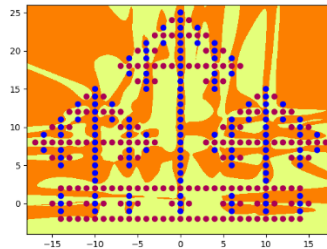


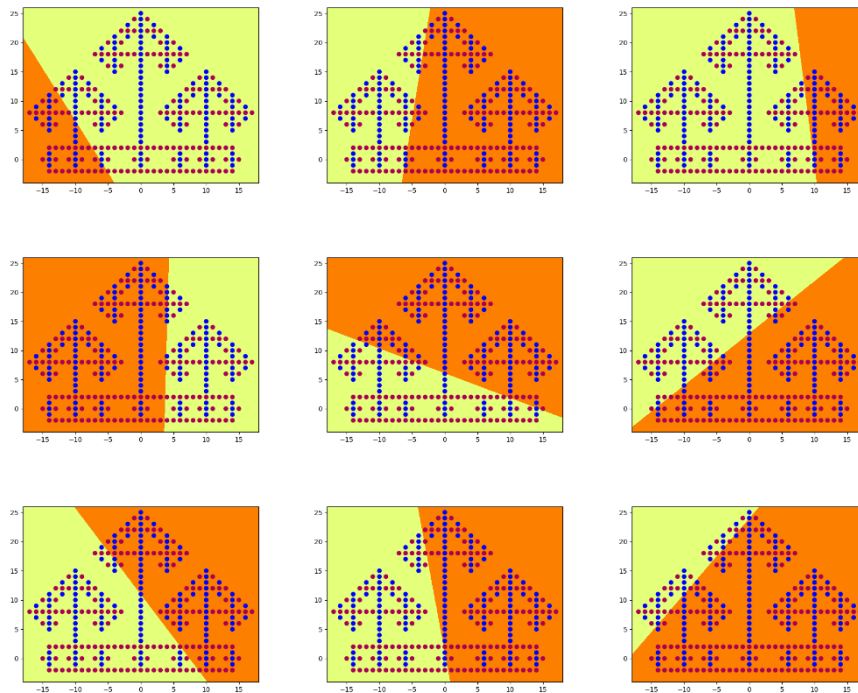
Part 1: Fractal Classification Task

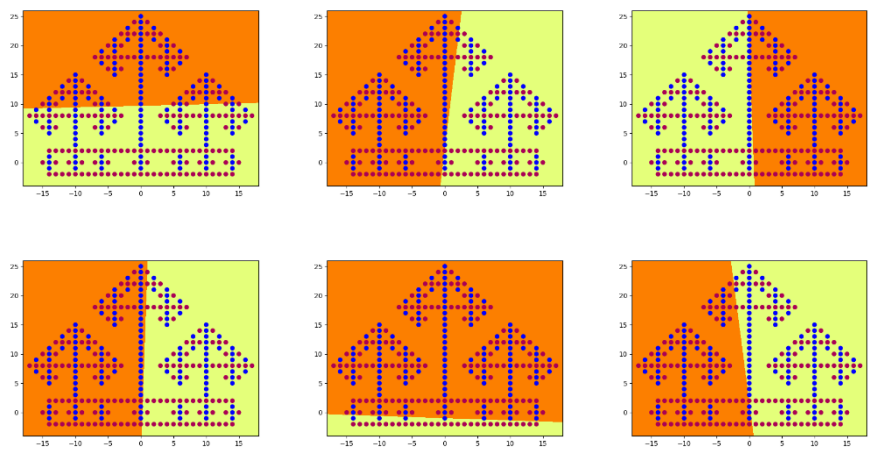
1. See the file crowd.py
2. After several tests on the number of hidden units, the minimum number of hidden nodes required is 24 (the accuracy is 99.66% when the number of hidden units was 23), at which point the classification accuracy is 100%, with the following results.



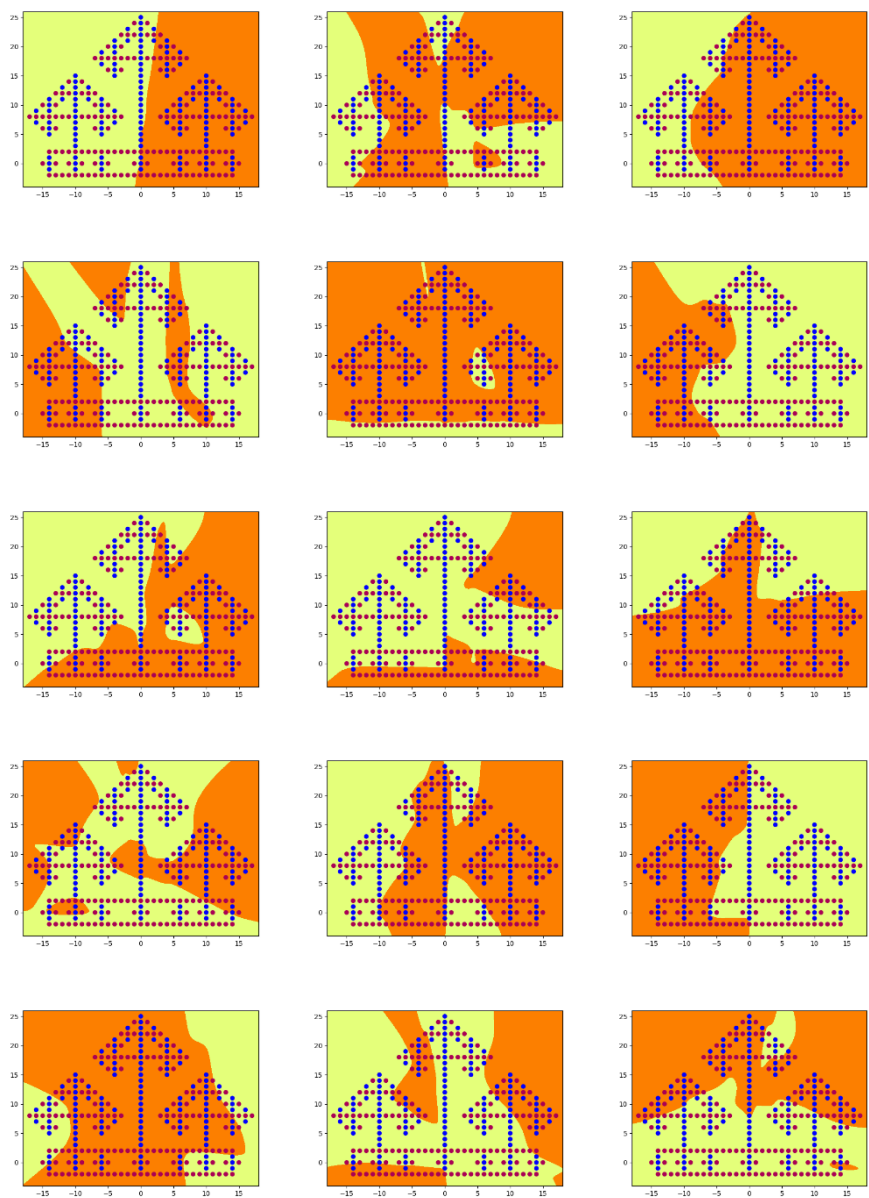
the total number of independent parameters = $2 \cdot 24 + 24 \cdot 24 + 24 \cdot 1 = 648$

3. See the file crowd.py
4. After several tests on the number of hidden units, the minimum number of hidden nodes required is 15 (the accuracy is 98.29% when the number of hidden units is 14), at which point the classification accuracy is 100%, with the following results.
Hid1:

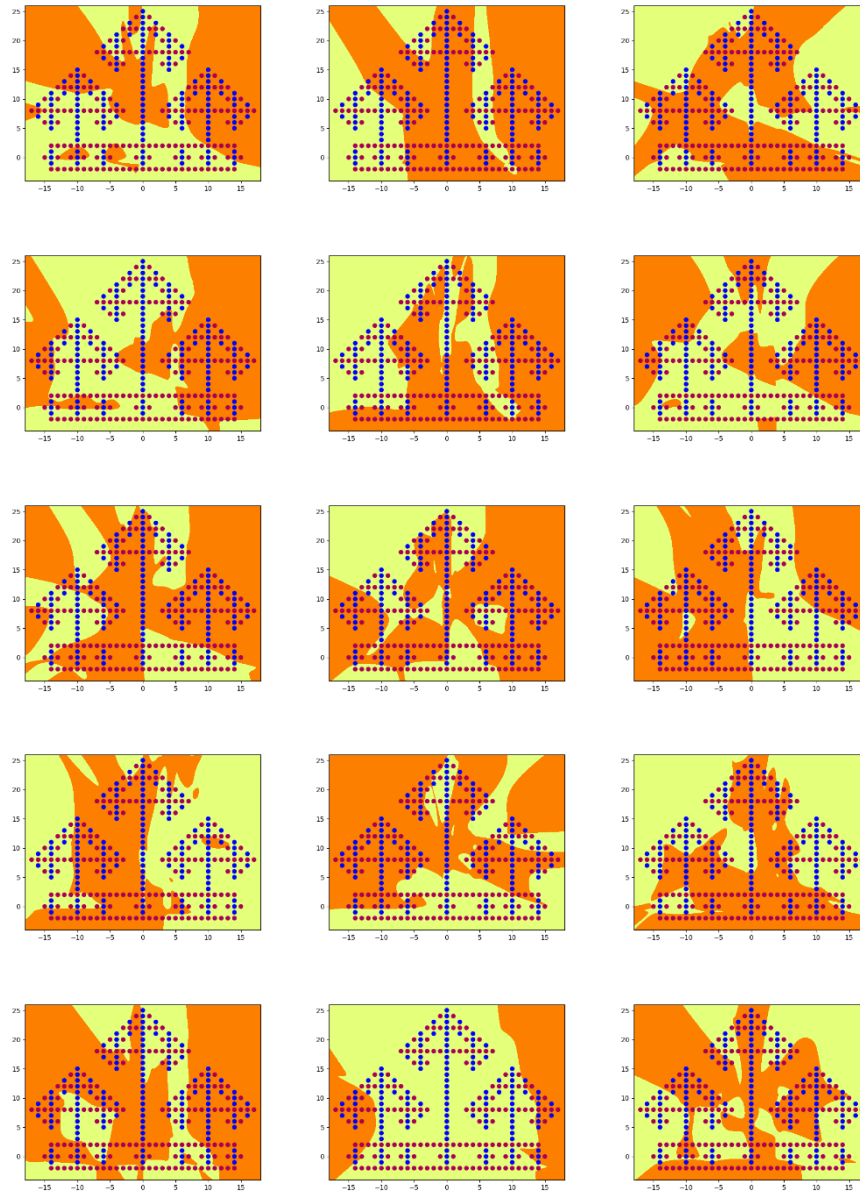




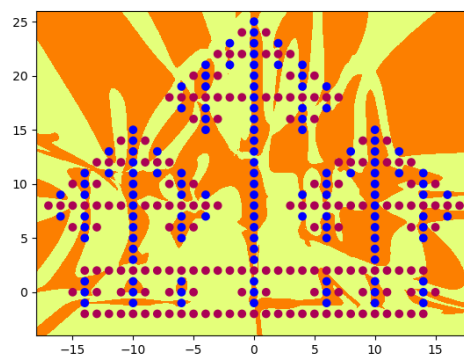
Hid2:



Hid3:



Output:

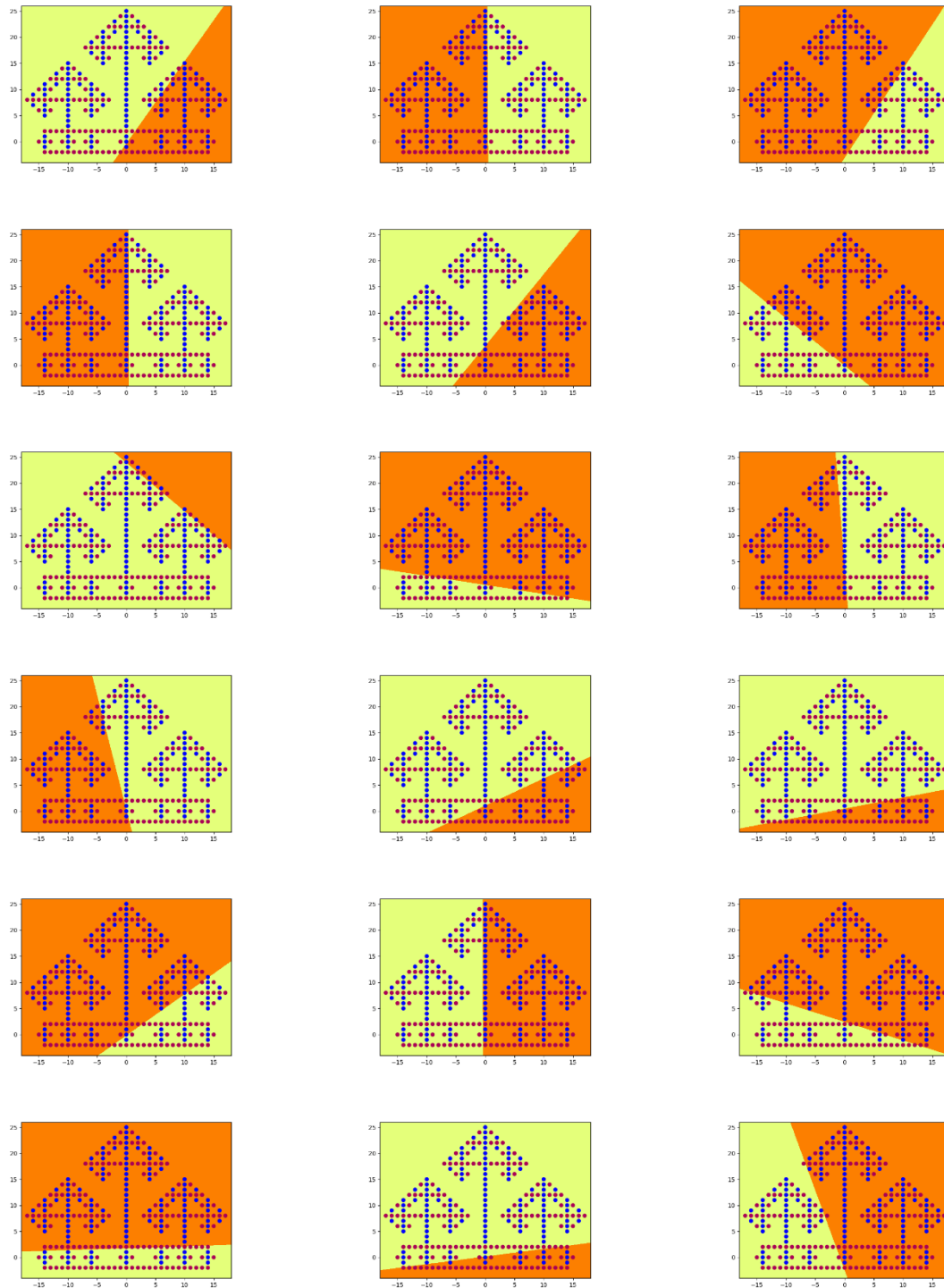


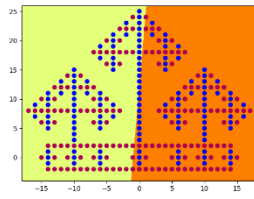
the total number of independent parameters = $2 \cdot 15 + 15 \cdot 15 + 15 \cdot 15 + 15 \cdot 1 = 495$

5. See the file crowd.py

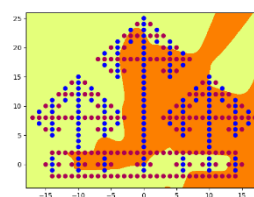
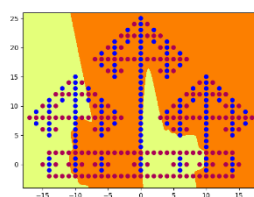
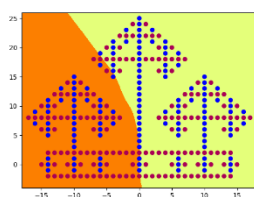
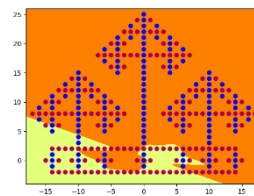
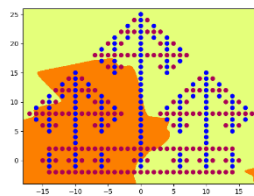
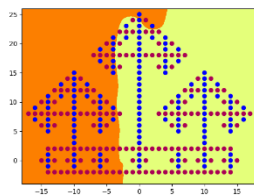
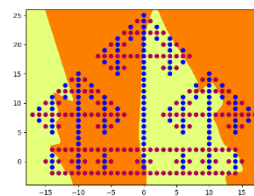
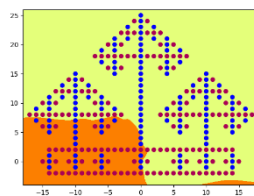
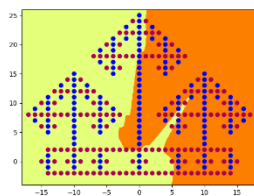
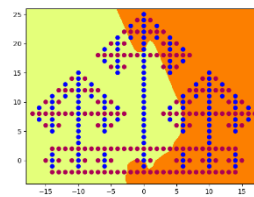
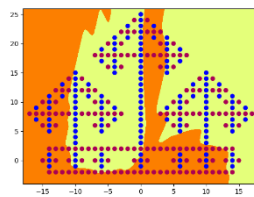
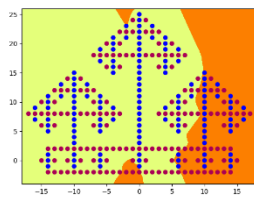
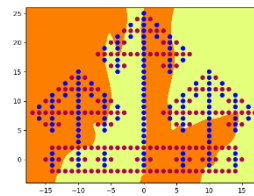
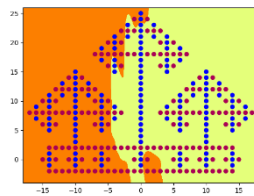
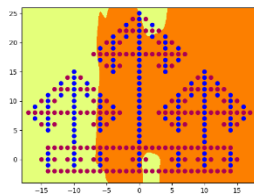
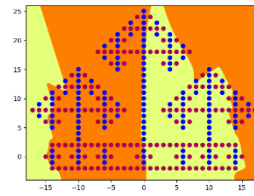
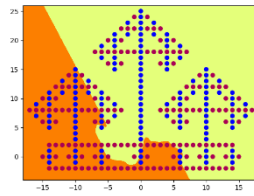
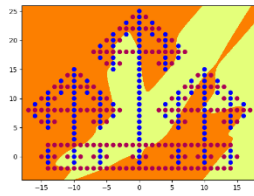
6. After several tests on the number of hidden units, the minimum number of hidden nodes required is 19 (the accuracy is 99.32% when the number of hidden units is 18), at which point the classification accuracy is 100%, with the following results.

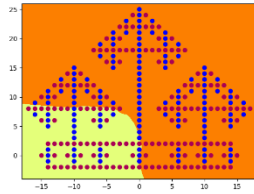
Hid1:



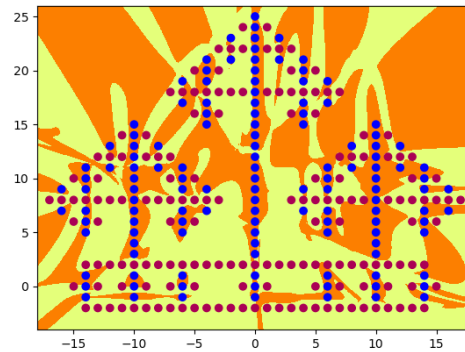


Hid2:





Output:



the total number of independent parameters = $2 \times 19 + 2 \times 19 + 2 \times 1 + 19 \times 19 + 19 \times 1 + 19 \times 1 = 513$

7.

a. the total number of independent parameters:

Full3Net:648

Full4Net:495

DenseNet:513

the approximate number of epochs required to train each type of network:

Full3Net:69400

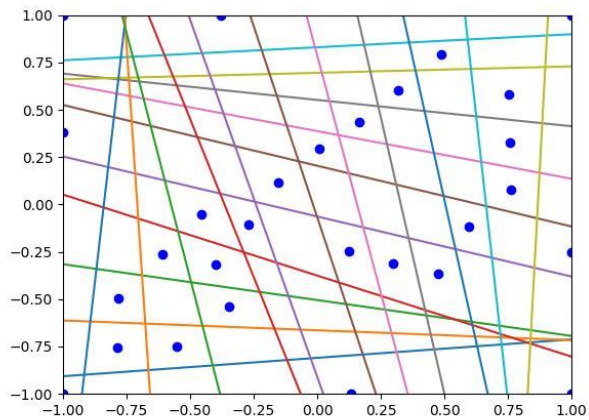
Full4Net:94100

DenseNet:46400

b. According to the result plots of each implied unit obtained in questions 4 and 6, it can be seen that the implied units in the first layer of both nets Full4Net and DenseNet are linear classifiers, while in the second layer they are non-linear classifiers.

c. All three neural networks can perform the classification task, Full3Net has both the largest total number of independent parameters and requires the largest number of training epochs, while Full4Net and DenseNet require fewer training epochs when the total number of independent parameters is close, so DenseNet is the better neural network.

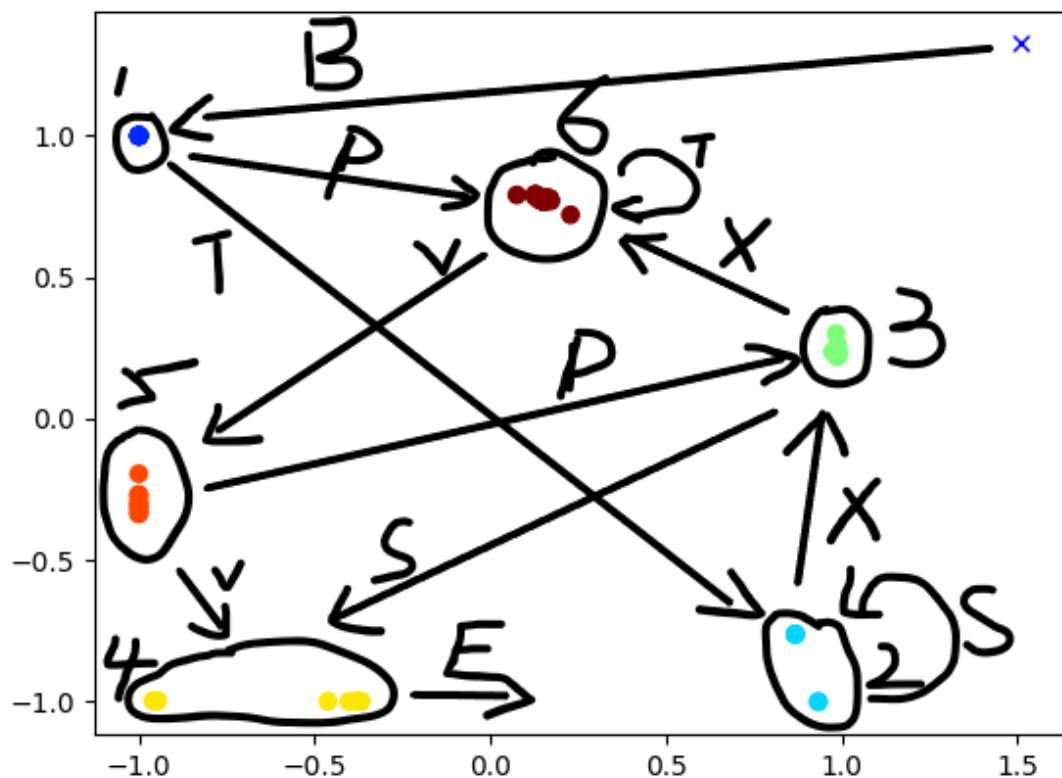
Part 2: Encoder Networks



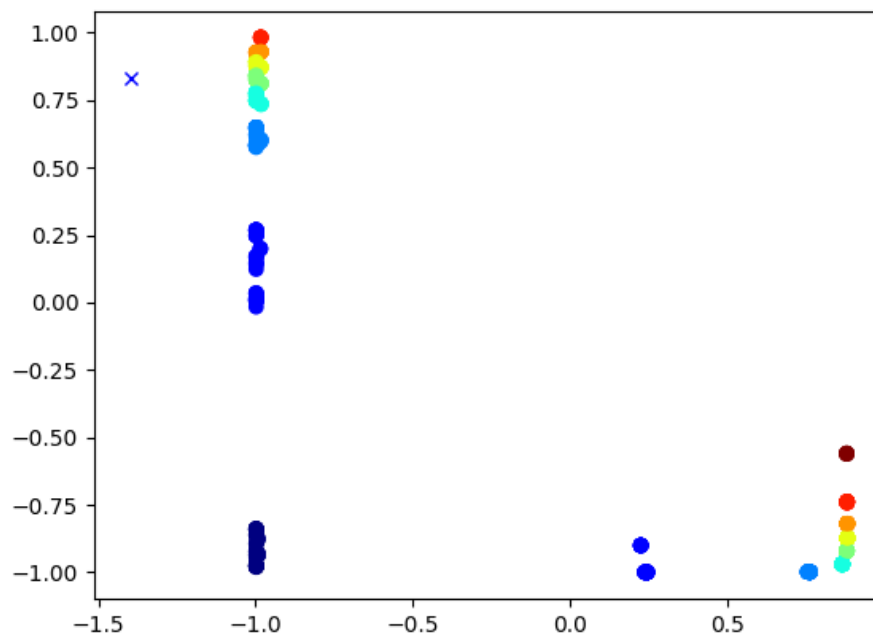
The code can be seen in the file encoder.py.

Part 3: Hidden Unit Dynamics for Recurrent Networks

1.

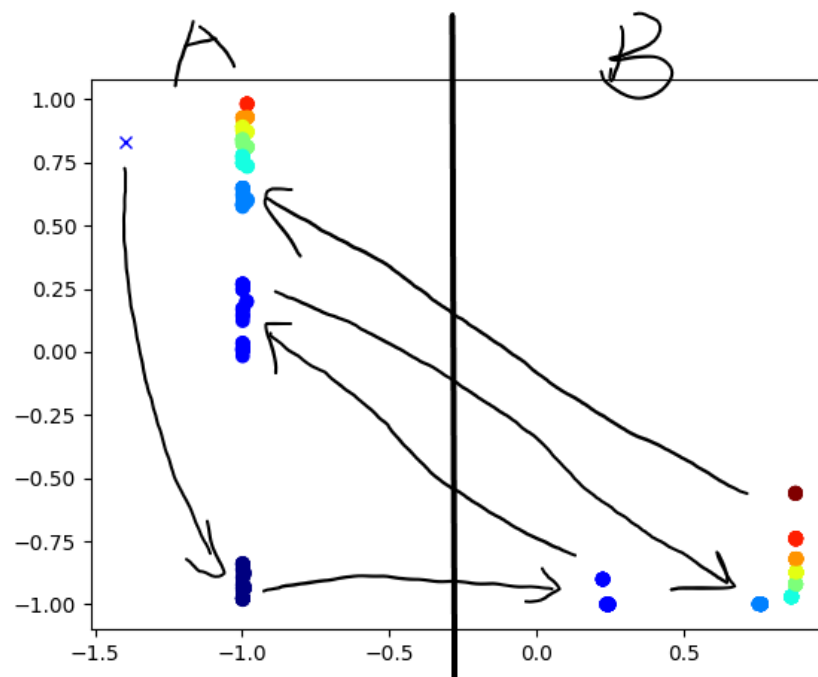


2.

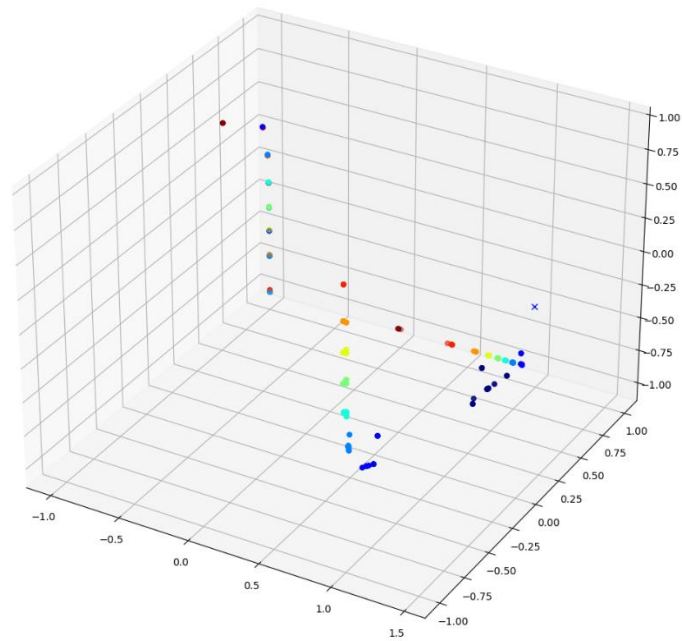


3.

Based on the graphs of the results obtained previously, it is possible to make the following analysis graphs. First, one can start by dividing the picture vertically in equal parts, with the left half being the prediction of A and the right half being the prediction of B. Starting from the blue cross, the network first reaches A's $(-1.0, -1.00)$. Then, the network will keep changing the prediction between A and B. Finally, the network will re-predict A and then continue to repeat the above steps.

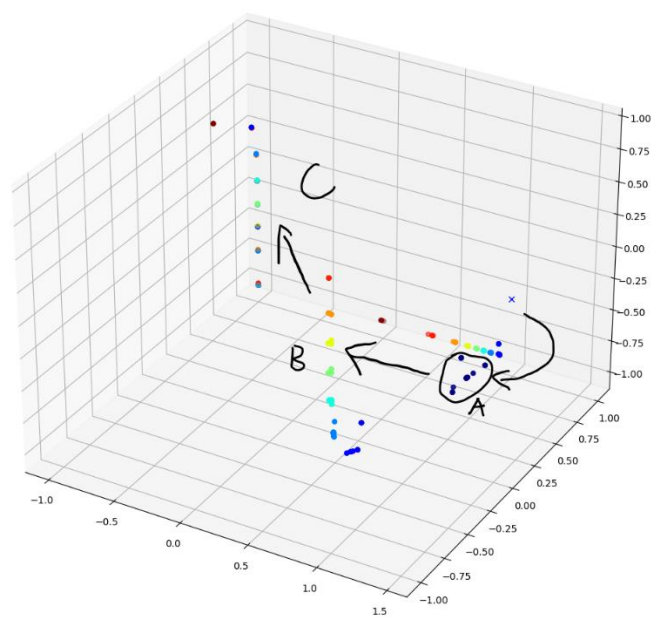


4.



5.

Similar to 3.3, it starts from receiving A 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.

The LSTM has a "gate" structure to remove or add information to the state, and there are three types of "gates" in the LSTM, forgetting gates, input gates and output gates. First, the network decides which information and states are not needed through the forgetting gates, then the input and output gates decide which information and states are to be input and output and transported and processed, and the results obtained from the output gates are added to the implied states, and the network continues to iterate in this way.

