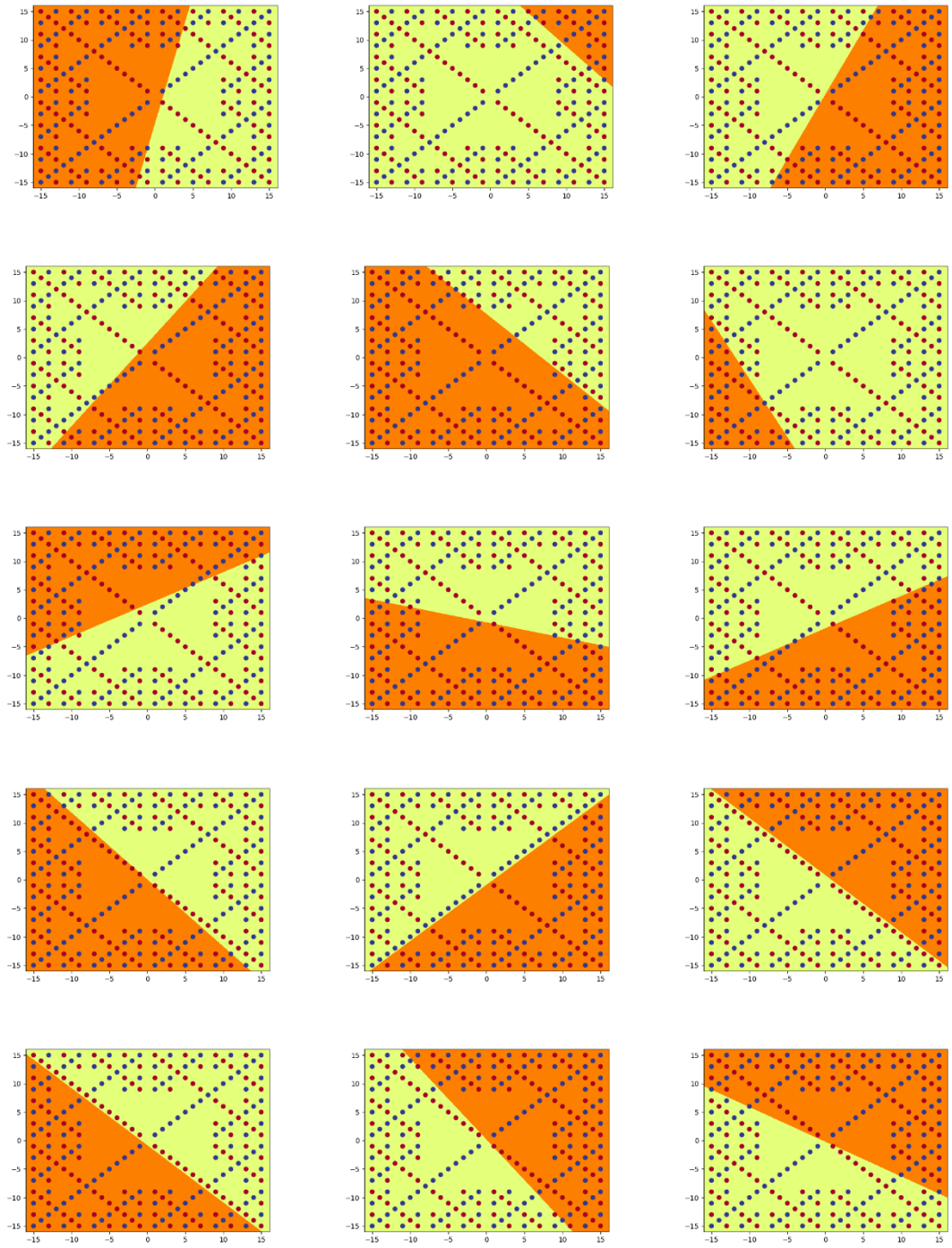
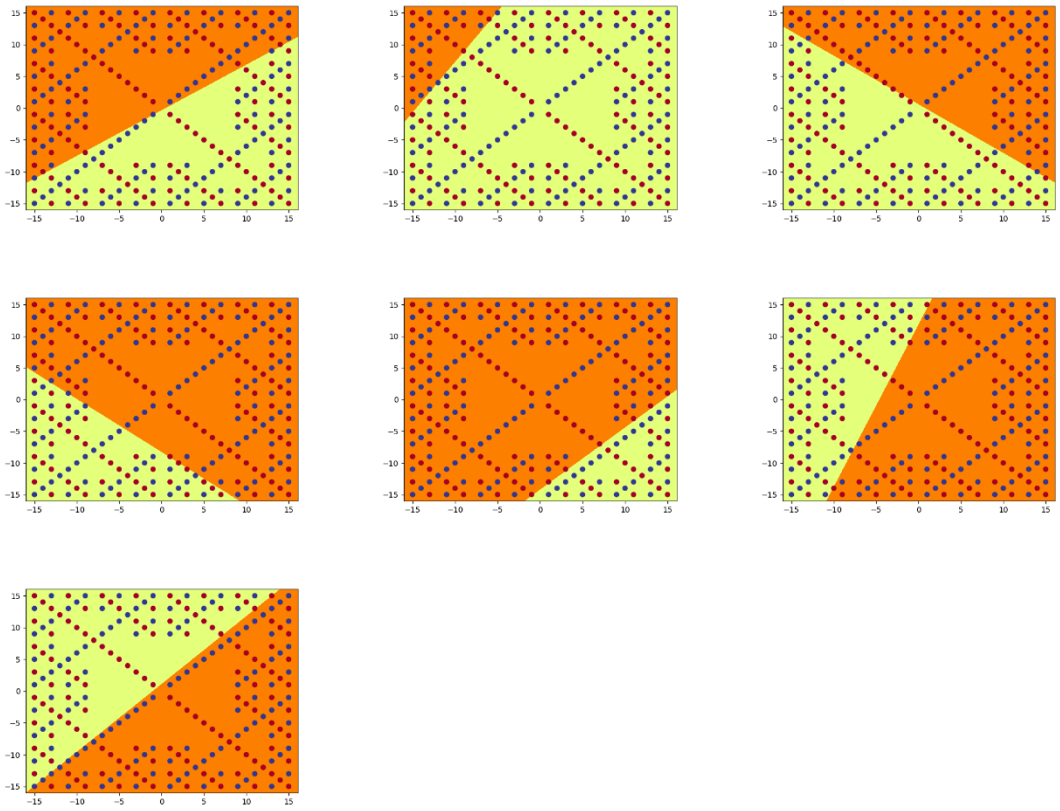
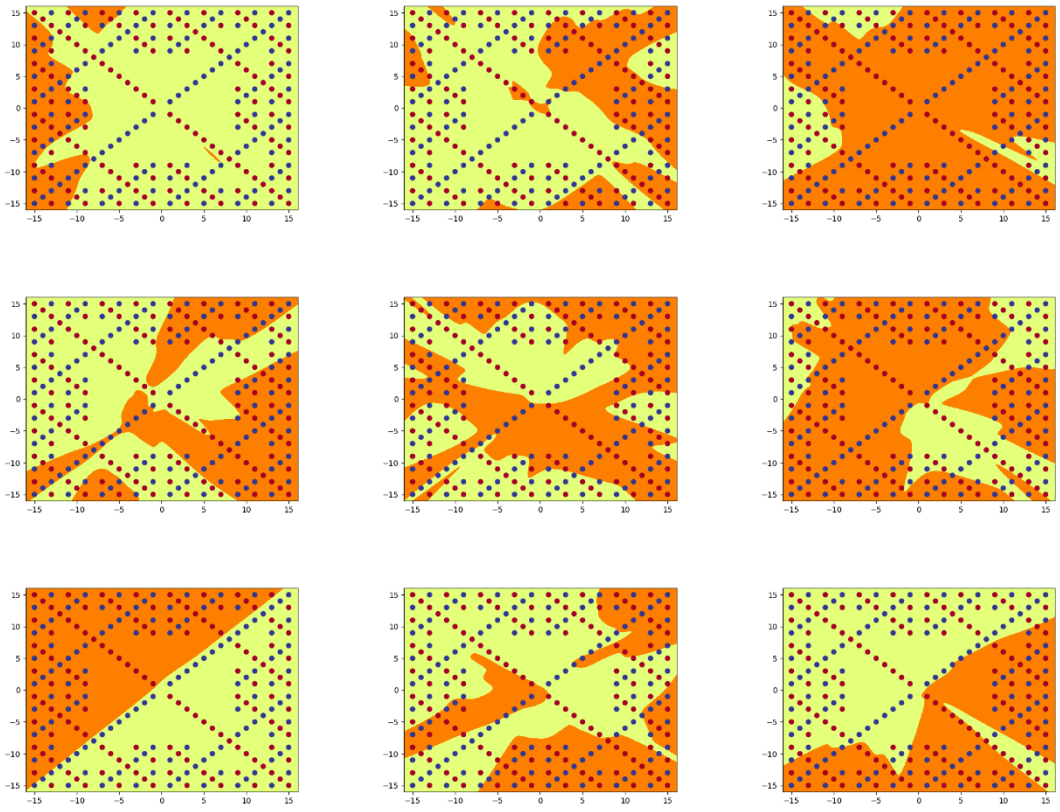
Part1

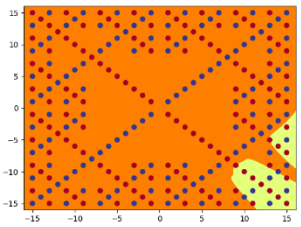
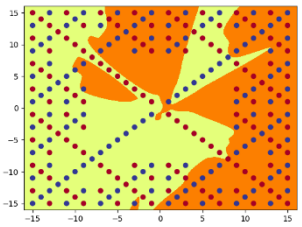
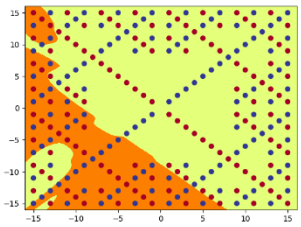
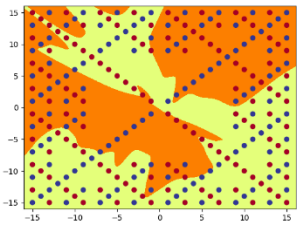
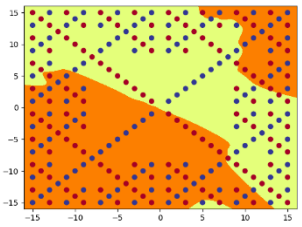
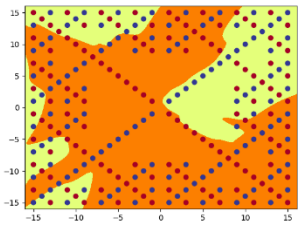
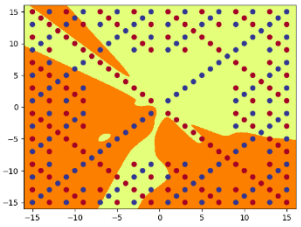
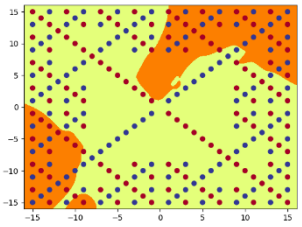
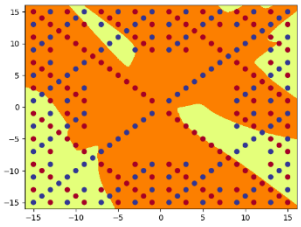
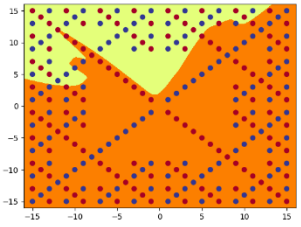
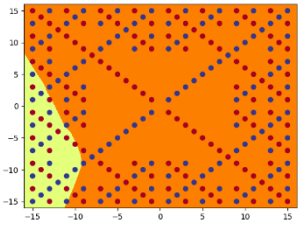
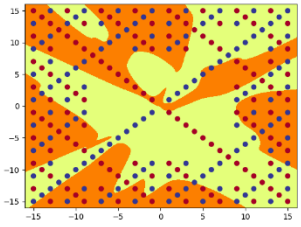
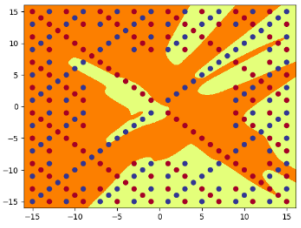
2.
hid = 22.
hid1:



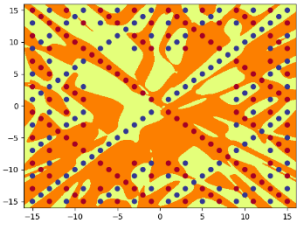


hid2:



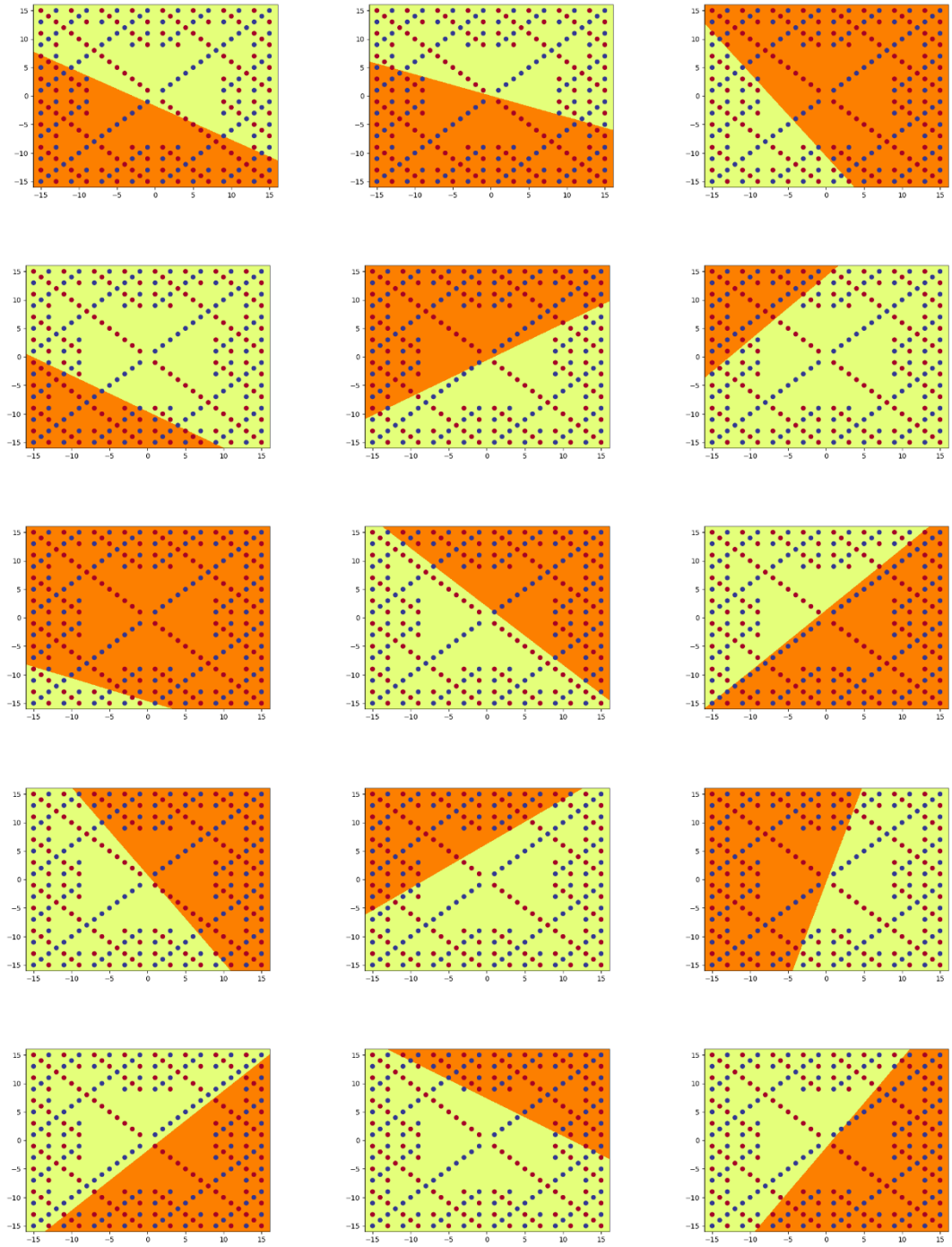














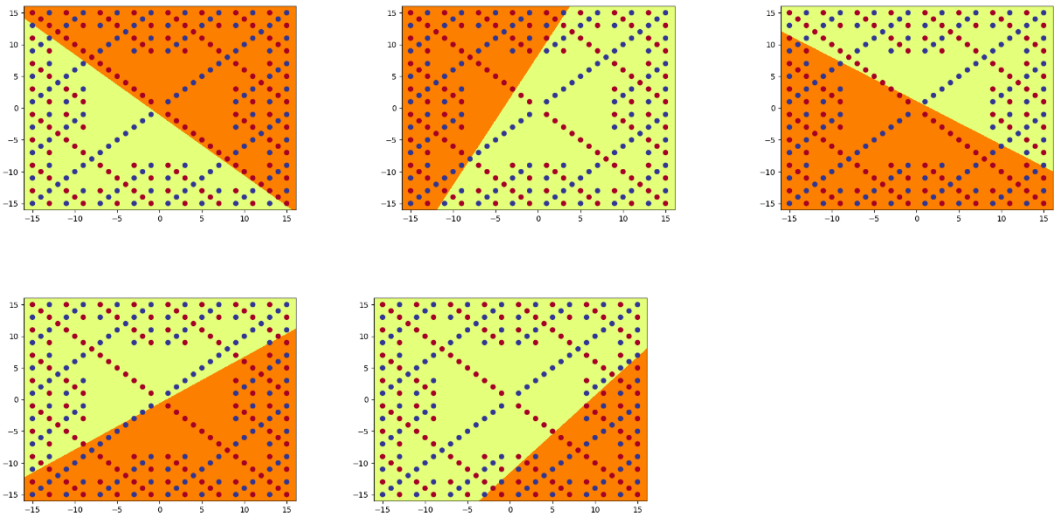
Output:



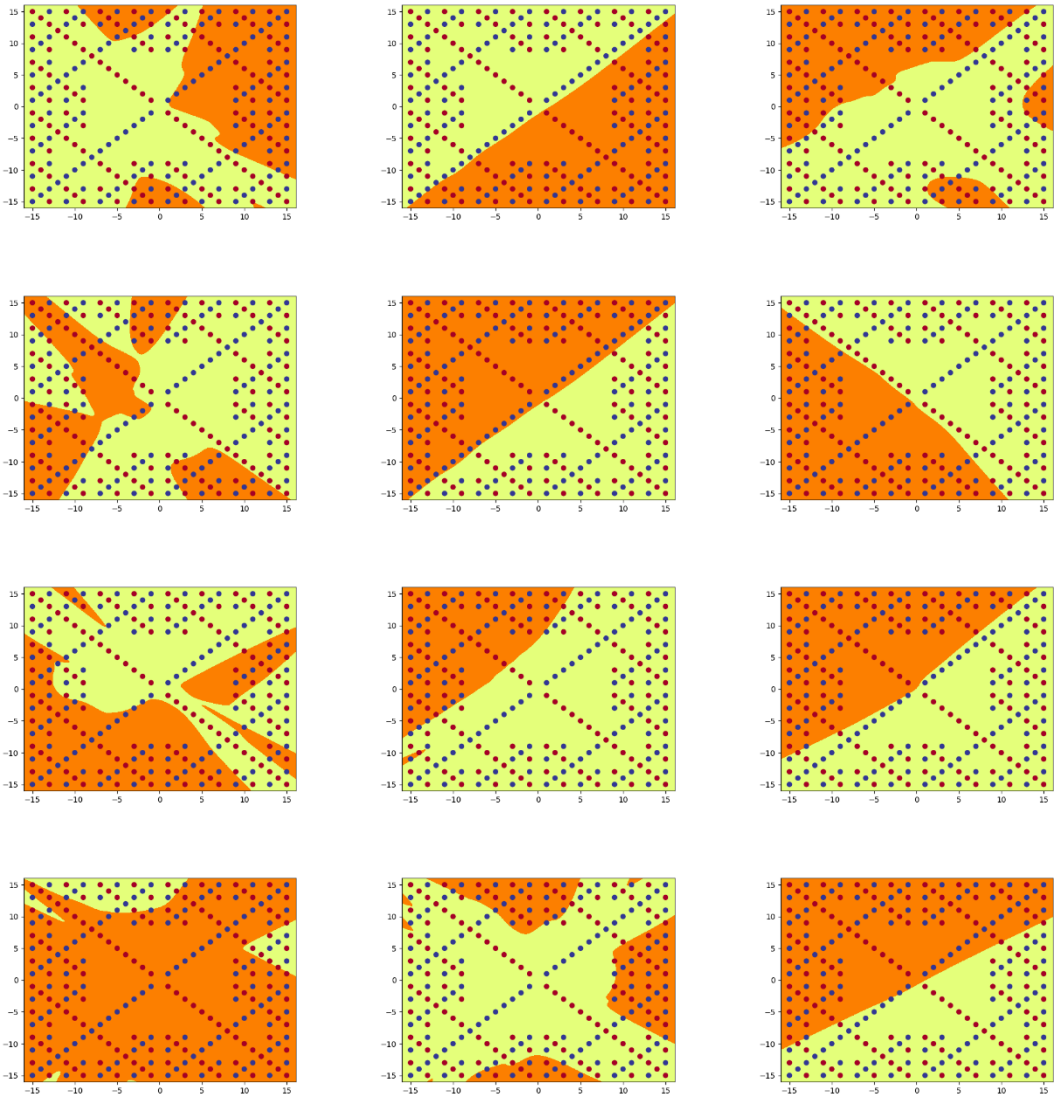
The number of independent parameters: 2*22+22*22+22*1=550

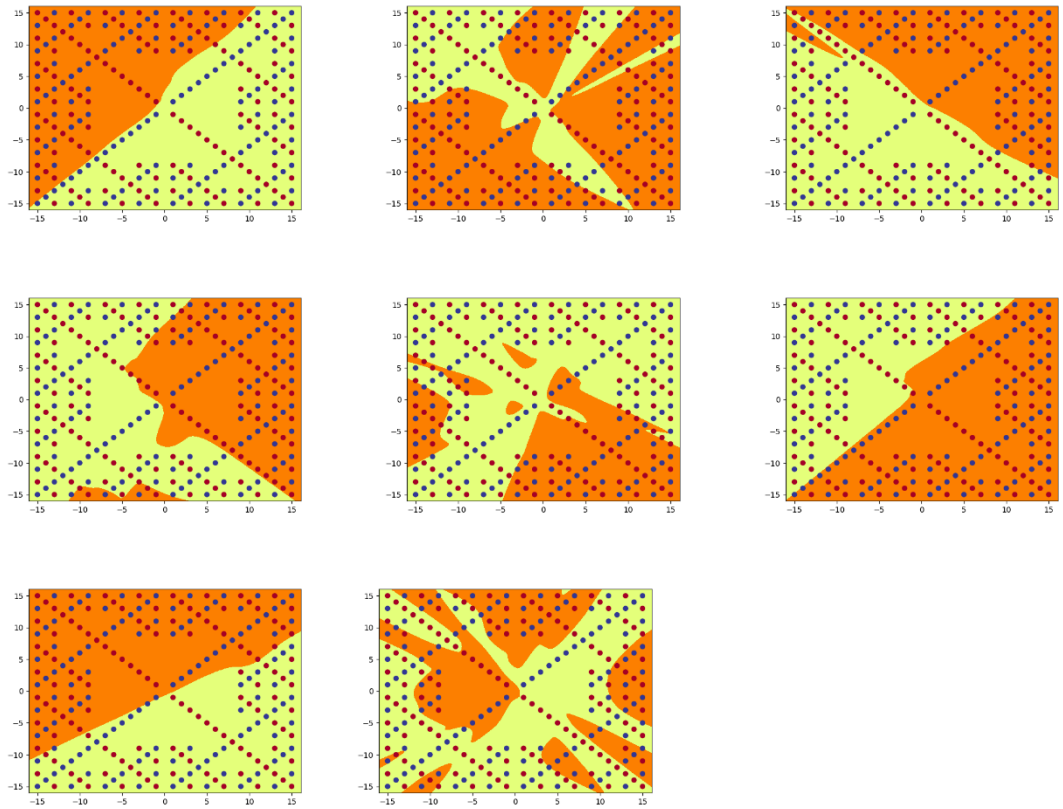
4.
hid = 20
hid1:



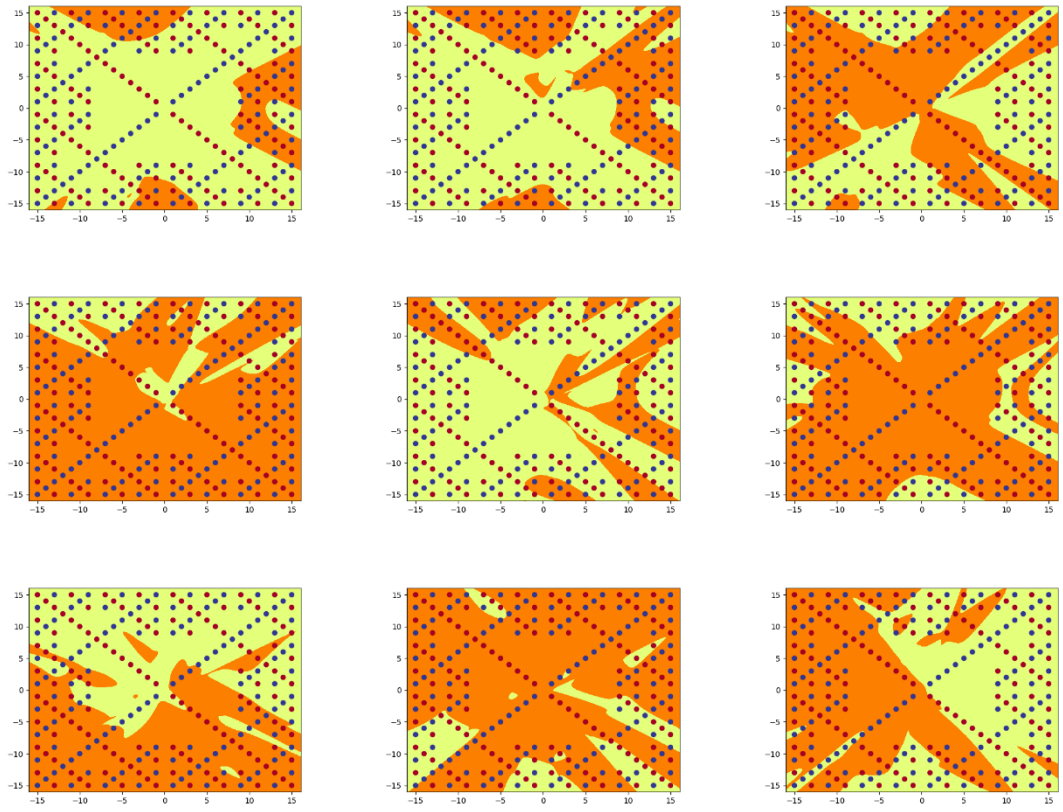


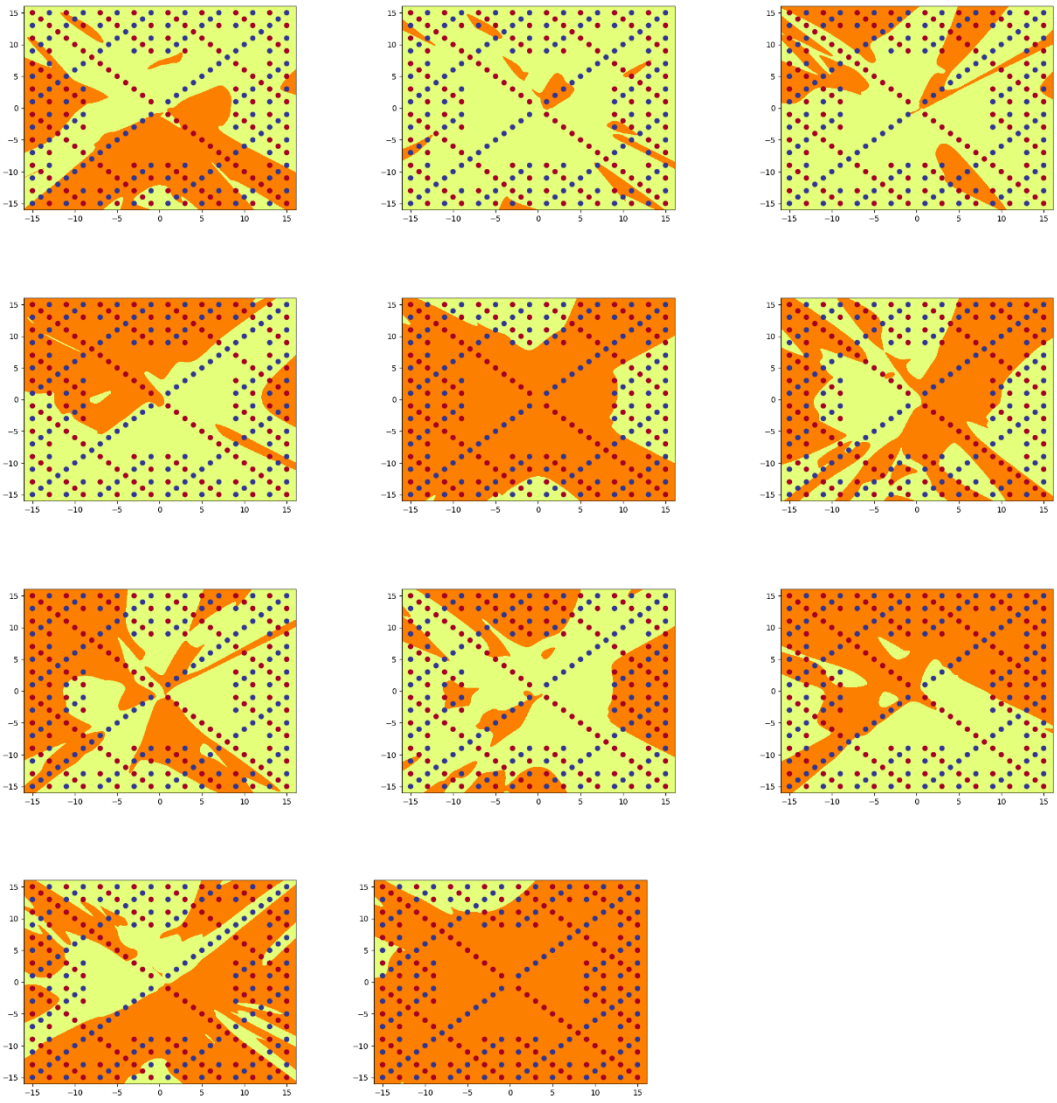
hid2:



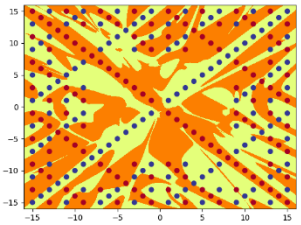


hid3:



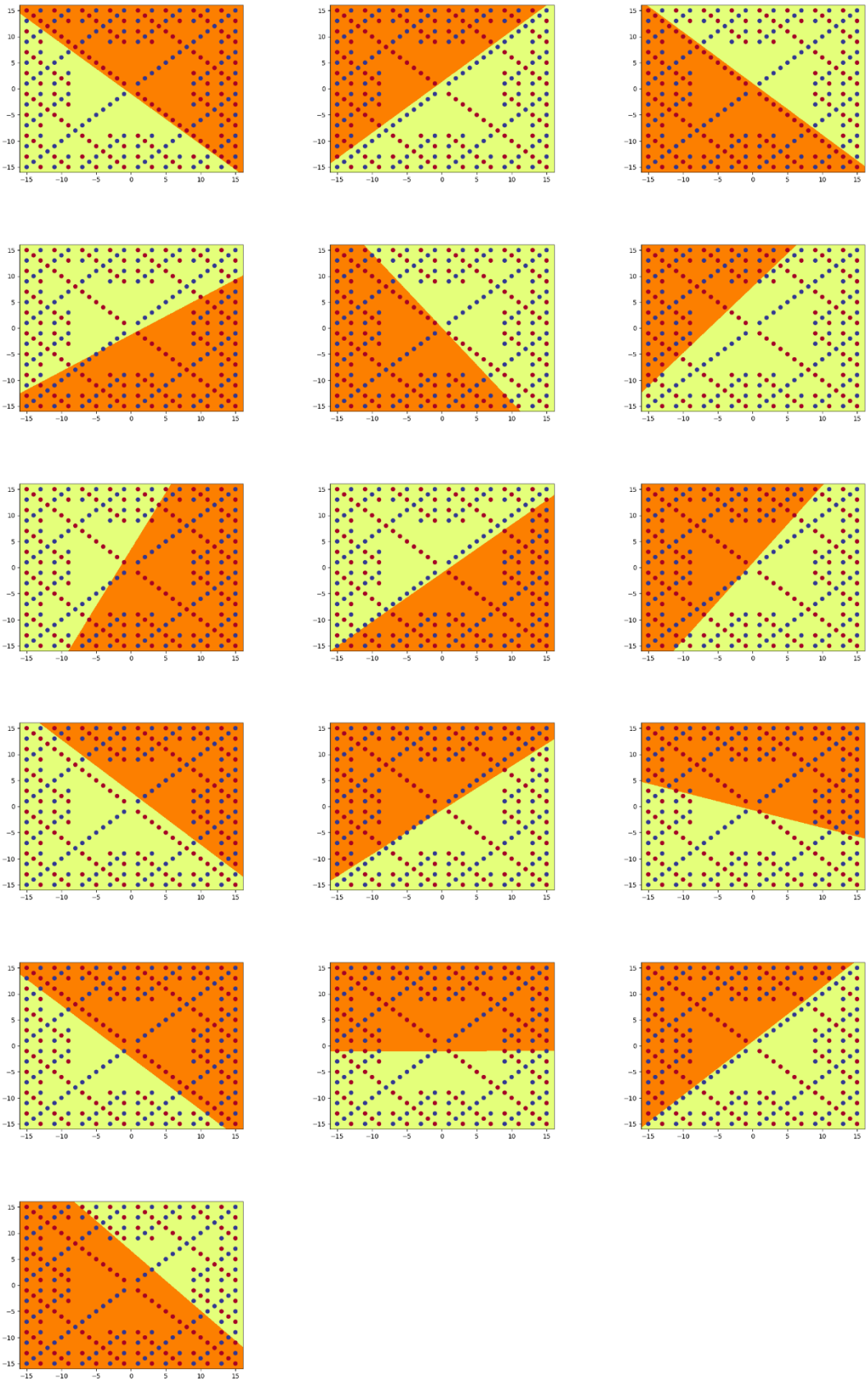


Output:

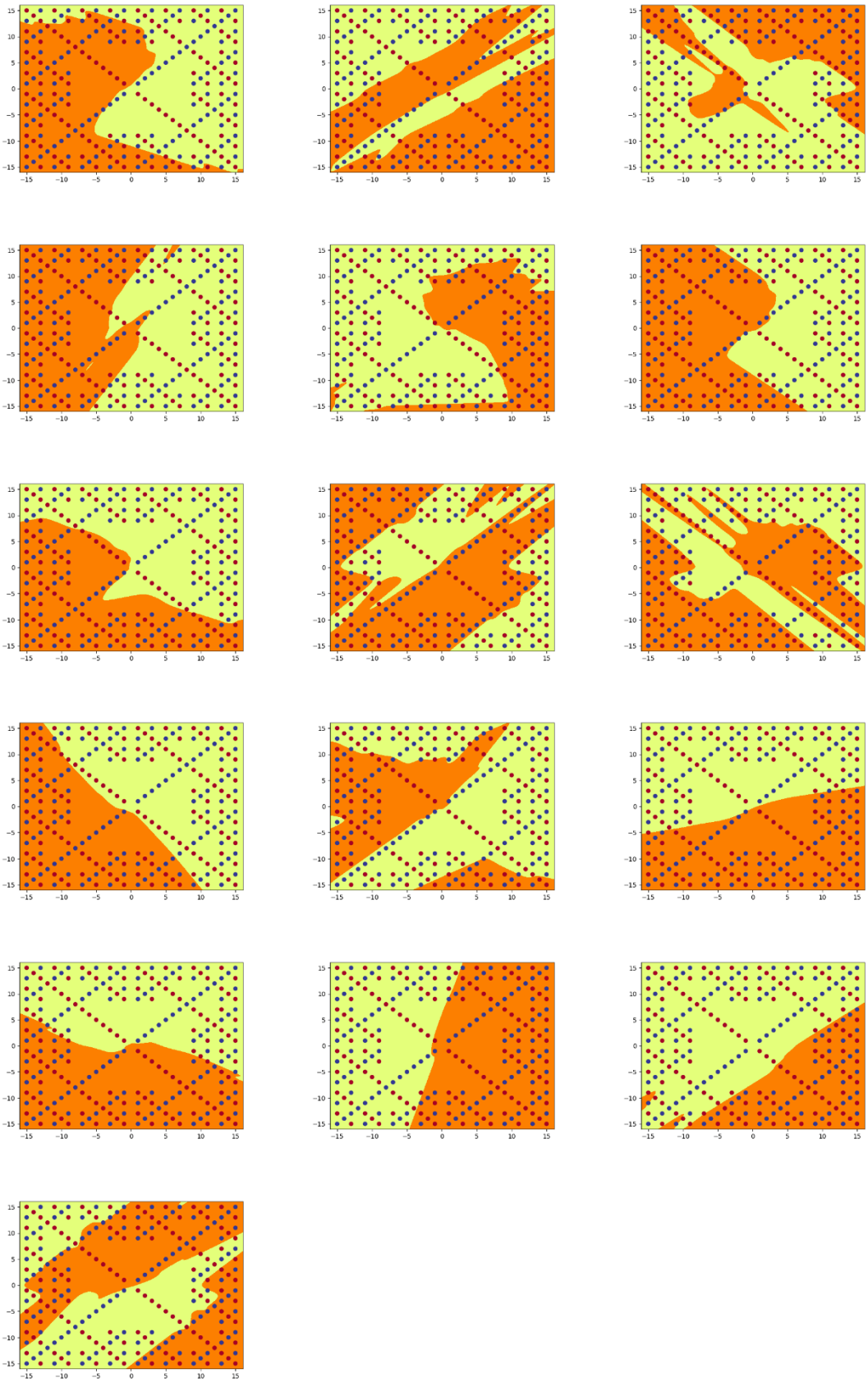


The number of independent parameters: 2*20+20*20+20*20+20*1=860

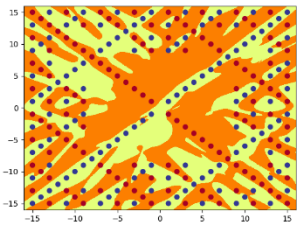
6.
hid = 16
hid1:



hid2:



Output:



The number of parameters: 2*16+2*16+2*1+16*16+16*1+16*1=354

7.

a.

Number of independent parameters:

full3: 550

full4: 860

dense: 354

Number of epochs required to train the network:

full3: 168000

full4: 56200

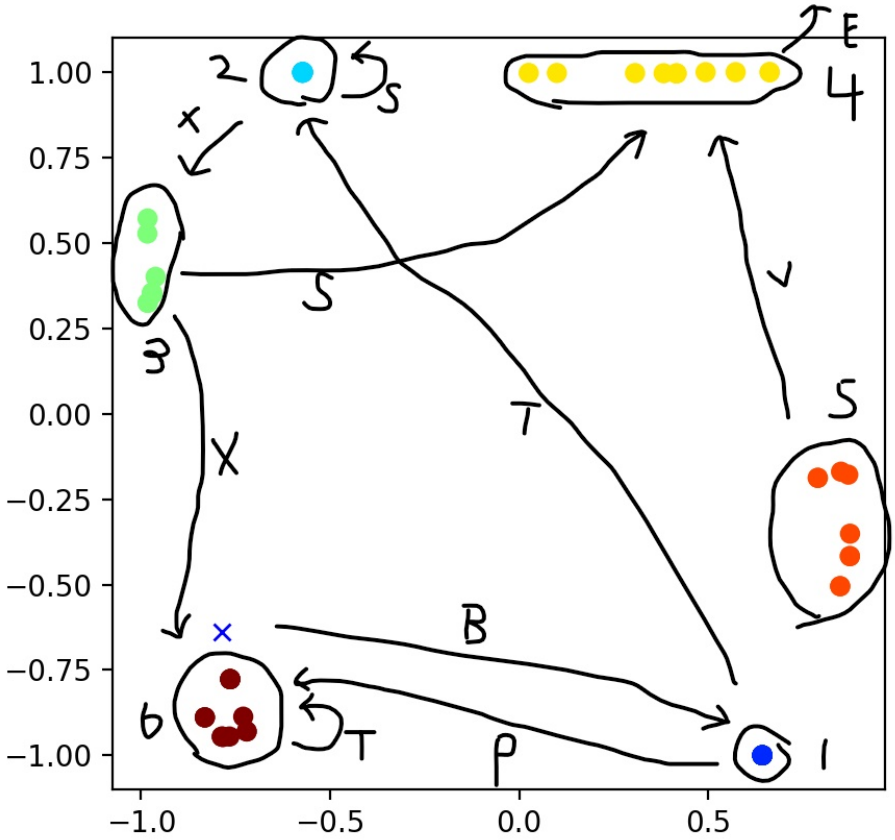
dense: 144600

b. The first layer of both full4 net and dense net are linear classifier, attempting to use a straight line to classify. The second and third layer of full4 net and second layer of dense net are non-linear, which means they try to do clarification using more complex shape and curves.

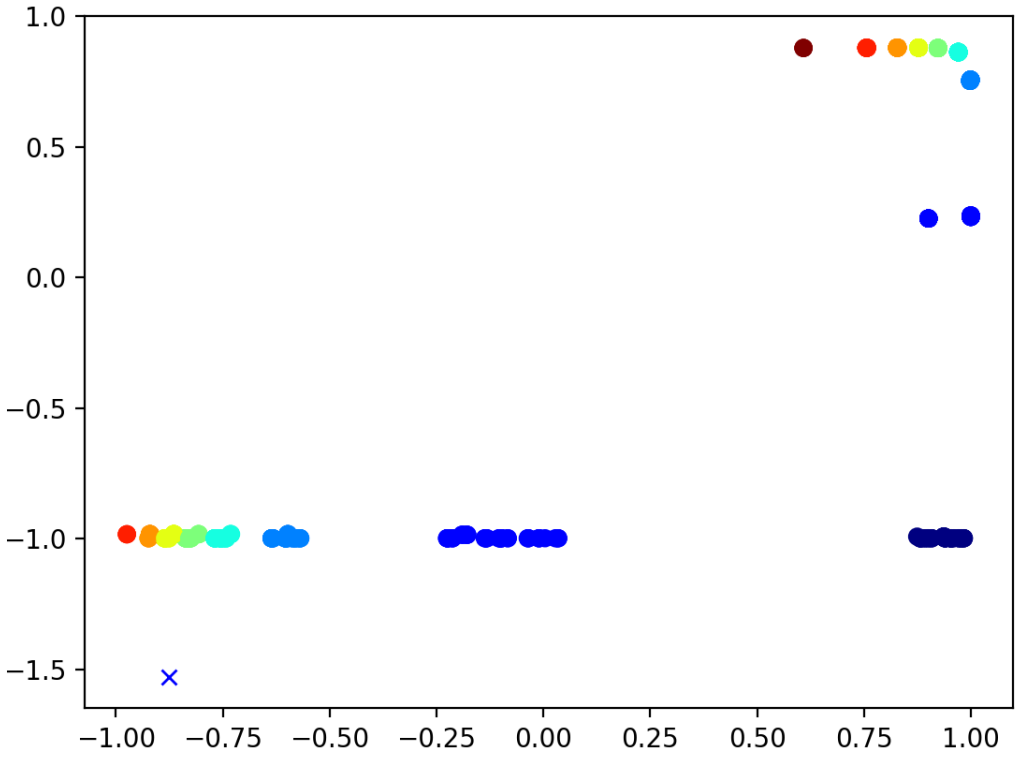
c. Though they all can do classification successfully, full3 net uses the most hidden units, while full4 use less hidden units, it have the most independent parameters. However, dense net have the least hidden units and dense parameters, which is the best.

part3

1.

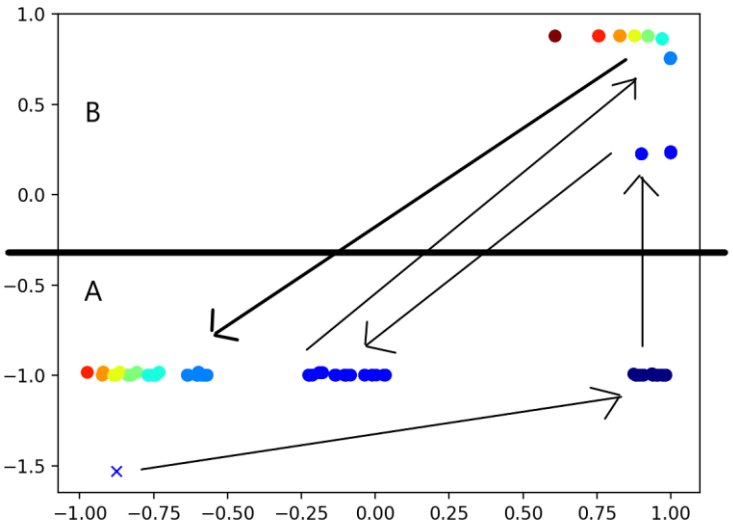


2.

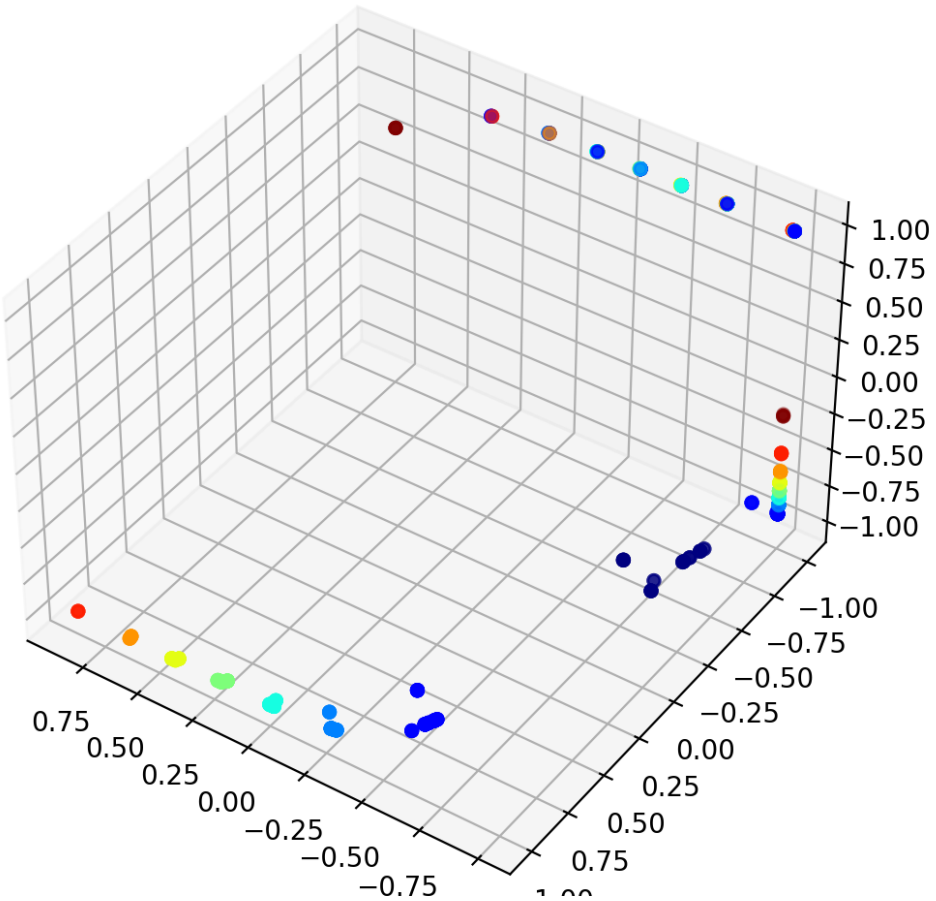


3. We can halve the picture in the middle horizontally and regard the halve

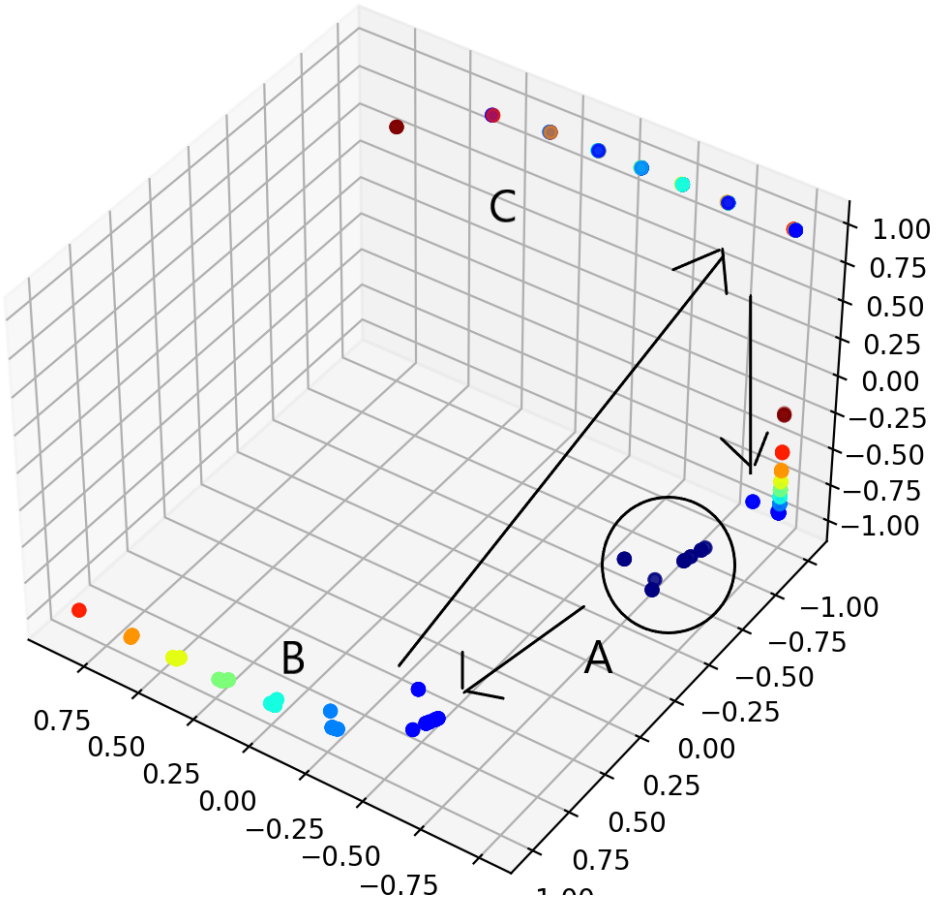
below is prediction of A, where the upper halve is prediction of B. Starting from the blue cross, we first receive some A and reach to state (1.0, -1.0). It could predict both A and B with more chance of A. After receiving a B, we switch to state (1.0, 0.25) and will self-loop to predict the same numbers of B. Then, it turns back to predict A and repeat above steps.



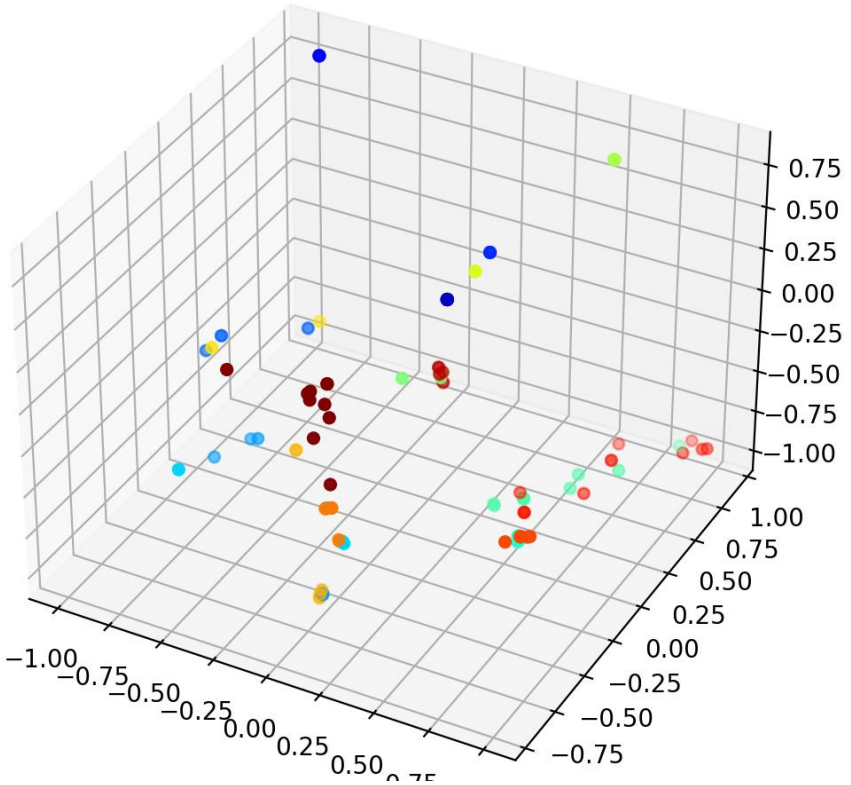
4.



5. Similarly to previous, it starts from receiving some A (the state in the circle) and predicting A and potentially B. After it receive B, it knows how many B and C to predict and goes to another state and begin to predict B and potentially C. After it receive C, it goes to the state to predict deterministic numbers of C. After that, it goes back to predict A.



6. As the picture showed, LSTM have gates for input, output and for forget. The forget gate is used to block memory we do not want, and input and output gates are used to convert and keep value in hidden state. Then hidden state will be passed into next iteration. In this way it memorize previous letters.


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
−0.75
−0.50
−0.25
0.00
0.25
0.50