train

self.TrainFn(self.network, inputs, targets, learning\_rate, 1)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 169, in Train

Backprop(network, inputs[j], targets[j], learning\_rate)

File "C:\Users\annie\_000\Documents\GitHub\CS181\_Proj\neural\_net\_impl.py", line 130, in Backprop

e\_m += network.inputs[-m].forward\_weights[j].value\*delta[network.inputs[-m].forward\_neighbors[j]]

KeyboardInterrupt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > test\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python parse2.py > validation\_new.txt

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>python neural\_net\_main.py -e 100 -r 1.0 -t hidden

weights len: 197

\* \* \* \* \* \* \* \* \*

Parameters => Epochs: 100, Learning Rate: 1.000000

Type of network used: HiddenNetwork

Input Nodes: 36, Hidden Nodes: 5, Output Nodes: 2

\* \* \* \* \* \* \* \* \*

1 Performance: 0.50000000 0.500 0.500

2 Performance: 0.50000000 0.500 0.500

3 Performance: 0.50000000 0.500 0.500

4 Performance: 0.50000000 0.500 0.500

5 Performance: 0.50000000 0.500 0.500

6 Performance: 0.50000000 0.500 0.500

7 Performance: 0.50000000 0.500 0.500

8 Performance: 0.82500000 0.794 0.764

9 Performance: 0.95250000 0.931 0.927

10 Performance: 0.98000000 0.975 0.973

11 Performance: 0.98250000 0.975 0.982

12 Performance: 0.98500000 0.981 0.991

13 Performance: 0.98500000 0.981 0.991

14 Performance: 0.98500000 0.988 0.991

15 Performance: 0.98500000 0.981 0.991

16 Performance: 0.98250000 0.975 0.973

17 Performance: 0.98500000 0.988 0.991

18 Performance: 0.80500000 0.750 0.755

19 Performance: 0.89500000 0.869 0.864

20 Performance: 0.96750000 0.963 0.955

21 Performance: 1.00000000 1.000 1.000

22 Performance: 0.78750000 0.738 0.718

23 Performance: 0.95500000 0.931 0.918

24 Performance: 1.00000000 1.000 1.000

25 Performance: 1.00000000 1.000 1.000

26 Performance: 1.00000000 1.000 1.000

27 Performance: 0.67500000 0.637 0.591

28 Performance: 0.82250000 0.775 0.764

29 Performance: 0.96000000 0.956 0.927

30 Performance: 0.99500000 0.988 1.000

31 Performance: 0.98750000 0.988 0.991

32 Performance: 0.90250000 0.875 0.882

33 Performance: 0.88500000 0.856 0.836

34 Performance: 0.97000000 0.963 0.964

35 Performance: 0.98000000 0.975 0.982

36 Performance: 0.99500000 0.988 1.000

37 Performance: 0.90500000 0.875 0.891

38 Performance: 1.00000000 1.000 1.000

39 Performance: 0.96250000 0.956 0.936

40 Performance: 1.00000000 1.000 1.000

41 Performance: 1.00000000 1.000 1.000

42 Performance: 1.00000000 1.000 1.000

43 Performance: 1.00000000 1.000 1.000

44 Performance: 1.00000000 1.000 1.000

45 Performance: 1.00000000 1.000 1.000

46 Performance: 1.00000000 1.000 1.000

47 Performance: 1.00000000 1.000 1.000

48 Performance: 1.00000000 1.000 1.000

49 Performance: 1.00000000 1.000 1.000

50 Performance: 1.00000000 1.000 1.000

51 Performance: 1.00000000 1.000 1.000

52 Performance: 1.00000000 1.000 1.000

53 Performance: 1.00000000 1.000 1.000

54 Performance: 1.00000000 1.000 1.000

55 Performance: 1.00000000 1.000 1.000

56 Performance: 1.00000000 1.000 1.000

57 Performance: 1.00000000 1.000 1.000

58 Performance: 1.00000000 1.000 1.000

59 Performance: 1.00000000 1.000 1.000

60 Performance: 1.00000000 1.000 1.000

61 Performance: 1.00000000 1.000 1.000

62 Performance: 1.00000000 1.000 1.000

63 Performance: 1.00000000 1.000 1.000

64 Performance: 1.00000000 1.000 1.000

65 Performance: 1.00000000 1.000 1.000

66 Performance: 1.00000000 1.000 1.000

67 Performance: 1.00000000 1.000 1.000

68 Performance: 1.00000000 1.000 1.000

69 Performance: 1.00000000 1.000 1.000

70 Performance: 1.00000000 1.000 1.000

71 Performance: 1.00000000 1.000 1.000

72 Performance: 1.00000000 1.000 1.000

73 Performance: 1.00000000 1.000 1.000

74 Performance: 1.00000000 1.000 1.000

75 Performance: 1.00000000 1.000 1.000

76 Performance: 1.00000000 1.000 1.000

77 Performance: 1.00000000 1.000 1.000

78 Performance: 1.00000000 1.000 1.000

79 Performance: 1.00000000 1.000 1.000

80 Performance: 1.00000000 1.000 1.000

81 Performance: 1.00000000 1.000 1.000

82 Performance: 1.00000000 1.000 1.000

83 Performance: 1.00000000 1.000 1.000

84 Performance: 1.00000000 1.000 1.000

85 Performance: 1.00000000 1.000 1.000

86 Performance: 1.00000000 1.000 1.000

87 Performance: 1.00000000 1.000 1.000

88 Performance: 1.00000000 1.000 1.000

89 Performance: 1.00000000 1.000 1.000

90 Performance: 1.00000000 1.000 1.000

91 Performance: 1.00000000 1.000 1.000

92 Performance: 1.00000000 1.000 1.000

93 Performance: 1.00000000 1.000 1.000

94 Performance: 1.00000000 1.000 1.000

95 Performance: 1.00000000 1.000 1.000

96 Performance: 1.00000000 1.000 1.000

97 Performance: 1.00000000 1.000 1.000

98 Performance: 1.00000000 1.000 1.000

99 Performance: 1.00000000 1.000 1.000

100 Performance: 1.00000000 1.000 1.000

C:\Users\annie\_000\Documents\GitHub\CS181\_Proj>