COMP9444

Project 1 - Characters, Spirals and Hidden Unit Dynamics

Yuxuan Huang z5274414

Part 1: Japanese Character Recognition

1.

```
<class 'numpy.ndarray'>
[[770.   5.   8.  12.  30.  64.   2.  62.  29.  18.]
 [  7. 665. 110.  18.  32.  22.  57.  15.  26.  48.]
 [  6.  60. 696.  27.  26.  20.  45.  37.  45.  38.]
 [  5.  34.  61. 757.  14.  57.  14.  18.  29.  11.]
 [ 59.  50.  82.  20. 624.  20.  32.  37.  19.  57.]
 [  7.  28. 124.  17.  20. 725.  27.   8.  34.  10.]
 [  5.  24. 147.  10.  26.  23. 723.  20.   9.  13.]
 [ 14.  29.  27.  10.  88.  18.  55. 622.  89.  48.]
 [ 12.  36.  93.  40.   5.  30.  44.   7. 712.  21.]
 [  8.  50.  87.   4.  56.  30.  19.  31.  41. 674.]]

Test set: Average loss: 1.0092, Accuracy: 6968/10000 (70%)
```

2.

```
[[857.   5.   2.   8.  31.  24.   4.  38.  26.   5.]
 [  4. 810.  32.   3.  21.  14.  59.   8.  18.  31.]
 [  7.  11. 844.  39.  11.  16.  27.  14.  19.  12.]
 [  3.   9.  28. 924.   1.  11.   7.   3.   6.   8.]
 [ 35.  27.  20.   3. 828.   7.  30.  16.  19.  15.]
 [ 10.  13.  72.  13.  14. 832.  23.   1.  15.   7.]
 [  3.   9.  52.  10.  16.   7. 888.   5.   2.   8.]
 [ 18.  11.  21.   6.  17.   8.  28. 826.  28.  37.]
 [ 11.  22.  28.  55.   4.  10.  33.   3. 826.   8.]
 [  6.  15.  50.   6.  30.   3.  16.  21.  10. 843.]]

Test set: Average loss: 0.5050, Accuracy: 8478/10000 (85%)
```

3.

```
<class 'numpy.ndarray'>
[[959.   4.   1.   0.  20.   2.   0.  10.   1.   3.]
 [  1. 938.   2.   0.   9.   2.  32.   5.   2.   9.]
 [ 12.   9. 885.  36.   6.  14.  18.  10.   4.   6.]
 [  1.   0.   9. 970.   0.   6.   7.   3.   0.   4.]
 [ 16.   9.   2.   5. 941.   1.   9.   4.   9.   4.]
 [  4.   8.  15.   4.   5. 931.  21.   1.   3.   8.]
 [  2.   1.   7.   0.   4.   3. 980.   1.   0.   2.]
 [ 10.   2.   1.   0.  12.   0.   5. 943.   4.  23.]
 [  4.   9.   4.   1.   8.   1.  10.   5. 954.   4.]
 [  5.   4.   5.   1.   4.   0.   3.   2.   1. 975.]]

Test set: Average loss: 0.2227, Accuracy: 9476/10000 (95%)
```

4.

a. the accuracy of single linear network is the lowest which is 70% and that of fully connected 2-layer network is better which is 85% and convolutional network performances best which is 95%. So the convolutional network is the most suitable for image recognition and it's better than single linear and full-connected network.

b. The confusion matrix for Net lin: the max number except the diagonal is 147 in the 7th row and the character is "ma" most likely to be mistaken for "su".

The confusion matrix for Net lin: the max number except the diagonal is 72 in the 6th row and the character is "ha" most likely to be mistaken for "su".

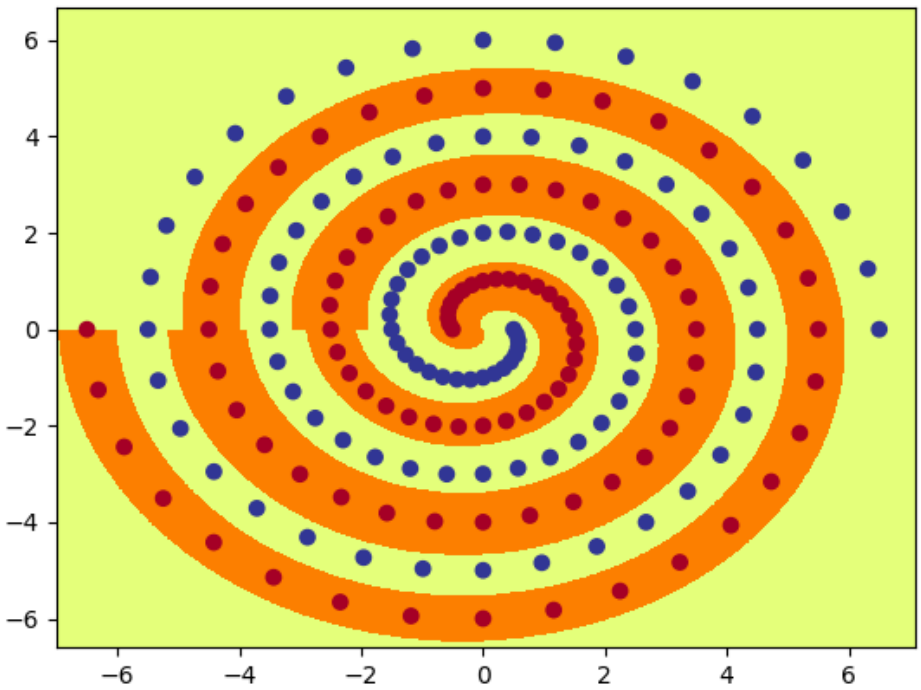
The confusion matrix for Net full: the max number except the diagonal is 32 in the 2nd row and the character is "ki" most likely to be mistaken for "ma".

The reason is their handwriting and visual structure is very similar.

c. I changed the values for momentum for Netfull. The values I changed from 0.1 to 0.9, and I found that the bigger moment it is, the higher accuracy the network can reach. The highest accuracy is 88% if the momentum is 0.9.

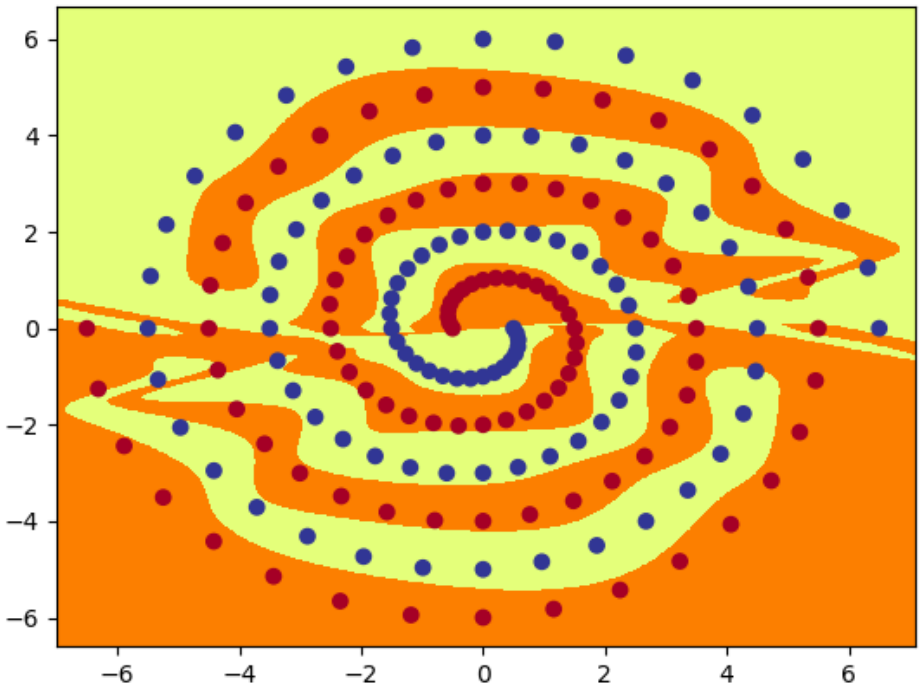
Part 2: Twin Spirals Task

1. See the codes in spiral.py
2. First I tried hidden node is 10 and it successed and then I decrease hidden node from 10 to 6 step by step and finally found if hidden node is 6, it failed to reach the goal. So the minimum hidden node is 7. Here is the polar_out.png.

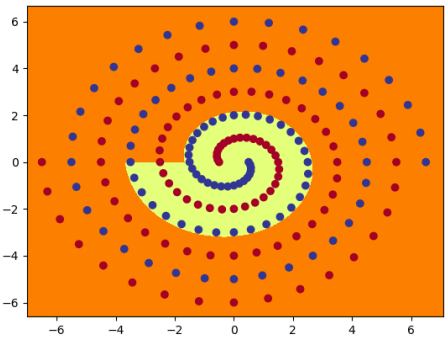


3. See the codes in spiral.py

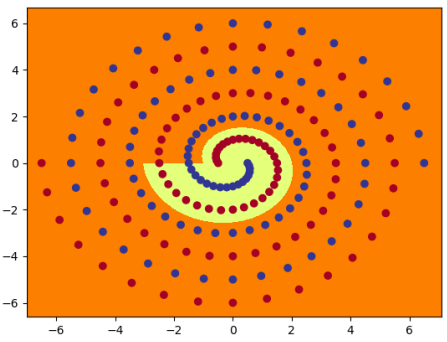
4. Hidden node = 10 initial weight = 0.17



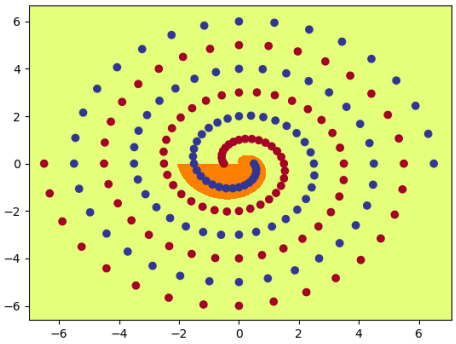
5. PolarNet: hidden node is 7



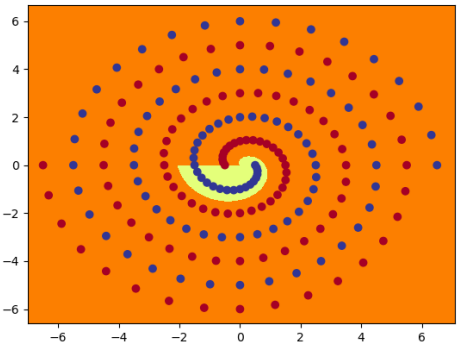
polar1_0.png



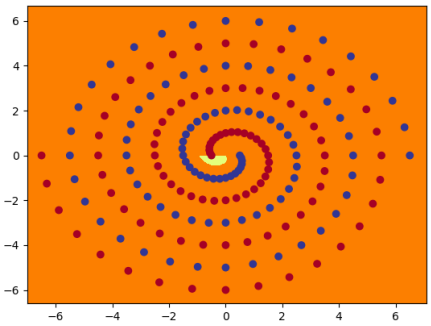
polar1_1.png



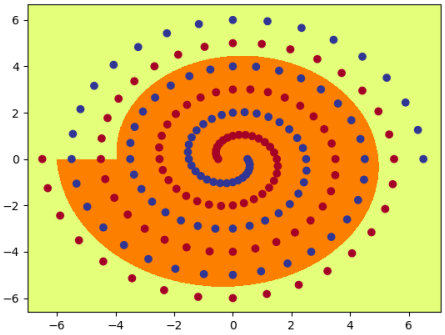
polar1_2.png



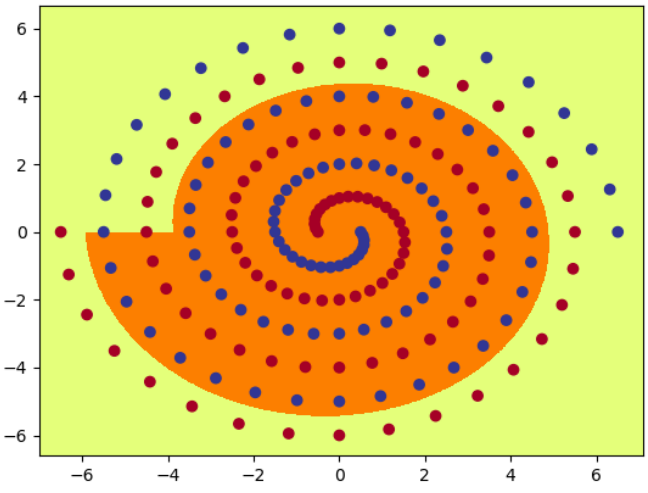
polar1_3.png



polar1_4.png



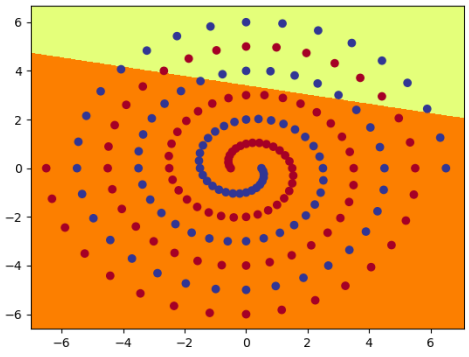
polar1_5png



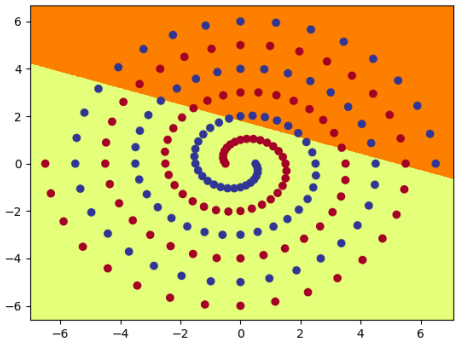
polar1_6.png

RawNet: hidden node is 10 and initial weight = 0.17

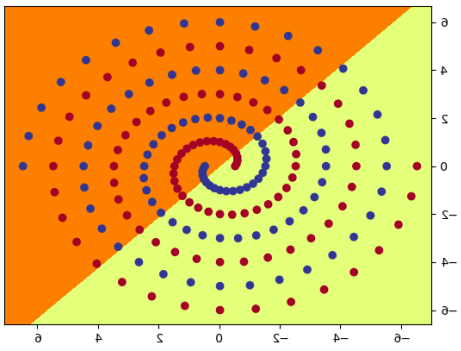
Hidden layer 1:



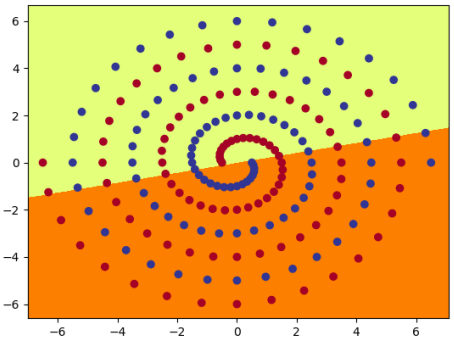
raw1_0.png



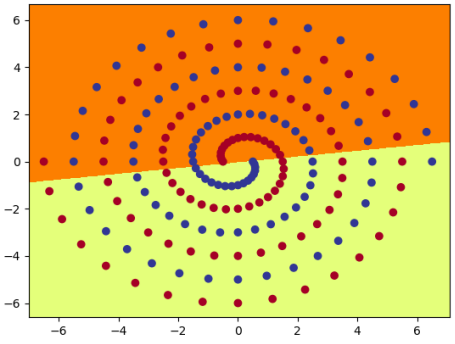
raw1_1.png



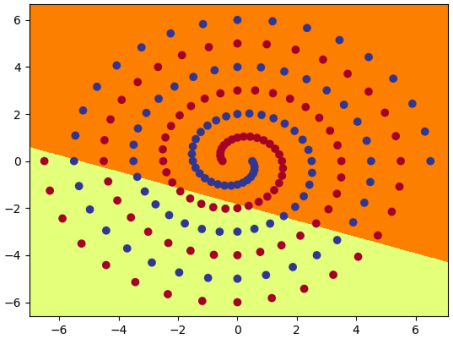
raw1_2.png



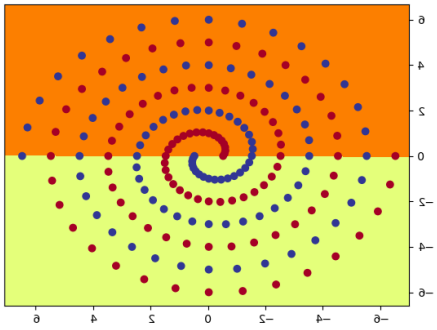
raw1_3.png



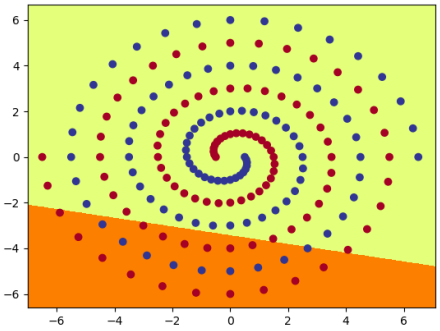
raw1_4.png



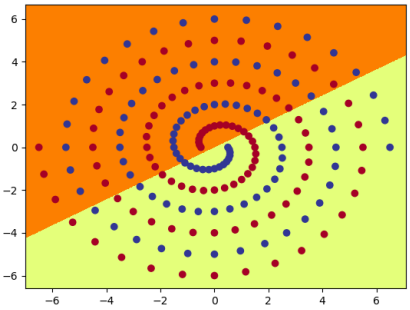
raw1_5.png



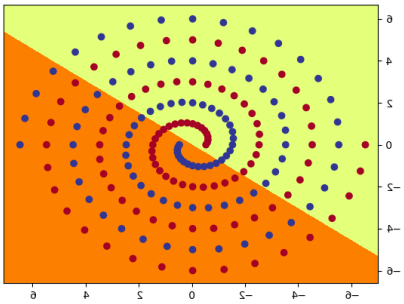
raw1_6.png



raw1_7.png

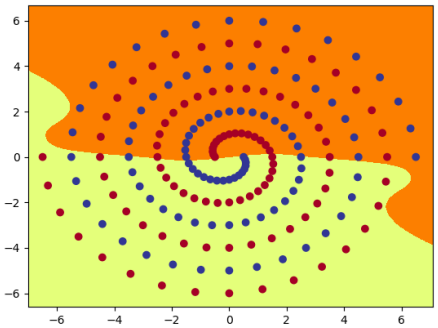


raw1_8.png

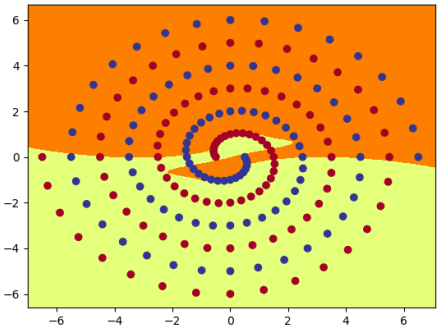


raw1_9.png

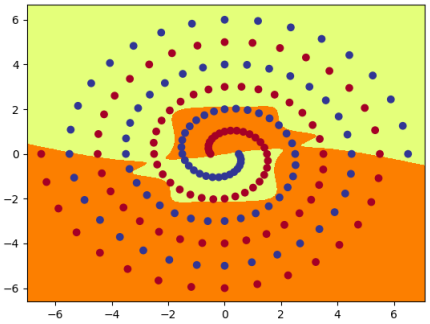
Hidden layer 2:



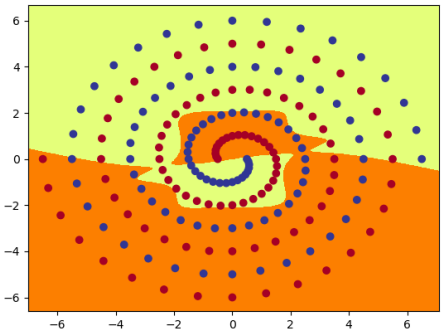
raw2_0.png



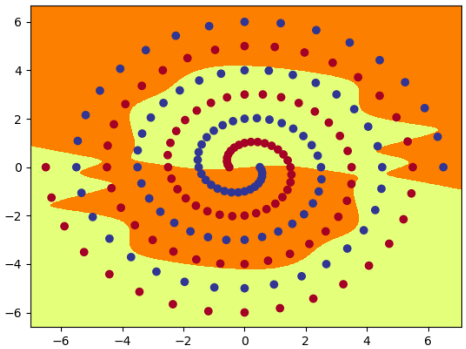
raw2_1.png



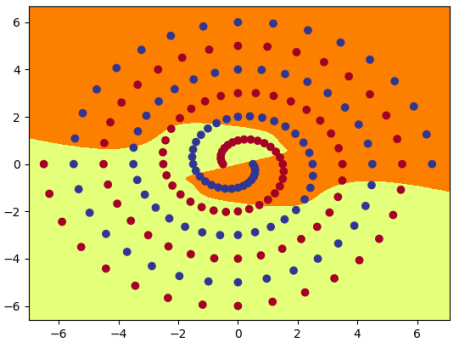
raw2_2.png



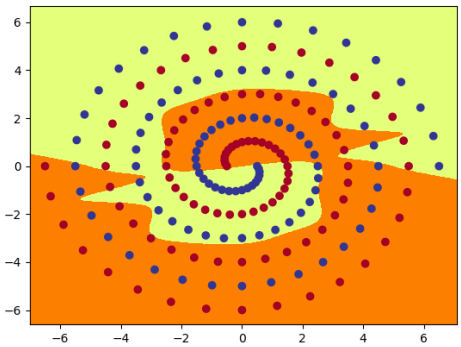
raw2_3.png



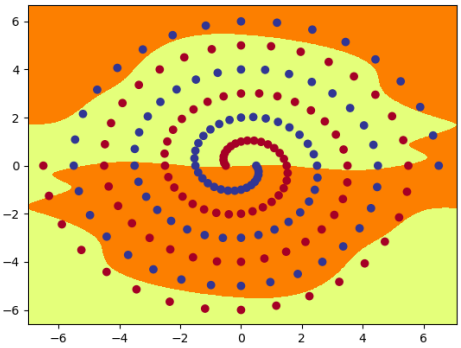
raw2_4.png



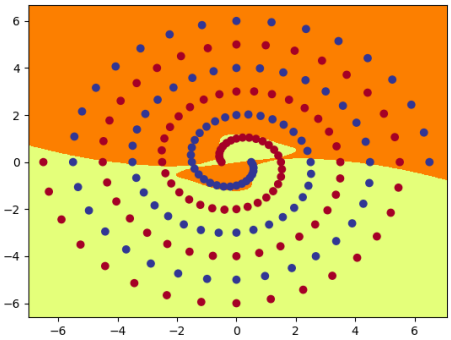
raw2_5.png



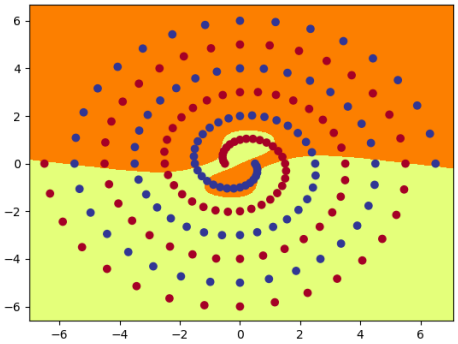
raw2_6.png



raw2_7.png



raw2_8.png



raw2_9.png

6.

a. The function of each hidden node in PolarNet is non-linear and the first hidden layer nodes have decision boundary linear function, but the second hidden layer nodes have non-linear function. Every hidden node only learnt a part of correct decision boundary and it will not be influenced by the network and the kind of function hidden nodes. To achieve the classification boundary, the weighted sum is applied for the network to learn.

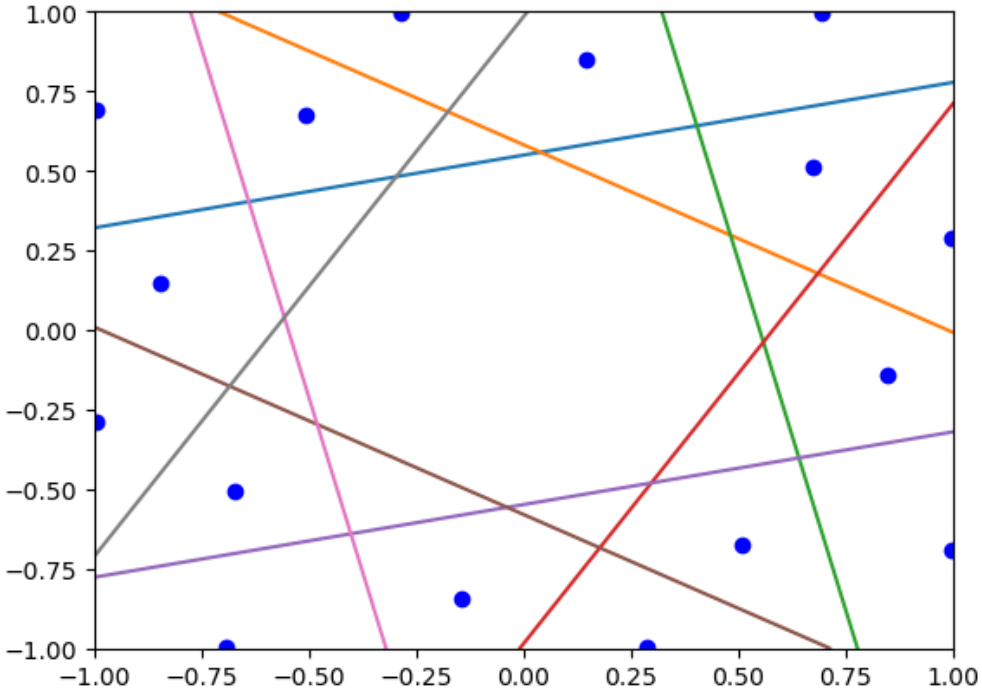
b. I tired different initial weights from 0.1 to 0.5 and found most successful weights is between 0.1 and 0.2. It's too large if the weight is over 0.3, and it's usually failed. So the success of learning is closely influenced by initial weight but the speed doesn't have close connect to the initial weight.

c. I changed the tanh function to the relu and keep others no change. I run the RawNet with 0.17 initial weight and the accuracy can't reach 100% within 20000 epochs and it reached 90% to 95%. This may because relu

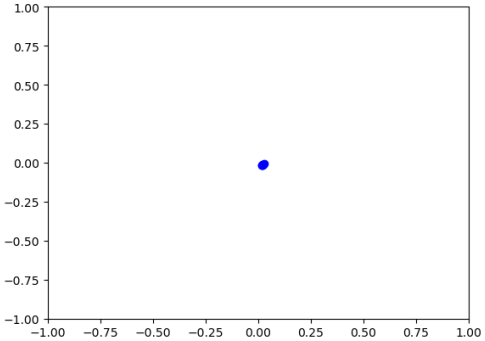
function may make some neurons to 0 and these can’t be activated by further layers. So relu function may cause underfitting and tanh is better than relu.

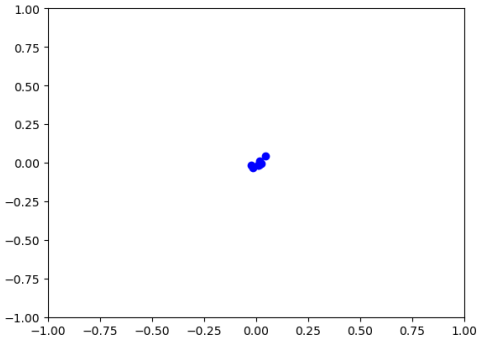
Part 3: Hidden Unit Dynamics

1.



2. first eleven images:





1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

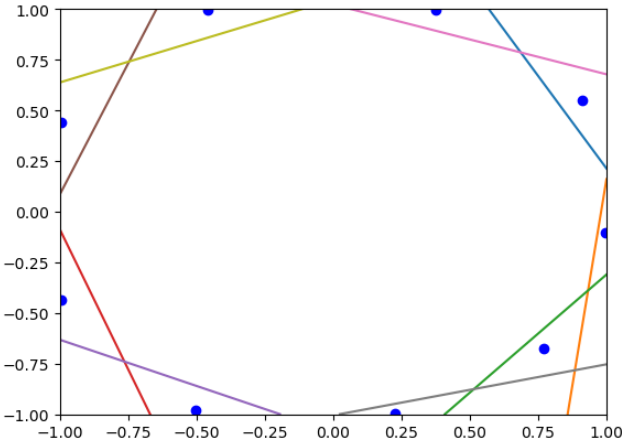
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

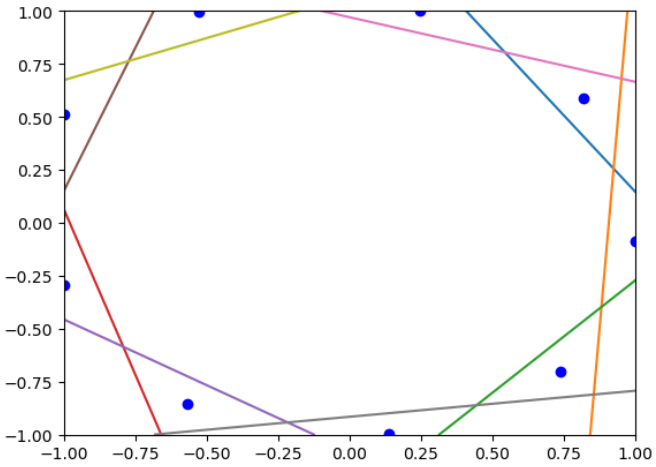
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

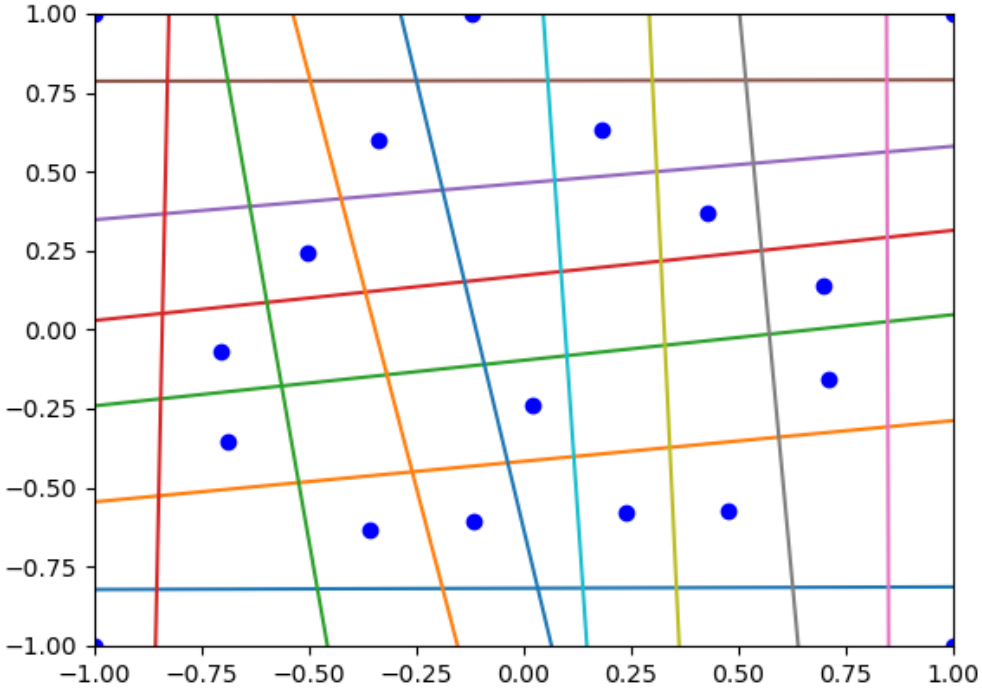


Final image:

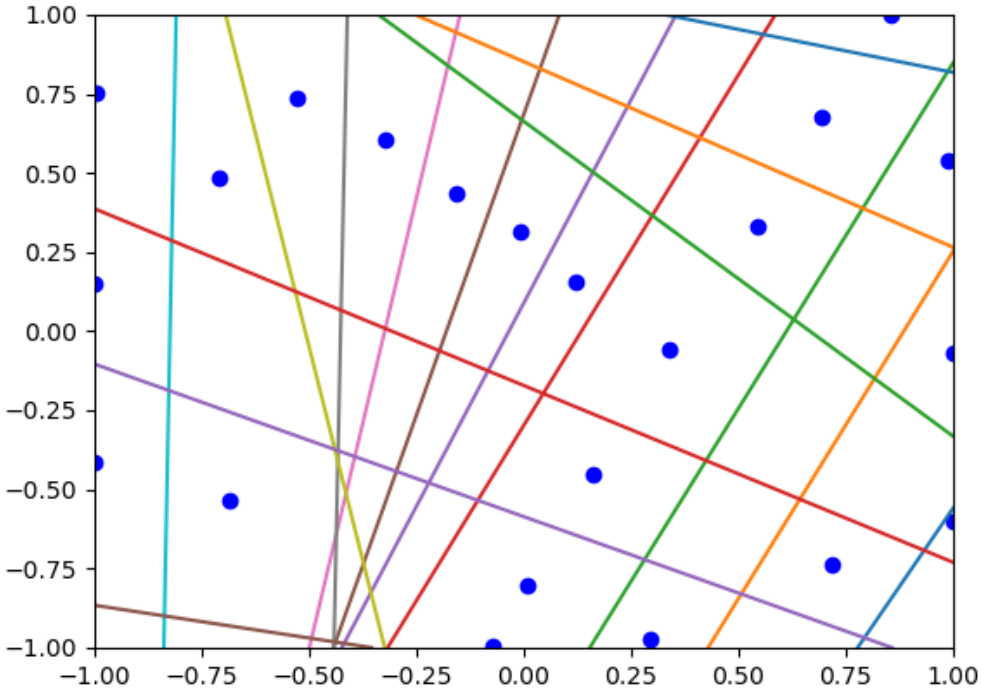


At first, all the hidden units get together and as the training progresses step by step, these units will disperse and the output boundaries will distinguish the different parts of the units.

3.



4. target 1: an umbrella



Target 2: a Joker face