COMP9444
Nan Zhao
Z5225777
Project 1

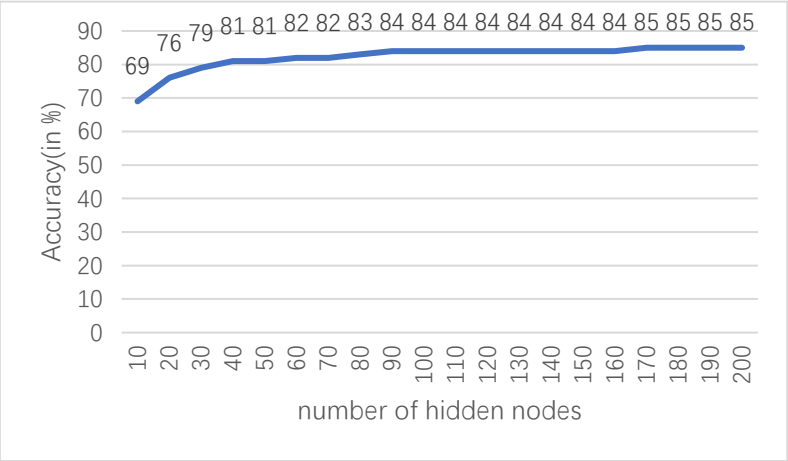
Project 1 – Japanese Characters and Intertwined Spirals

Part 1: Japanese Character Recognition

1.

```
[[769.   7.   8.   4.  60.   7.   5.  18.  11.   7.]
 [  5. 672.  64.  39.  53.  28.  23.  28.  35.  52.]
 [  7. 108. 692.  59.  77. 122. 144.  26.  96.  82.]
 [ 13.  18.  25. 758.  18.  17.  10.  10.  40.   3.]
 [ 30.  27.  26.  15. 625.  19.  25.  87.   6.  54.]
 [ 66.  21.  21.  54.  21. 727.  24.  18.  29.  34.]
 [  2.  58.  46.  14.  33.  27. 725.  54.  44.  18.]
 [ 61.  13.  36.  19.  36.   8.  20. 620.   6.  28.]
 [ 30.  26.  44.  28.  22.  34.  10.  91. 712.  40.]
 [ 17.  50.  38.  10.  55.  11.  14.  48.  21. 682.]]

Test set: Average loss: 1.0089, Accuracy: 6982/10000 (70%)
```

2. By trying different values for the number of hidden nodes manually, I got the corresponding accuracies shown in the following chart.



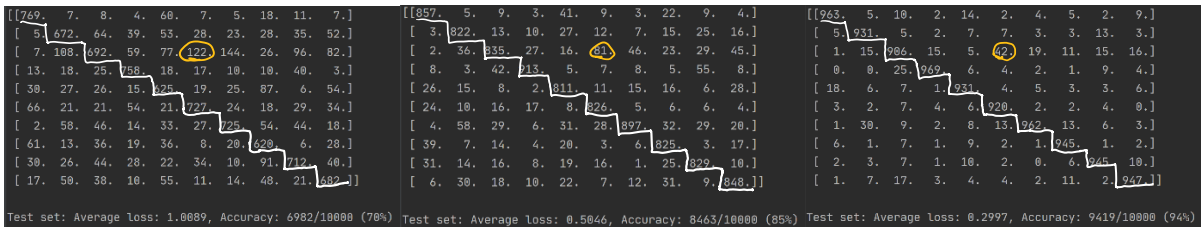
The accuracy reaches the highest value 85% when the number of hidden nodes is 170.

```
[[857.   5.   9.   3.  41.   9.   3.  22.   9.   4.]
 [  3. 822.  13.  10.  27.  12.   7.  15.  25.  16.]
 [  2.  36. 835.  27.  16.  81.  46.  23.  29.  45.]
 [  8.   3.  42. 913.   5.   7.   8.   5.  55.   8.]
 [ 26.  15.   8.   2. 811.  11.  15.  16.   6.  28.]
 [ 24.  10.  16.  17.   8. 826.   5.   6.   6.   4.]
 [  4.  58.  29.   6.  31.  28. 897.  32.  29.  20.]
 [ 39.   7.  14.   4.  20.   3.   6. 825.   3.  17.]
 [ 31.  14.  16.   8.  19.  16.   1.  25. 829.  10.]
 [  6.  30.  18.  10.  22.   7.  12.  31.   9. 848.]]

Test set: Average loss: 0.5046, Accuracy: 8463/10000 (85%)
```

3. I have tried several parameter values and finally got a 94% accuracy.

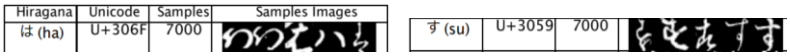
```
[[963.   5.  10.   2.  14.   2.   4.   5.   2.   9.]
 [  5. 931.   5.   2.   7.   7.   3.   3.  13.   3.]
 [  1.  15. 906.  15.   5.  42.  19.  11.  15.  16.]
 [  0.   0.  25. 969.   6.   4.   2.   1.   9.   4.]
 [ 18.   6.   7.   1. 931.   4.   5.   3.   3.   6.]
 [  3.   2.   7.   4.   6. 920.   2.   2.   4.   0.]
 [  1.  30.   9.   2.   8.  13. 962.  13.   6.   3.]
 [  6.   1.   7.   1.   9.   2.   1. 945.   1.   2.]
 [  2.   3.   7.   1.  10.   2.   0.   6. 945.  10.]
 [  1.   7.  17.   3.   4.   4.   2.  11.   2. 947.]]

Test set: Average loss: 0.2997, Accuracy: 9419/10000 (94%)
```

4. a. Accuracy of the linear function model: 70%
Accuracy of the fully connected 2-layer network model: 85%
Accuracy of the convolutional network: 94%
With the increasing complexity, the accuracy increases from 70% to 94%. The linear function is too simple to classify images, which will cause underfitting. The convolutional network model have better ability to extract information from images.
b.

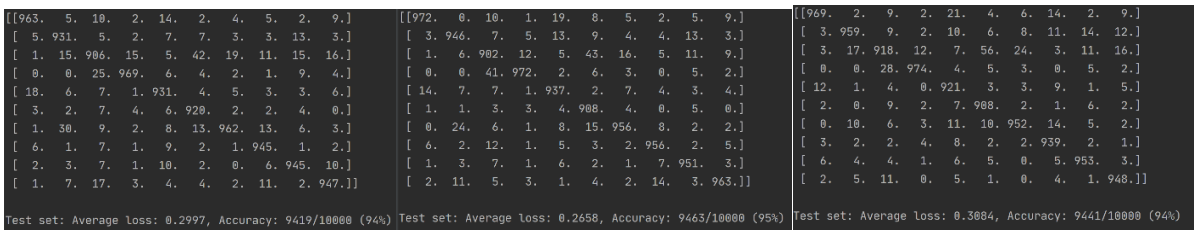


Linear function fully connected 2-layer network convolutional network

The confusion matrix is a matrix used to measure the performance of machine learning. The rows indicate the target character and the columns indicate the predicted character. The correct predictions are located in the diagonal of the table, which are underlined in white. The character that is most likely to be mistaken among three models are circled in yellow. In these cases, the character “ha” is predicted to “su”. The sample images for these two characters are not consistent, which may be one of the reasons of mistaken predictions.



c. Architecture 1: Conv1->relu->Conv2->relu->Linear->relu->Linear->log_softmax
Architecture2:
Conv1->relu->Conv2-> relu-> max_pooling->Linear->relu->Linear->log_softmax
Architecture3:
Conv1->relu->Conv2-> relu-> max_pooling-> Conv3->relu->Conv4-> relu-Linear->relu->Linear->log_softmax

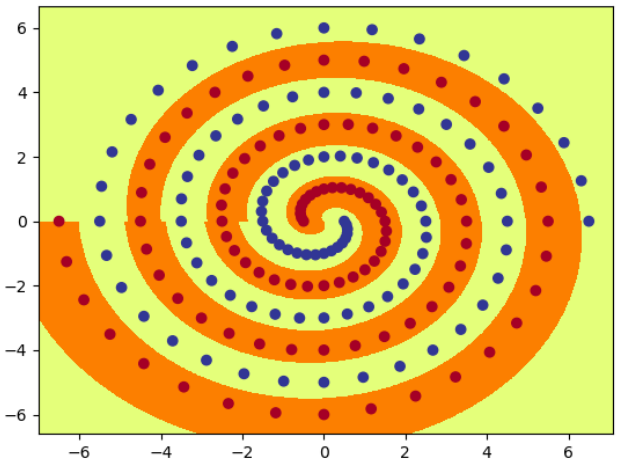


Architecture 1 Architecture 2 Architecture 3

3 different architectures are implemented. Architecture 2 has a max pooling layer which architecture 1 does not have and it slightly increase the accuracy. Architecture 3 has a deeper framework but the accuracy is almost the same with architecture 1. Further parameter configurations may be required in order to reach higher accuracy and better performance.

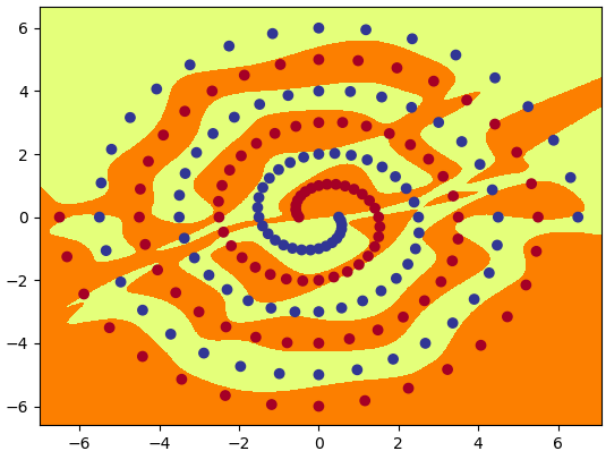
Part 2: Twin Spirals Task

2. Minimum number of hidden nodes: 7

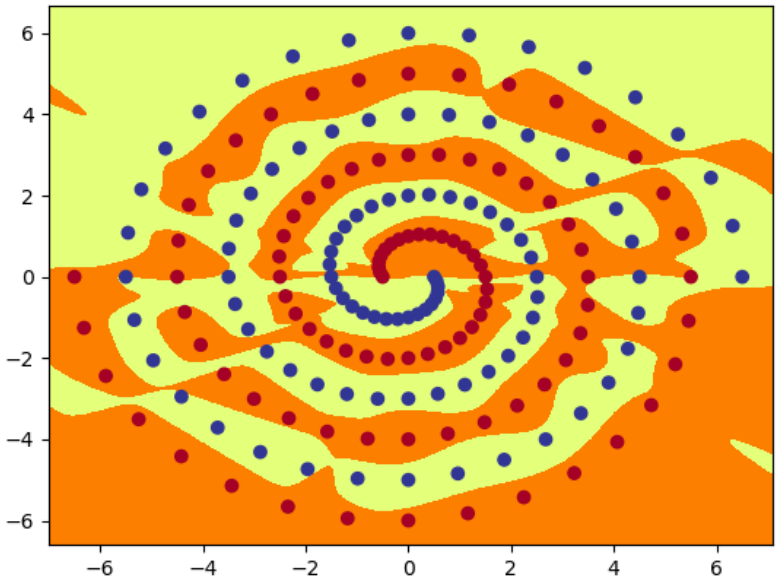


4. Initial weights: 0.2; Humber of hidden nodes: 10

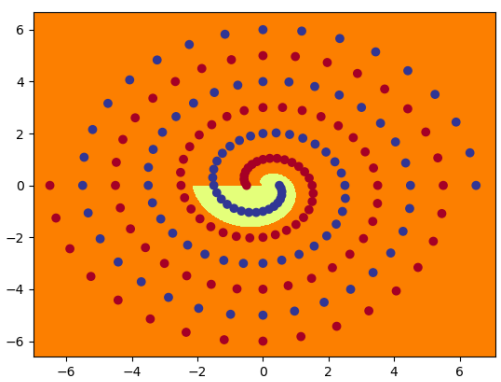
In this case, this RawNet learns to correctly classify all of the training data within 20000 epochs in all 10 attempts.

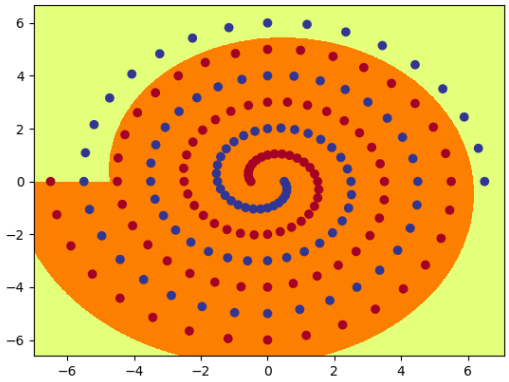


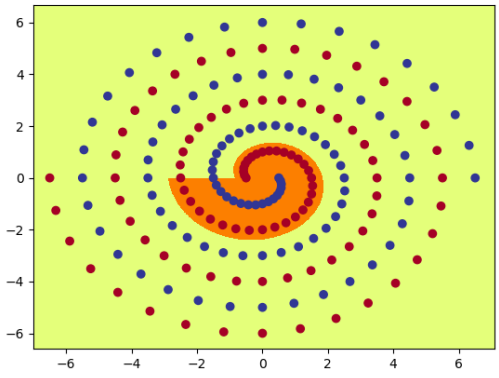
6. Initial weight size: 0.163; minimum number of hidden nodes per layer: 7

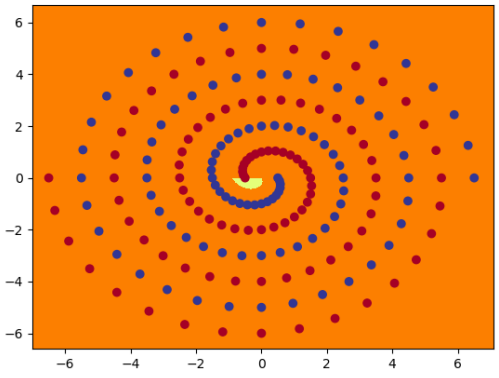


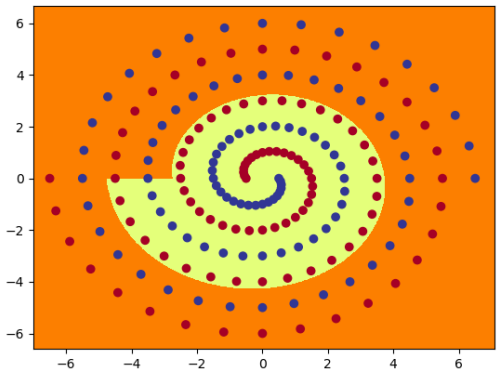
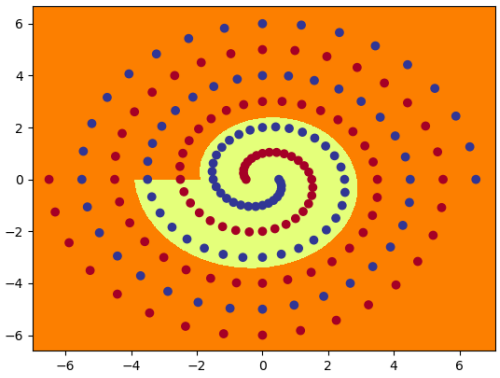
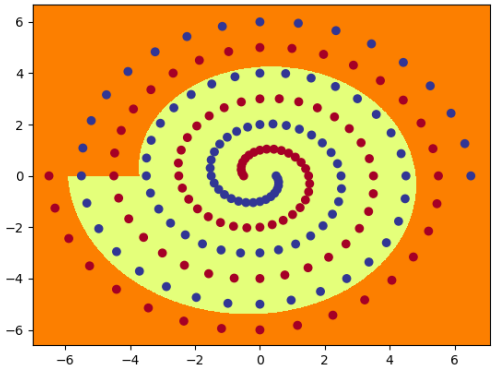
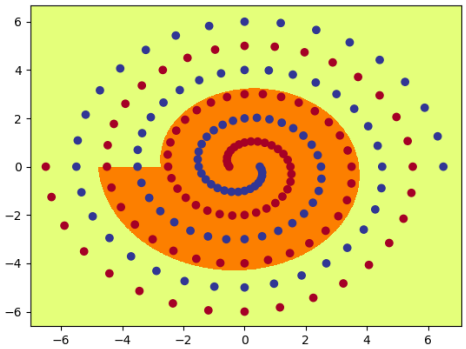
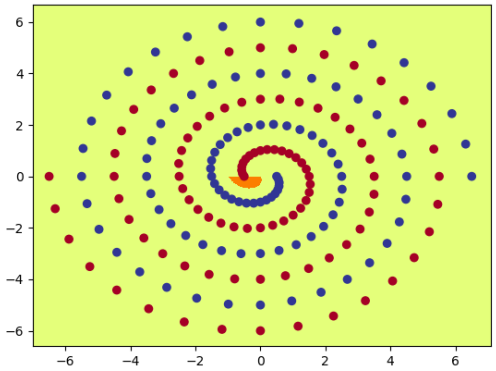
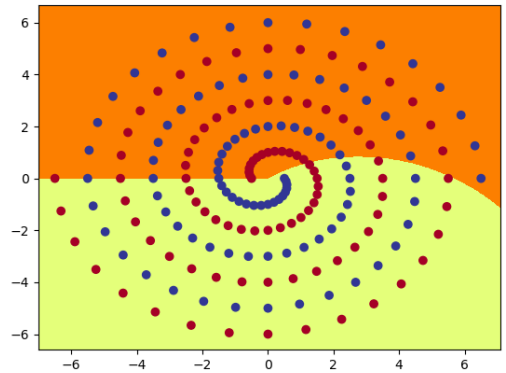
7. PolarNet:



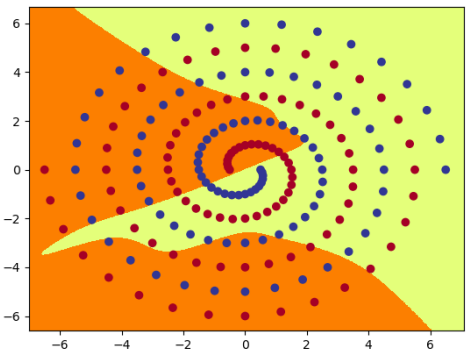
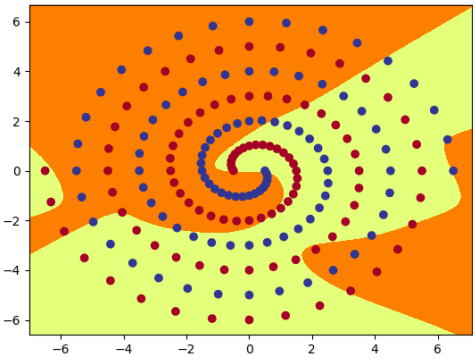


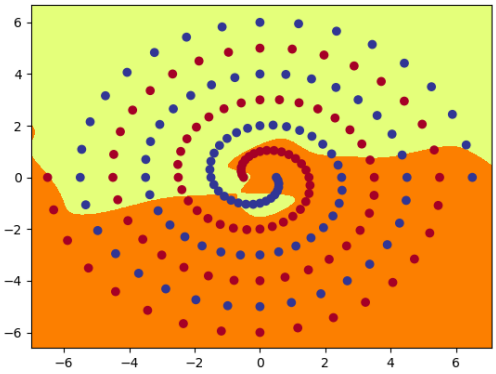
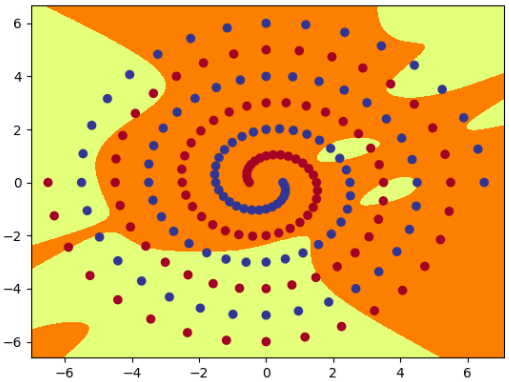
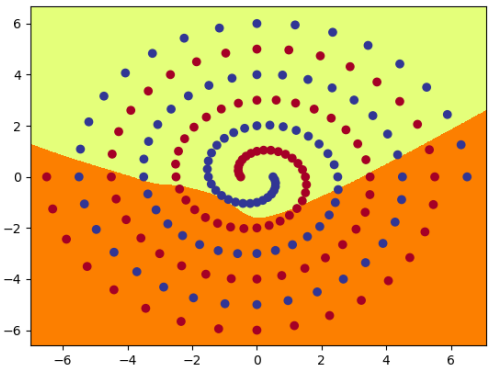
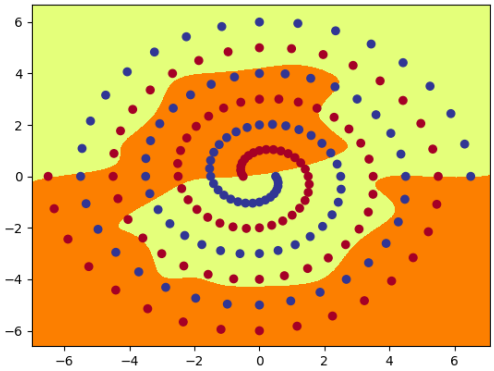
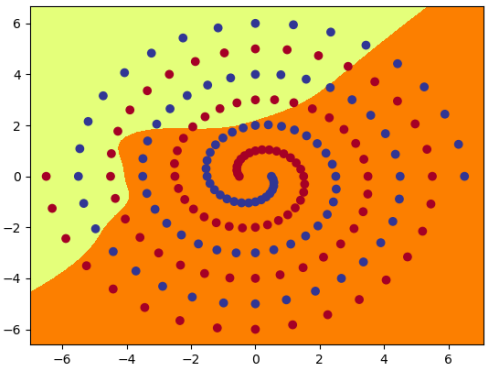
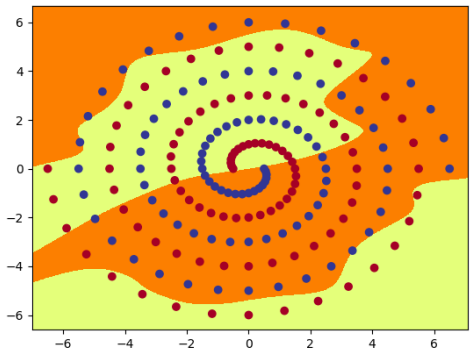
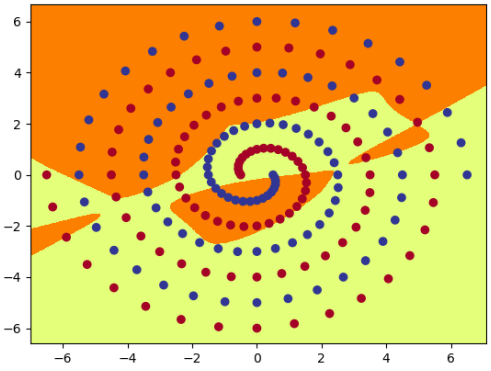
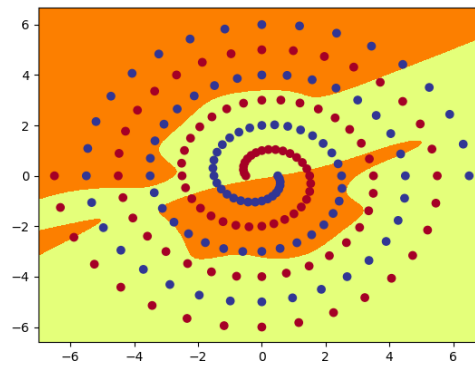




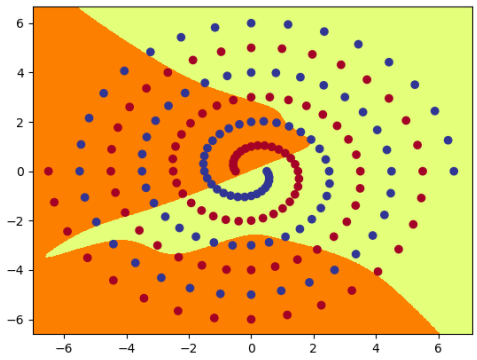
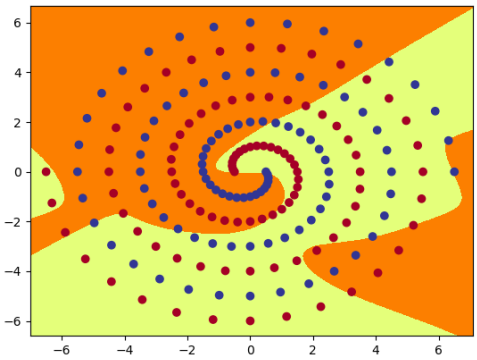
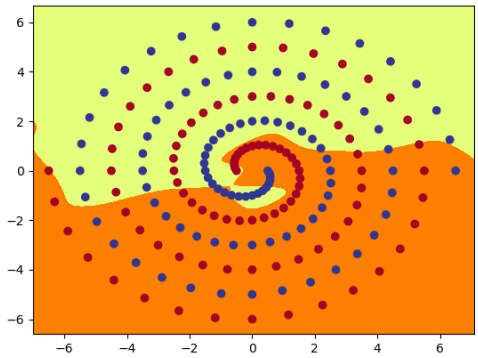
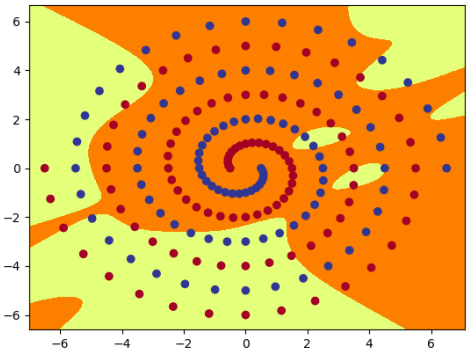
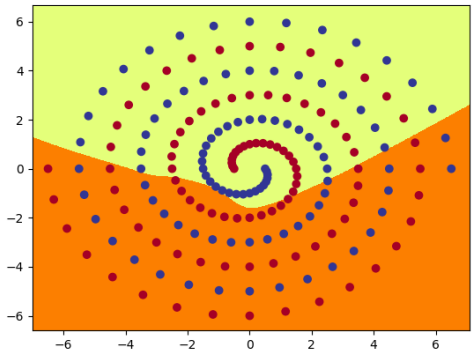
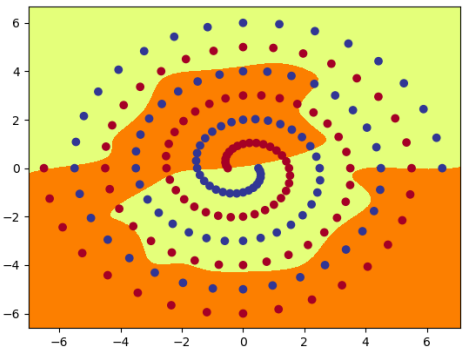
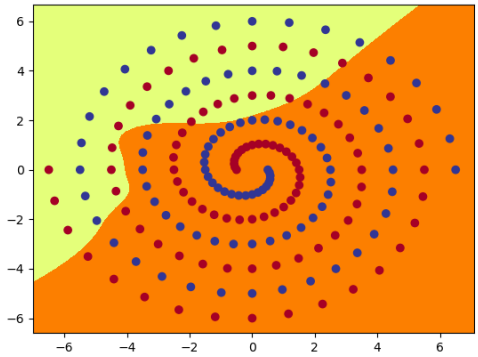
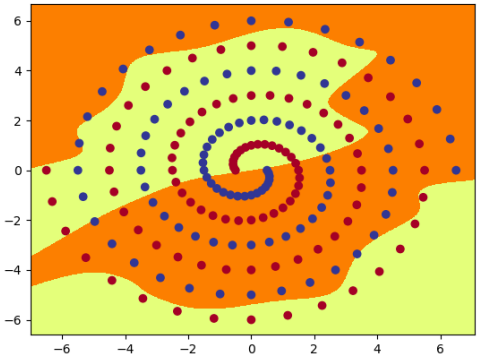







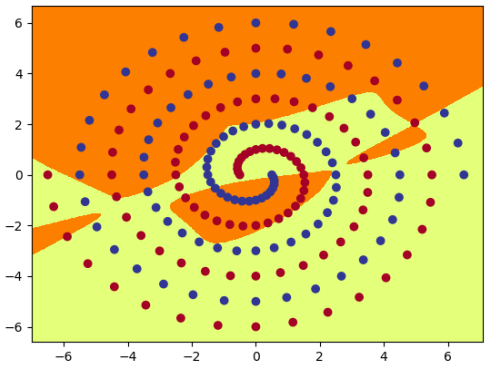
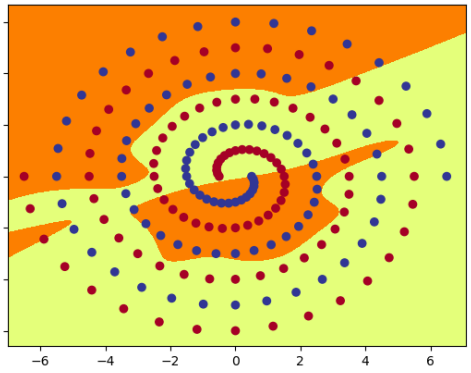
RawNet hidden layer 1:

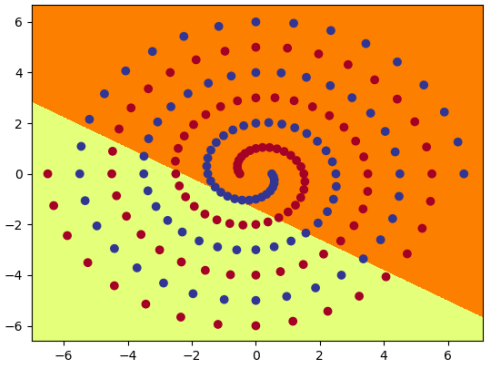
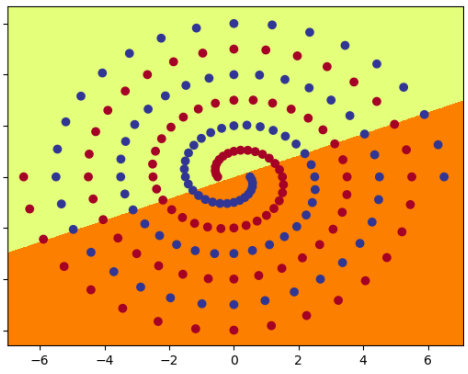
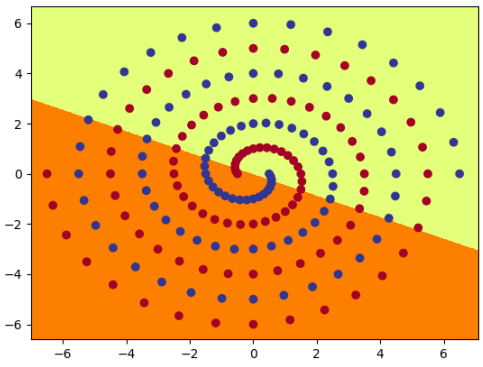
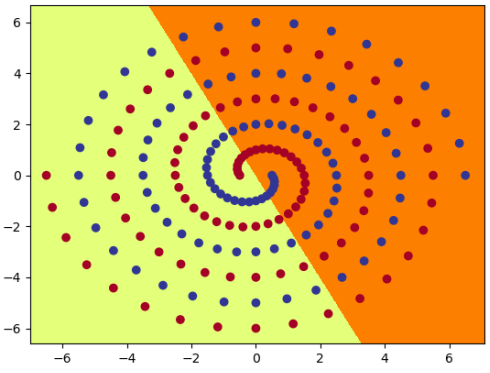
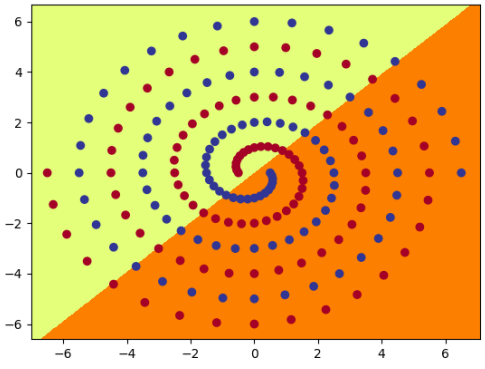
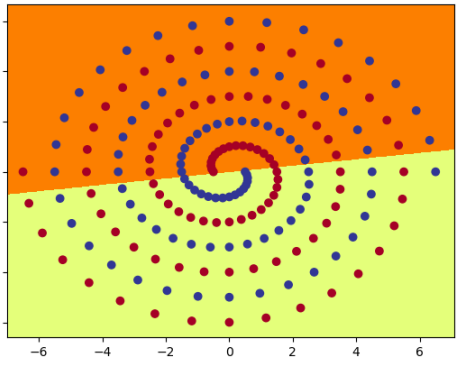









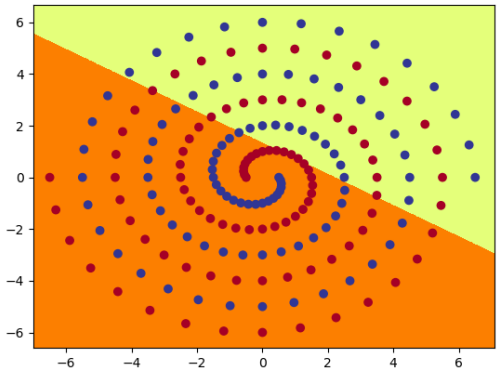
RawNet hidden layer 2:

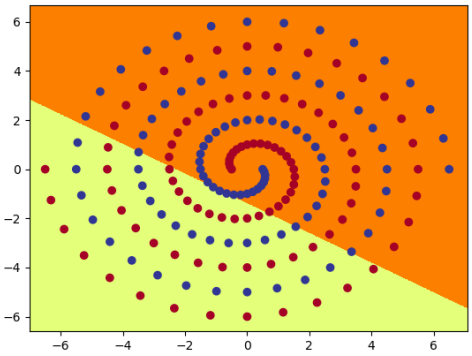



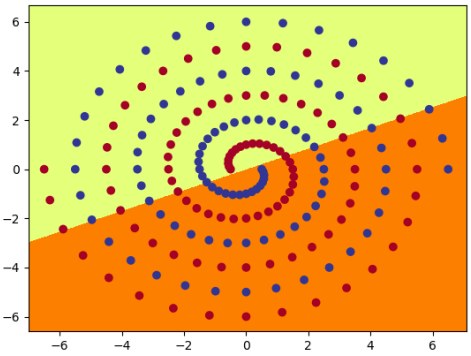
ShortNet Layer 1:

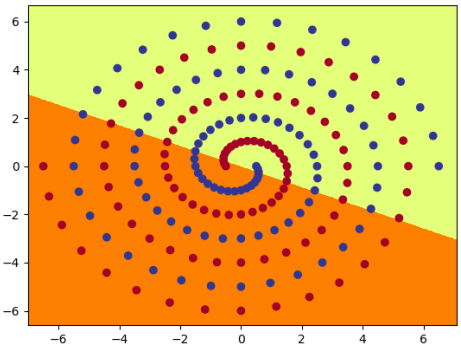







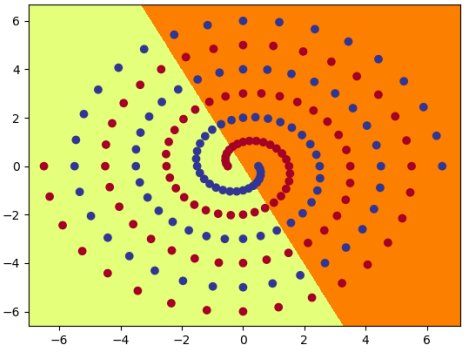


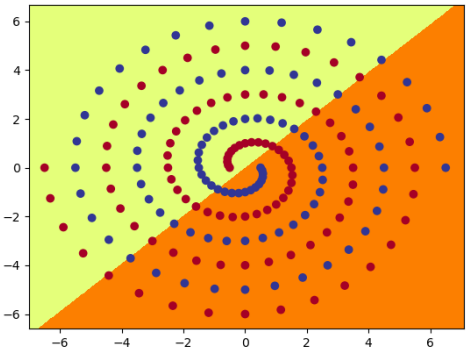
ShortNet Layer 2:

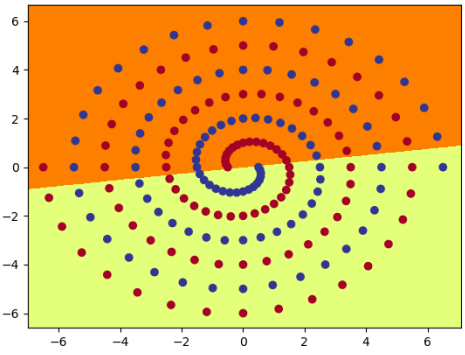


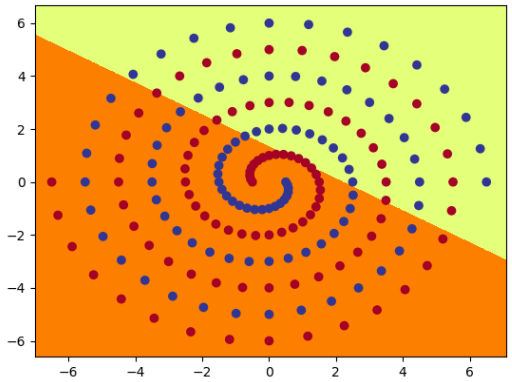












8. a. According to the graphs of hidden layers in q7, we can see that in PolarNet model, the hidden layer graphs tend to be glossy and some of them looks like a spiral. Each hidden can contribute some parts of the spirals with corresponding weights. Thus these hidden layers have a good performance in classify the twin spirals.
In the RawNet model, every hidden layers are look like irregularity curves. These layers can somehow classify the spirals but the performance is not good.
In the ShortNet model, every hidden layers are straight lines. The result model is sharp and has the worst performance in classify the spirals.
b. After several attempts, I found that neither model could correctly classify the model with initial weight less than 0.1 or hidden layer less than 6. Some successful initial weights are from 0.14 to 0.17 but the speeds are not affected. Even with same initial weights and hidden nodes, the speeds can be different in different runs. Successful hidden nodes are from 6 to 10 but in many cases hidden node number 6 did not work.
c. The result of PolarNet looks very “natural”. I believe that given some test data, the PolarNet model can do a good job. RawNet and ShortNet did successfully classify the trainning data but they look not natual at all. Although deep learning can usually successfully classify the training dataset but they can be not generalize enough for other data or test data.
d.

Batch size changed to 194	Successfully classify the data or not/Comments
PolarNet with default parameters	Yes and very fast
RawNet –init 0.2 –hid10	No. Accuracy stack at 95.88
ShortNet –init 0.2 –hid10	Yes and very fast

Optimiser changed to SGD lr=0.01, momentum=0.9	Successfully classify the data or not/Comments
PolarNet with default parameters	Yes but very slow. Success at epochs 39000.
RawNet –init 0.2 –hid10	Yes but very slow. Success at epochs 46000.
ShortNet –init 0.2 –hid10	Yes.