Part 1

1.

```
<class 'numpy.ndarray'>
[[764.   6.   7.   5.  58.   8.   5.  19.  11.   7.]
 [  6. 672.  60.  36.  52.  28.  25.  31.  38.  51.]
 [  9. 108. 691.  57.  79. 124. 144.  26.  93.  86.]
 [ 12.  18.  25. 755.  20.  17.   9.  12.  40.   3.]
 [ 30.  29.  26.  15. 626.  20.  25.  80.   7.  55.]
 [ 64.  22.  21.  58.  20. 725.  25.  14.  29.  33.]
 [  2.  58.  45.  14.  33.  26. 721.  55.  45.  18.]
 [ 63.  11.  40.  19.  37.   8.  21. 626.   6.  28.]
 [ 32.  26.  48.  28.  20.  33.  11.  88. 711.  40.]
 [ 18.  50.  37.  13.  55.  11.  14.  49.  20. 679.]]

Test set: Average loss: 1.0101, Accuracy: 6970/10000 (70%)
```

```
<class 'numpy.ndarray'>
[[844.   6.   7.   2.  40.   8.   3.  19.  13.   2.]
 [  6. 819.  11.  10.  38.  15.  13.  16.  32.  17.]
 [  2.  32. 827.  26.  16.  76.  41.  24.  25.  42.]
 [  6.   3.  44. 923.   5.   6.   8.   5.  53.   5.]
 [ 32.  15.  14.   3. 811.  11.  13.  18.   4.  28.]
 [ 36.   9.  20.  15.   8. 840.   4.  10.   7.   7.]
 [  4.  59.  27.   3.  31.  21. 902.  24.  30.  22.]
 [ 41.   6.  12.   3.  20.   2.   7. 833.   3.  16.]
 [ 25.  20.  23.   6.  16.  17.   1.  24. 826.  13.]
 [  4.  31.  15.   9.  15.   4.   8.  27.   7. 848.]]

Test set: Average loss: 0.4955, Accuracy: 8473/10000 (85%)
```

2.

```
<class 'numpy.ndarray'>
[[941.   1.   7.   1.  14.   2.   3.   0.   4.   6.]
 [  6. 910.   8.   0.   7.  12.   1.   2.  25.   3.]
 [  0.   5. 890.  12.   2.  43.  10.   1.   5.  14.]
 [  1.   0.  40. 967.   8.   4.   1.   0.   3.   1.]
 [ 32.  17.   8.   3. 947.   8.   6.   0.   9.   9.]
 [  3.   0.  12.   3.   0. 900.   1.   0.   1.   0.]
 [  0.  36.  10.   5.   4.  14. 971.   3.   8.   4.]
 [ 12.   8.  12.   3.   7.   8.   2. 973.   4.  11.]
 [  1.   8.   5.   2.   9.   4.   1.   5. 936.  13.]
 [  4.  15.   8.   4.   2.   5.   4.  16.   5. 939.]]

Test set: Average loss: 0.2365, Accuracy: 9374/10000 (94%)
```

3.

4. a. the first model accuracy is 70%, the second is 85%, the third is 94%. The NetConv model is better than NetFull model and the NetFull model is better than NetLin model according to the result from three model. I can find that convolutional network behave better in processing image and one fully connected 2-layer is better than a linear function.

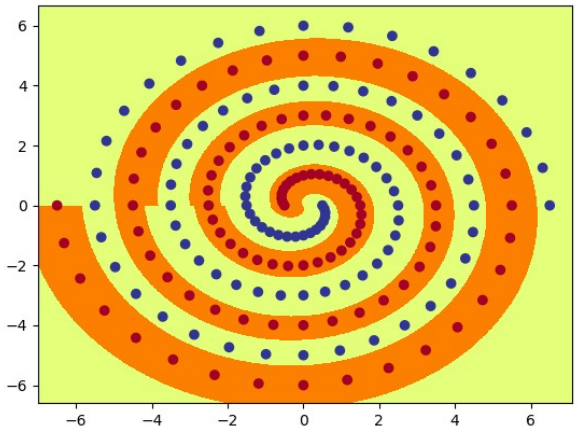
   b. on lin model, the character す is most likely to be mistake for ま, on NetFull and NetConv model, the character す is most likely to be mistake for は. These characters looks similar and from three model, the convolutional network make less mistake than a fully connected 2-layer and a fully connected 2-layer make less mistake than a linear function.

   c. I use different architectures and metaparameters on the Netconv model. Firstly, I don't use max pooling architecture and the accuracy is 92% which is less than 93%. Then, I change the Conv2 parameters and the accuracy is also 92%. Additionally, I use one max pooling after first Conv2-Tanh and the accuracy is 93%. Finally, I use two max pooling after every Con2-Tanh and the accuracy is 94%which is high than 94%. I also use different parameters for the NetFull model and find the accuracy is the same, being 85%. For these experiments, I find use more architecture can get high accuracy and convolutional network can handle image better.

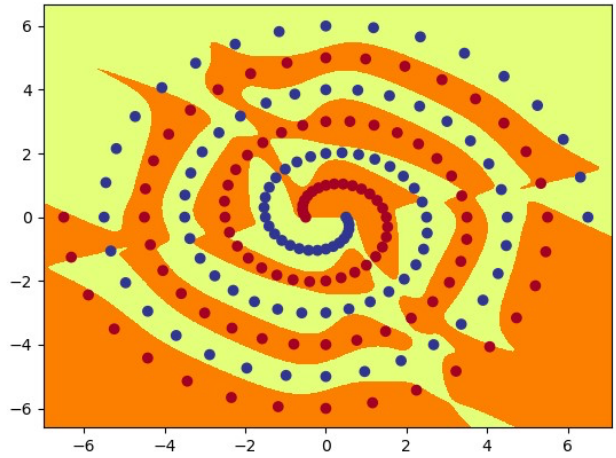
Part 2

2. The minimum number of hidden nodes is 7.

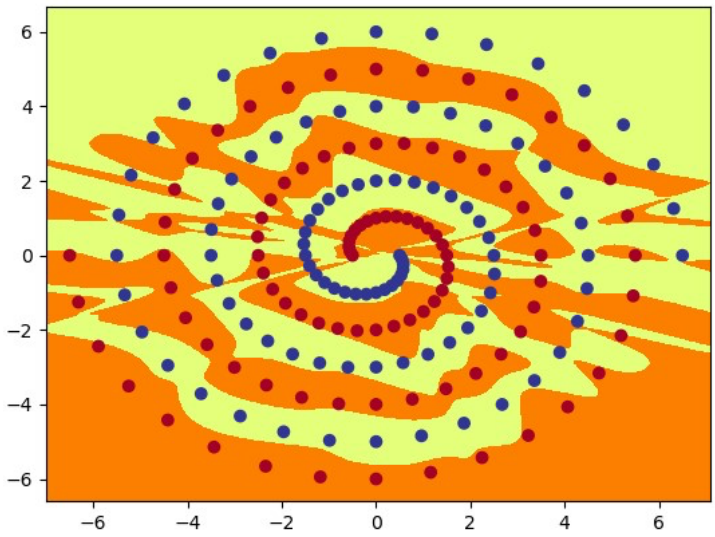
polar_out.png:



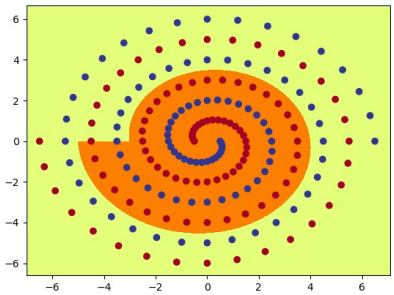
4. The initial weight is 0.15 and the hidden nodes is 10. raw_out.png:

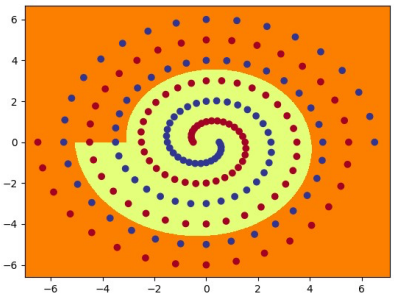


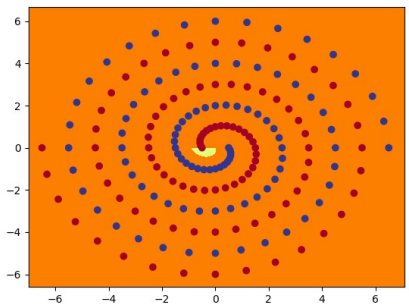
6. The init weight size is 0.15 is an appropriate number and the hidden nodes is 8. short_out.png:

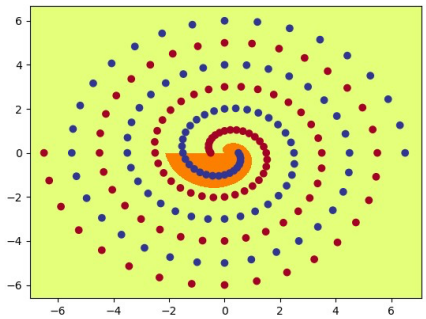


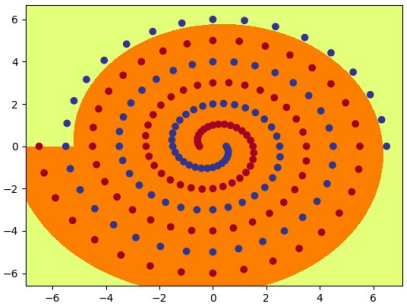
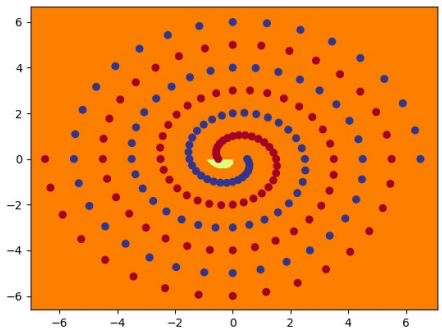
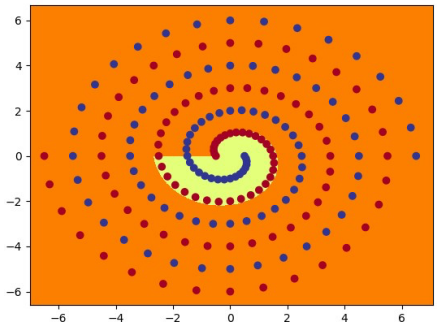
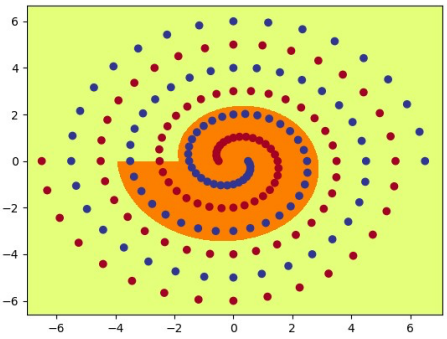
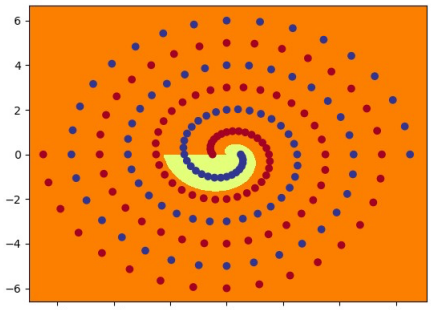
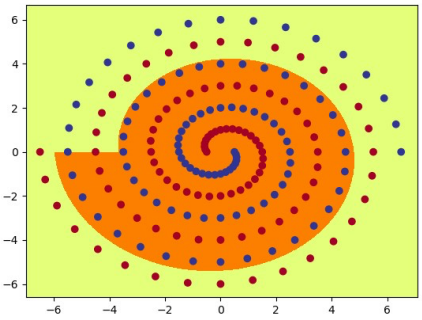
7. for PolarNet model, the hidden layer nodes:



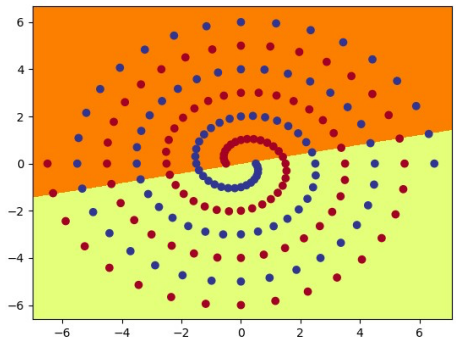
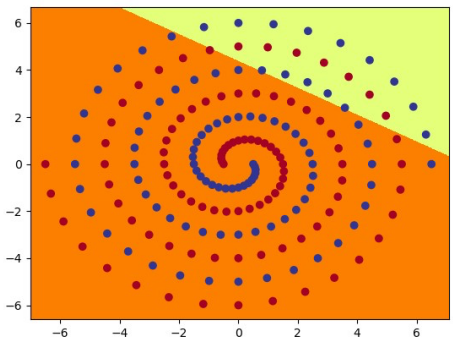


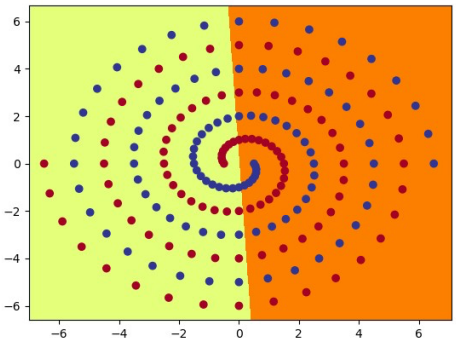
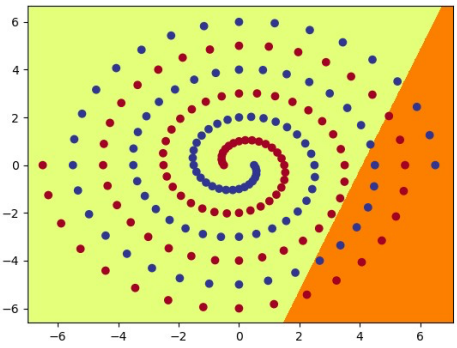
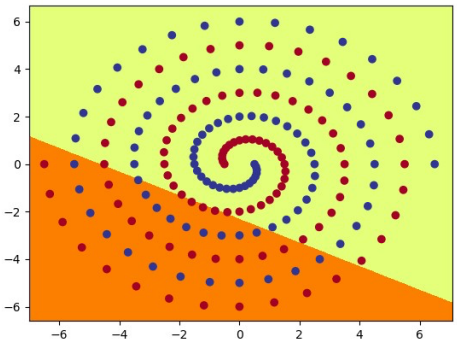
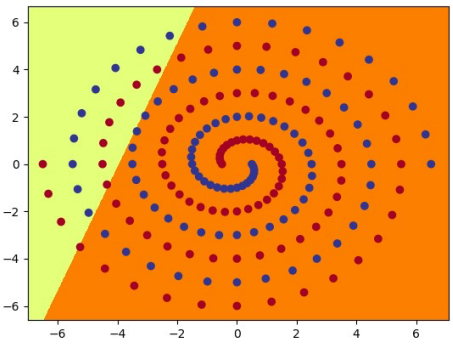
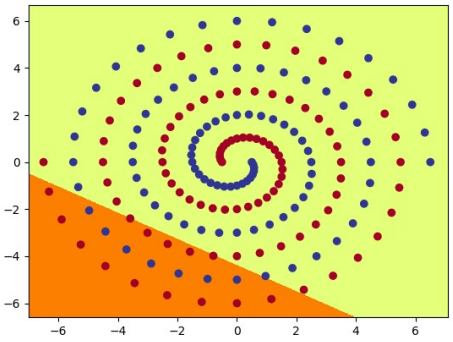
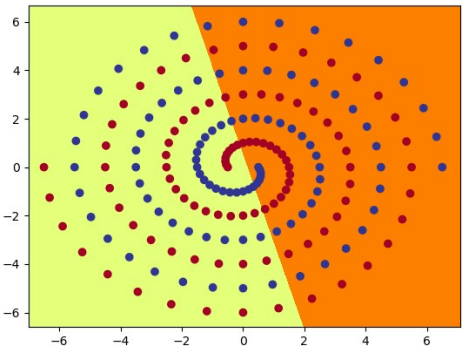
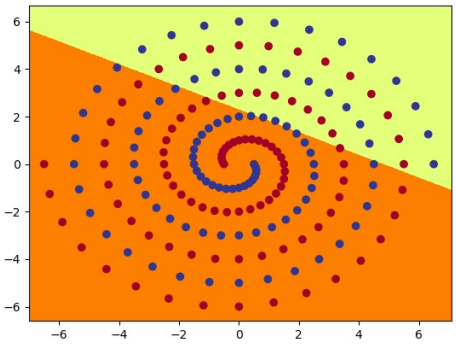
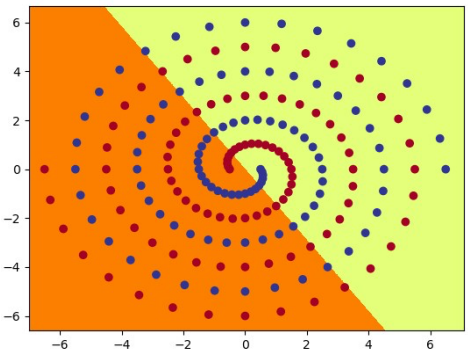


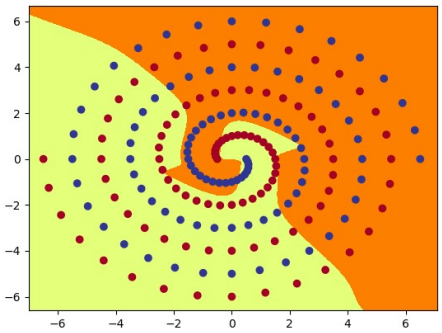
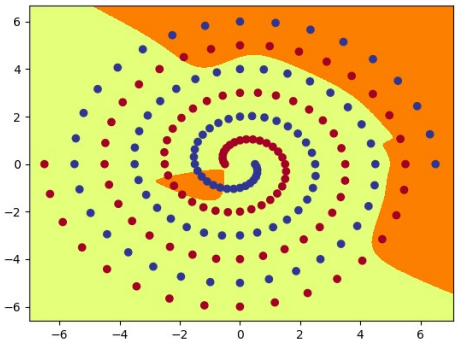
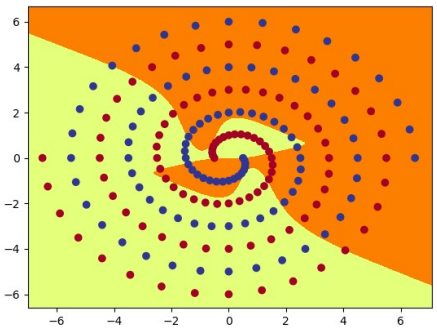
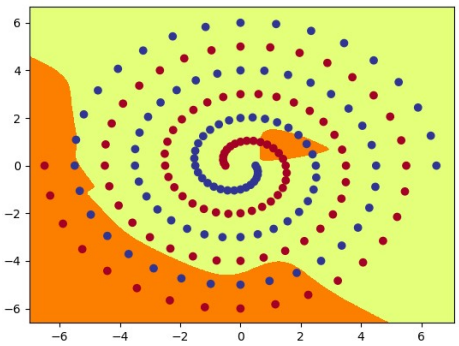
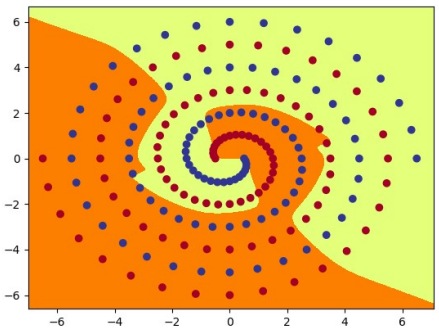
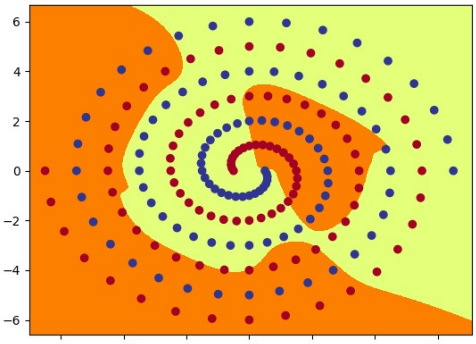
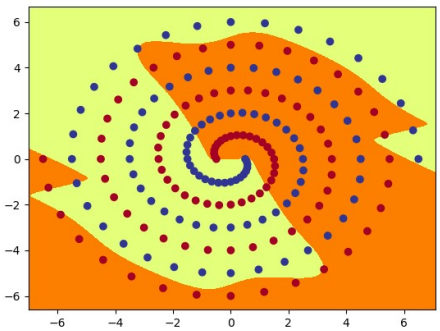
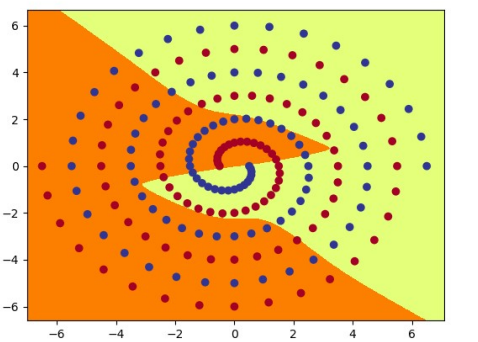


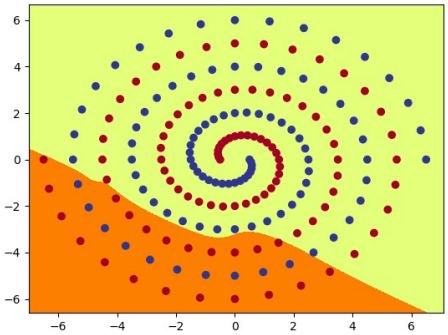
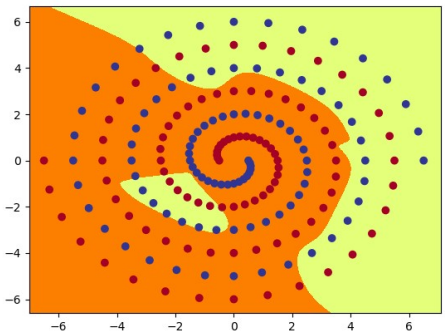







for RawNet model, the hidden layer nodes:

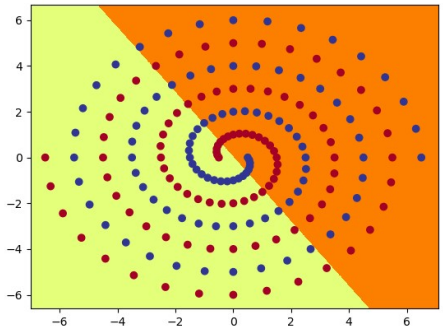
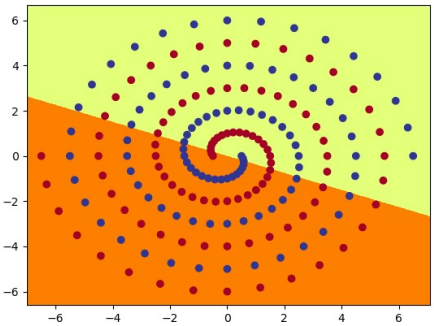
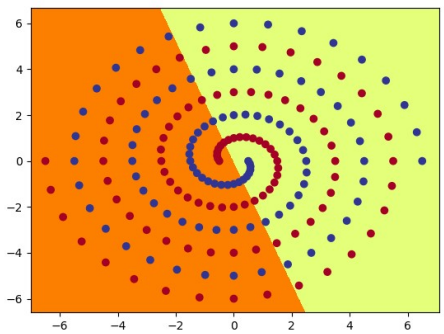
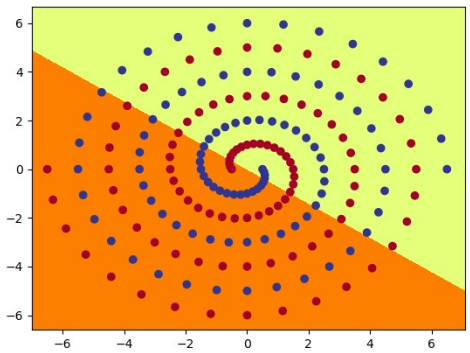
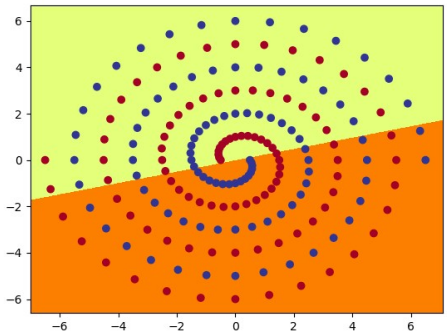
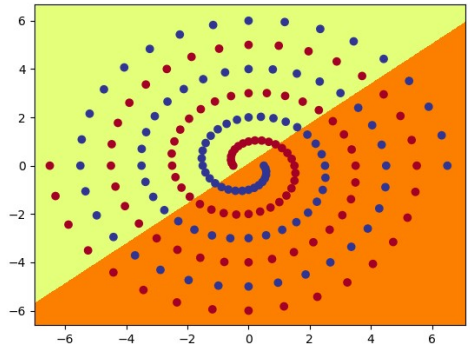



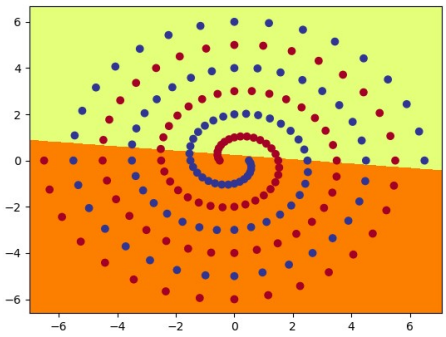
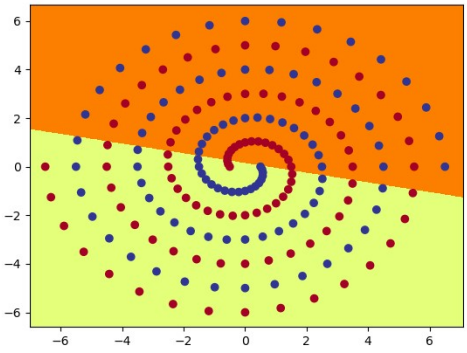
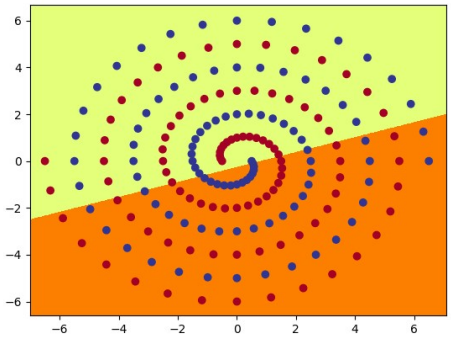
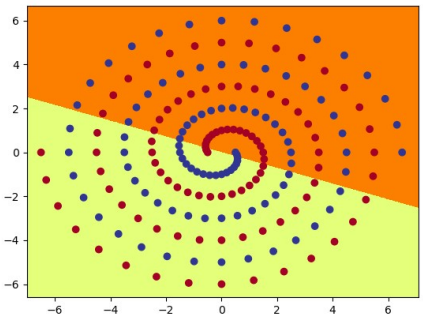
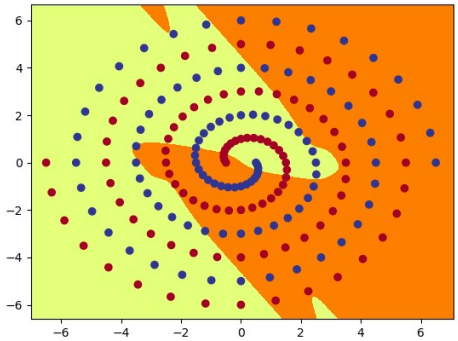
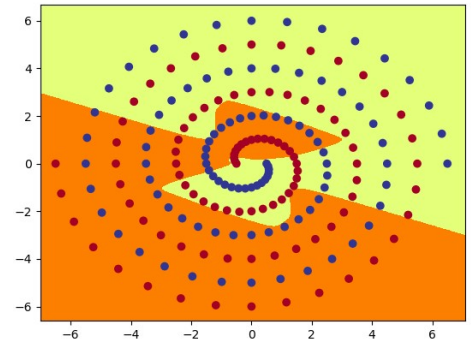
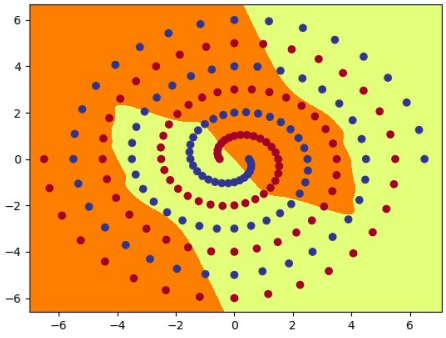
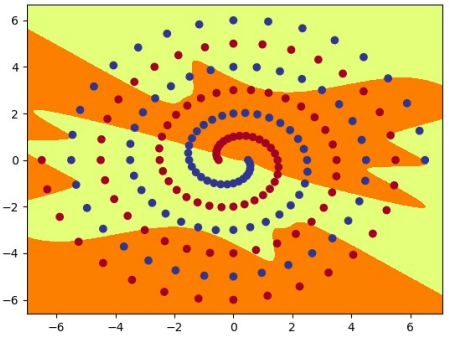









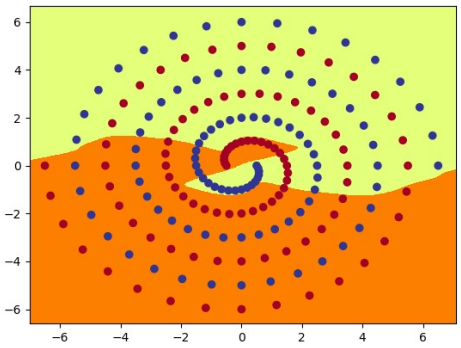
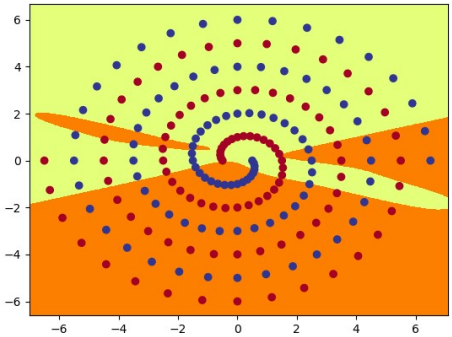
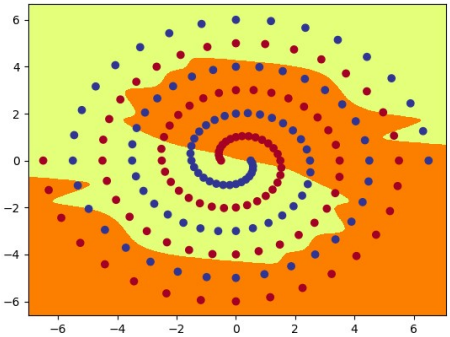
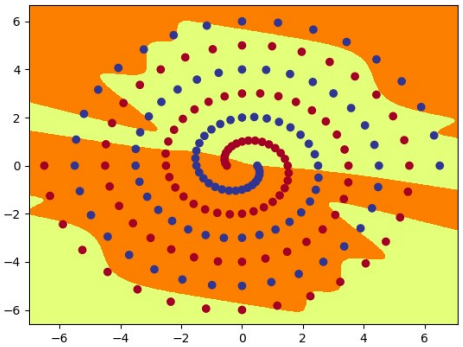
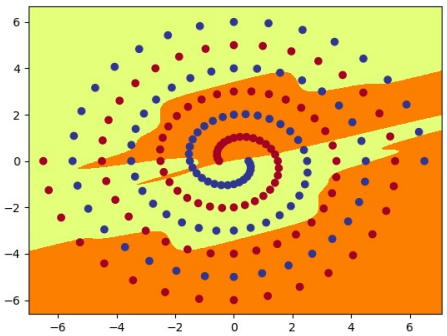
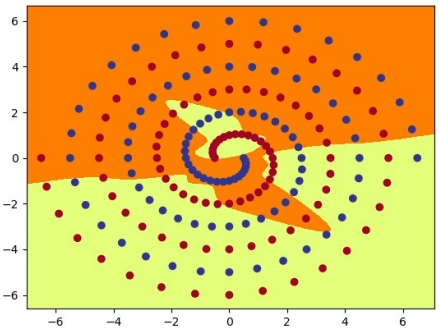









for ShortNet model, the hidden layer nodes:

8. a. From above images, for RawNet and ShortNet model, the image of hidden nodes of layer 1 is linear and the image of hidden nodes of layer 2 is not liner. For PolarNet, the image of hidden nodes is spiral. For raw and short net, using one linear function, the ouput is linear. If use one linear after a linear, the ouput is not linear. For polar net, There are some action to the input which input is converted to polar co-ordinates(r,a) with their expression and then use a fully connected neural network and tanh activation to get the hidden nodes which is a spiral. For raw net and short net, first using one linear to get hidden nodes which is linear and then use one linear again to get hidden nodes which is not linear.

b. I try different intit weight size for both RawNet and ShortNet. The size under 0.1has no effect and cannot do success of learing. The size between 0.1 and 0.2 can do success of learning and I try different size and find the 0.13 is fast which epochs are 3200. The 0.14 init size epochs are 9100. The 0.15 init size epochs are 6300.

c. For the three different models output images, I find the NetPolar model is more naturalness. I find that if hidden nodes is more, the output is more fitting. Also, the shortcut ouput is not naturalness.

d. Firstly, I change batch size from 97 to 194. I find speed is faster and the epochs is less then previous, I experiment three models, the PolarNet epochs are 900, the RawNet epochs are 3400 and the ShortNet epochs are 3900 but the output images look similar. Then, I change tanh to relu, I find speed is slow and needs more epochs than previous. I find change batch size from 97 to 194 can promote speed but change tanh to relu not and the accuracy is lower than previous.