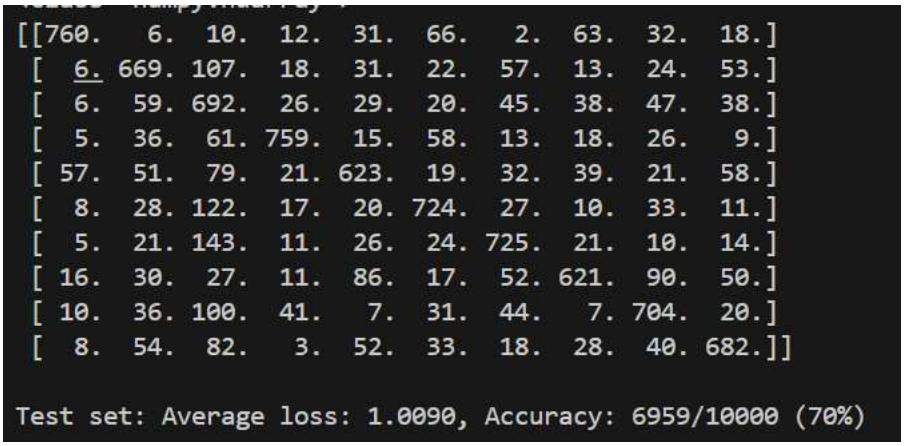
COM9444 Project1

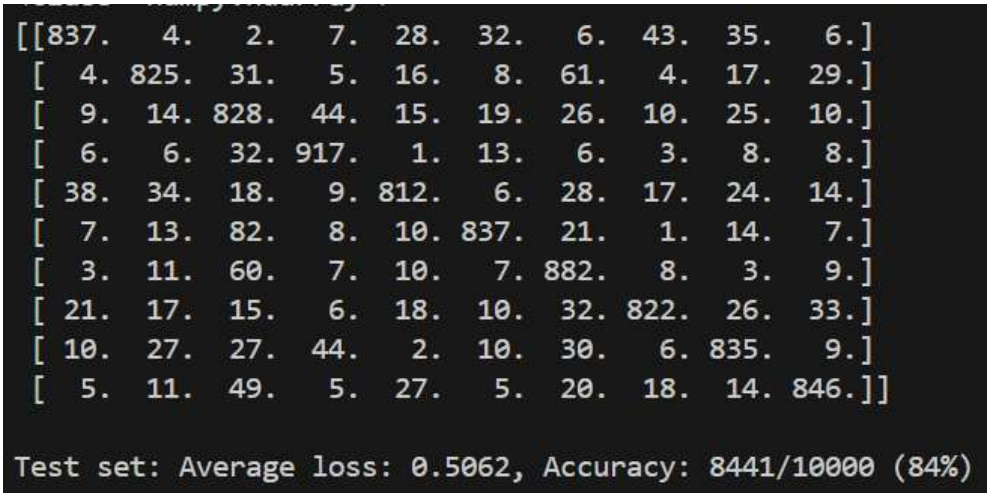
a.1

This is the screen shot for the confusion matrix and final accuracy



a.2

I choose 150 hidden nodes and it achieves 84% recognition accuracy



a.3

I use two convolutional layers plus two fully connected layers with Relu function as activation function after each layer node and log softmax at the output node. At the same time, I choose max pooling to enhance feature representation.cIn the end, I get 94% accuracy rate.

```
[[955.   4.   2.   0.  21.   2.   0.  10.   3.   3.]
 [  2. 924.   4.   0.  10.   1.  36.   7.   5.  11.]
 [ 11.   5. 904.  25.   7.   6.  12.   8.   6.  16.]
 [  1.   1.  10. 968.   3.   4.   5.   1.   2.   5.]
 [ 22.   5.   2.   6. 938.   0.  12.   4.   9.   2.]
 [  7.   7.  49.   9.   4. 894.  17.   1.   1.  11.]
 [  5.   5.  12.   1.   6.   4. 958.   5.   0.   4.]
 [  4.   5.   2.   0.   3.   1.   5. 953.   6.  21.]
 [  3.  13.   5.   1.   5.   1.   7.   4. 955.   6.]
 [  6.   2.   8.   1.   8.   0.   6.   6.   7. 956.]]

Test set: Average loss: 0.2334, Accuracy: 9405/10000 (94%)
```

First convolutional layers (self.conv1)

Input channel :1

output channel :32

Convolutional kernel:3*3

Number: (1 * 32 * 3 * 3) + 32 = 288 + 32 = 320

Second convolutional layers (self.conv2)

Input channel :32

output channel :64

Convolutional kernel:3*3

Number: (32 * 64 * 3 * 3) + 64 = 18,432 + 64 = 18,496

First Fully connected layer(self.fc1)

nn.Linear(64 * 7 * 7, 128)

input feature size 64 * 7 * 7 = 3,136

(3136 * 128) (weights) + 128 (biases) = 401,408 + 128 = 401,536

Second Fully connected layer(self.fc2)

nn.Linear(128, 10)

(128 * 10) (weights) + 10 (biases) = 1,280 + 10 = 1,290

Total Parameters = (params of conv1) + (params of conv2) + (params of fc1) + (params of fc2)

Total Parameters = 320 + 18,496 + 401,536 + 1,290

Total Parameters = 421,642

a.4

(1)Relative Accurtacy

The model Netlin achieve 70%accuracy rate

The model NetFull achieve 84%accuracy rate

The model NetConv achieve 93%accuracy rate

We can find that the accuracy rate is higher because of the improvement of model.

(2)Parameter Number

The model Netlin have 784*10+10 parameters

The model NetFull have 784*150+150+150*10+10 parameters

The model NetConv have 421642 parameters(shown above)

(3)Confusion Matrix

NetLin： The model frequently confuses character (2) with (3),(6)with(3),(7)with(3),(9) with(3).We find that the structure is simple and cannot extract spatial/local features. The chatacter(3) is always confused because (3) don't have obvious "global feature"(above 100 is recognized as confused recognition.

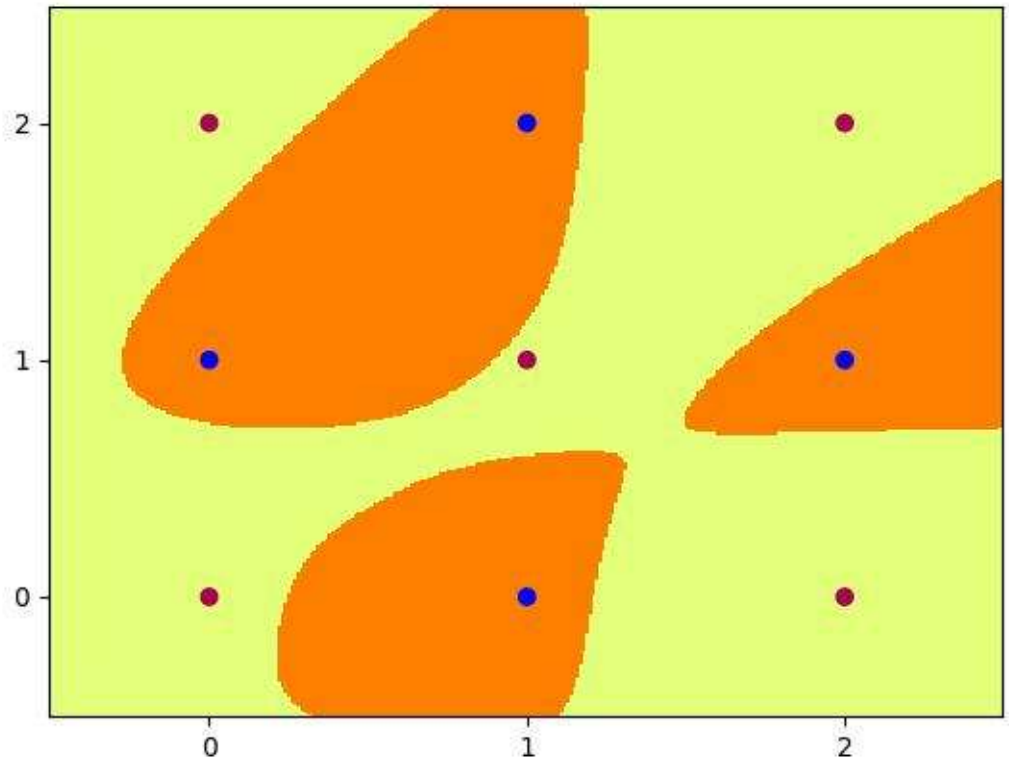
NetFull: The model introduces nonlinear activation (tanh) and hidden layers, enhancing its fitting ability. But The model frequently confuses characters(2) with (7),(6)with(3),(7)with (3).Because the model can only learn global features. So (2) and (3),(9)and (3) confusion recognition rate lower down.(above 50 is recognized as confused recognition.

NetConv:The model can extract global and partial features. Although partial features have been extracted, there are still limitations. We can see that most of time the character can be recognized well, but sometimes confuse characters(2) with character(7),confused

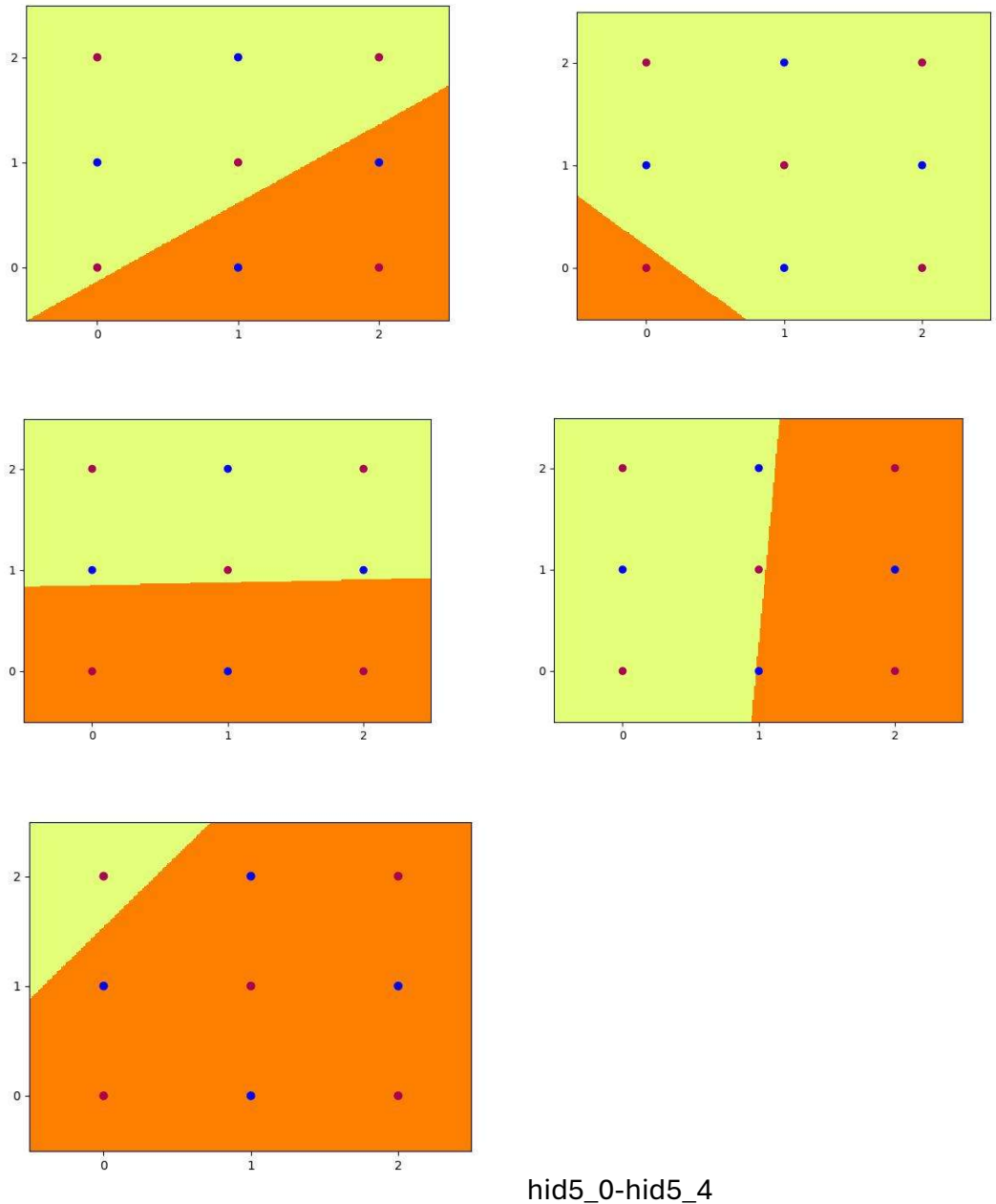
character(6) and character(3),because (6) and (3) have same part, which might be recognized as the same character.

(b).1 We trained a 2-layer neural network with 5 hidden nodes using sigmoid activation functions at both the hidden and output layers. After several runs, the network reached 100% accuracy. The hidden units learned different linear boundaries that together cover the positive class. The final output correctly classifies all training points.

```
Final Weights:
tensor([[ 6.1273, -8.2085],
        [-6.3099, -6.4143],
        [ 0.1266, -4.6791],
        [ 5.4285, -0.3799],
        [ 6.9754, -5.3298]])
tensor([-1.0732,  1.3610,  3.9791, -5.3208,  8.1937])
tensor([[ 7.2493, -5.5165, -5.9690, -7.0194,  7.0316]])
tensor([-2.9322])
Final Accuracy:  100.0
```



Out.5



hid5_0-hid5_4

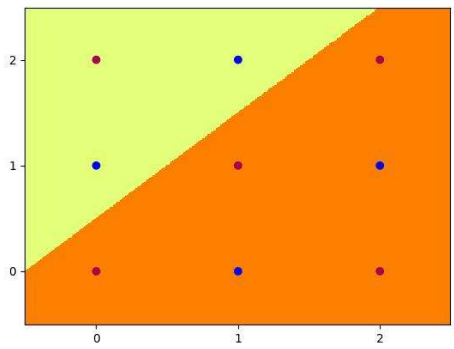
(b).2

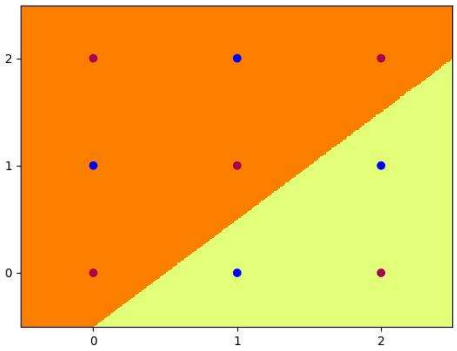
We manually designed a 2-layer network with 4 hidden units using Heaviside (step) activation. The network classifies all 9 points correctly.

Hidden Node	Weight (w1, w2)	Bias (b)	Equation
H1	(1, -1)	0.5	x - y + 0.5 = 0
H2	(-1, 1)	0.5	-x + y + 0.5 = 0
H3	(1, 1)	-1.5	x + y - 1.5 = 0
H4	(-1, -1)	2.5	-x - y + 2.5 = 0

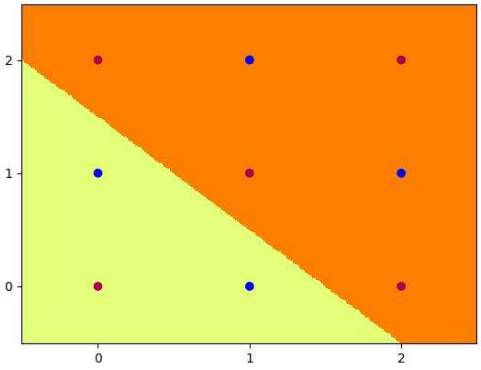
(b).3

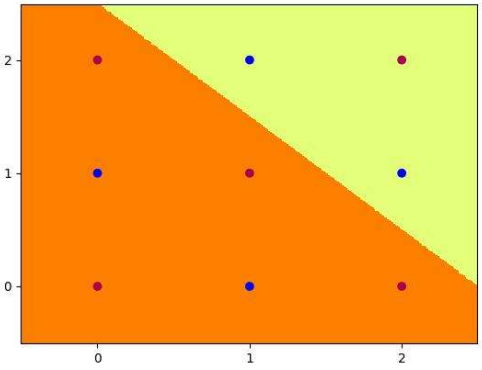
We rescaled all weights and biases from Part 2 by a factor of 10. This caused the sigmoid activation to approximate the step function behavior, allowing the network to achieve 100% accuracy



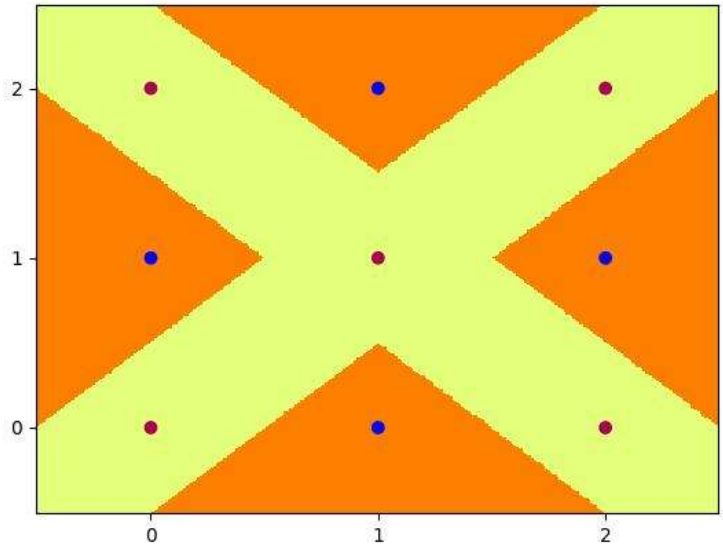


.





Hid_4_0-hid4_3

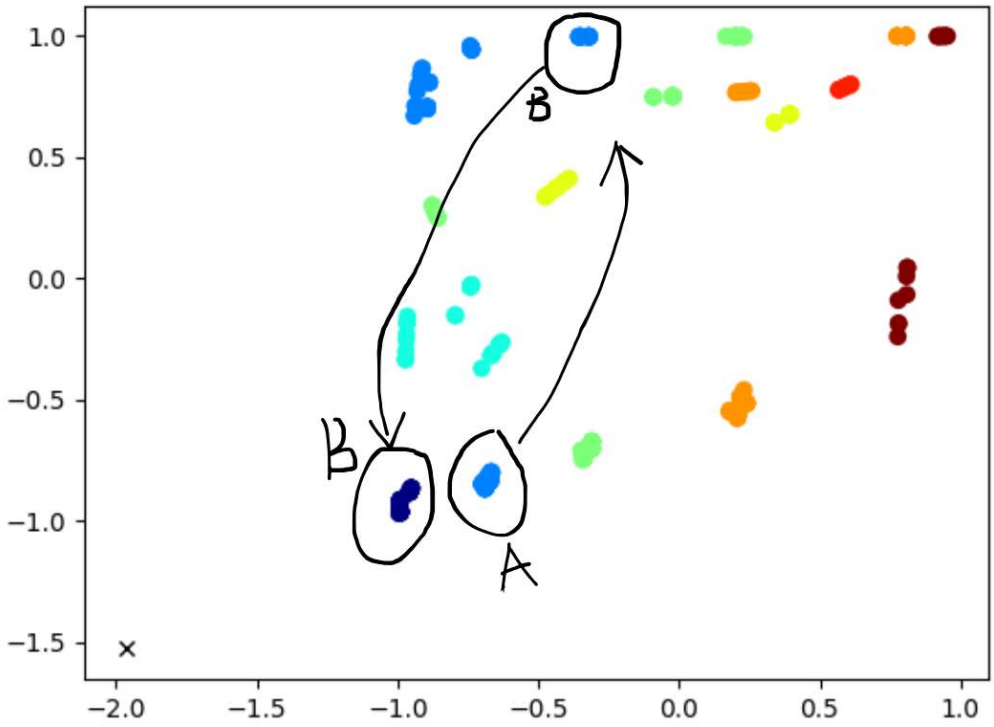


Out_4

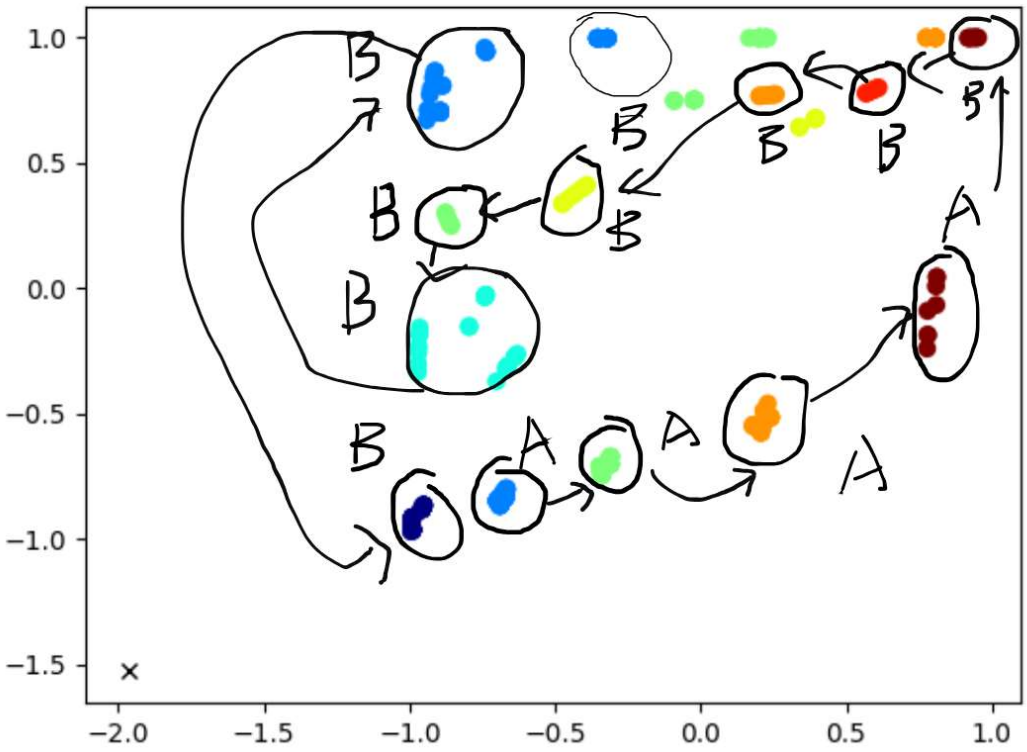
C.1

```
symbol= ABBAAAABBBBBBBBABBAABBBBAABBBBA
hidden activations and output probabilities:
A [-0.67 -0.83] [0.81 0.19]
B [-0.32  1.  ] [0. 1.]
B [-0.95 -0.86] [0.83 0.17]
A [-0.69 -0.86] [0.84 0.16]
A [-0.31 -0.67] [0.62 0.38]
A [ 0.21 -0.57] [0.49 0.51]
A [0.81 0.05] [0.02 0.98]
B [0.92 1.  ] [0. 1.]
B [0.57 0.78] [0. 1.]
B [0.2  0.77] [0. 1.]
B [-0.48  0.34] [0. 1.]
B [-0.88  0.3 ] [0. 1.]
B [-0.97 -0.33] [0.13 0.87]
B [-0.91  0.87] [0. 1.]
B [-0.99 -0.97] [0.91 0.09]
A [-0.67 -0.8 ] [0.77 0.23]
A [-0.67 -0.8 ] [0.77 0.23]
B [-0.35  1.  ] [0. 1.]
B [-0.96 -0.88] [0.85 0.15]
A [-0.69 -0.86] [0.84 0.16]
A [-0.31 -0.68] [0.63 0.37]
B [0.2 1. ] [0. 1.]
B [-0.66 -0.31] [0.12 0.88]
B [-0.74  0.95] [0. 1.]
B [-0.99 -0.96] [0.91 0.09]
A [-0.67 -0.8 ] [0.78 0.22]
A [-0.34 -0.74] [0.72 0.28]
B [0.22 1.  ] [0. 1.]
B [-0.64 -0.27] [0.09 0.91]
B [-0.74  0.94] [0. 1.]
B [-0.99 -0.96] [0.9 0.1]
epoch: 9
 loss: 0.0300
```

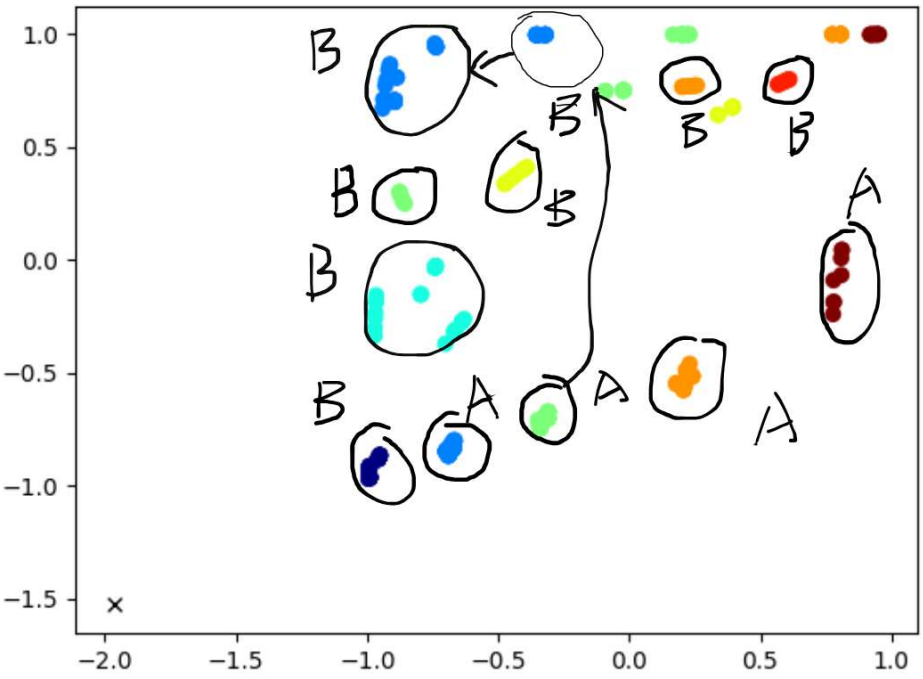
First prediction ABB



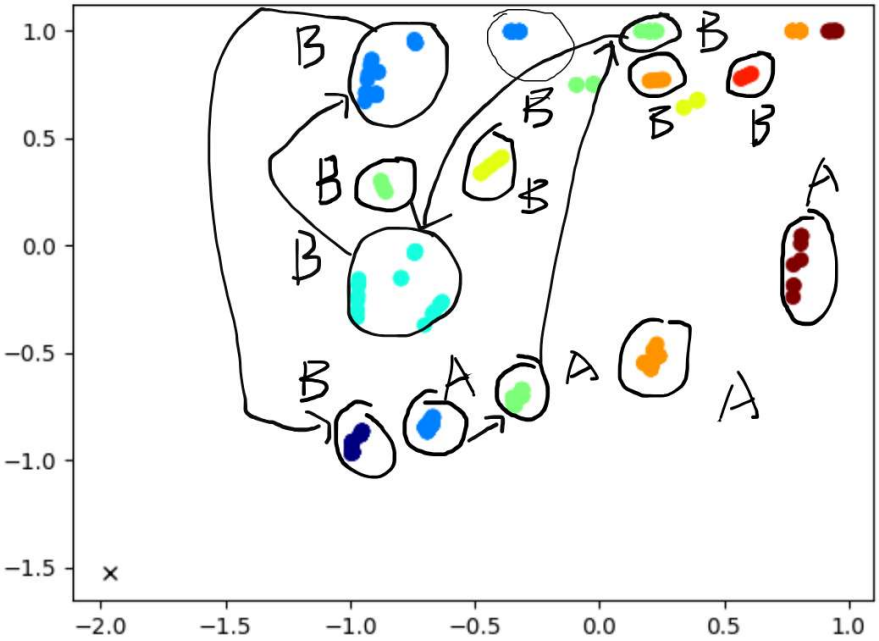
Next prediction AAAABBBBBBBB



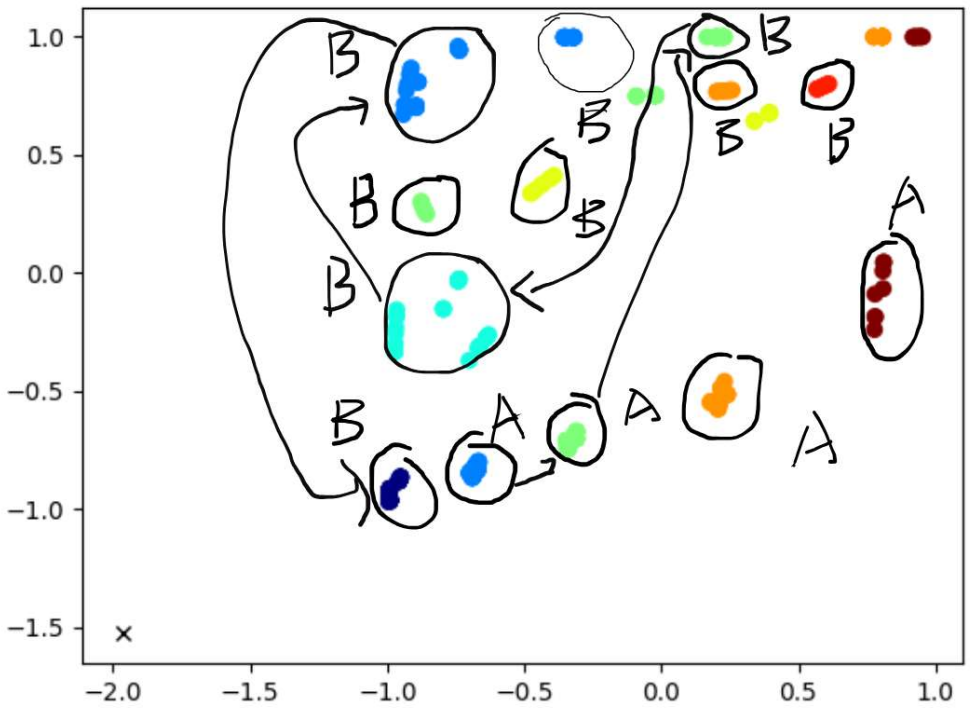
Next prediction ABB



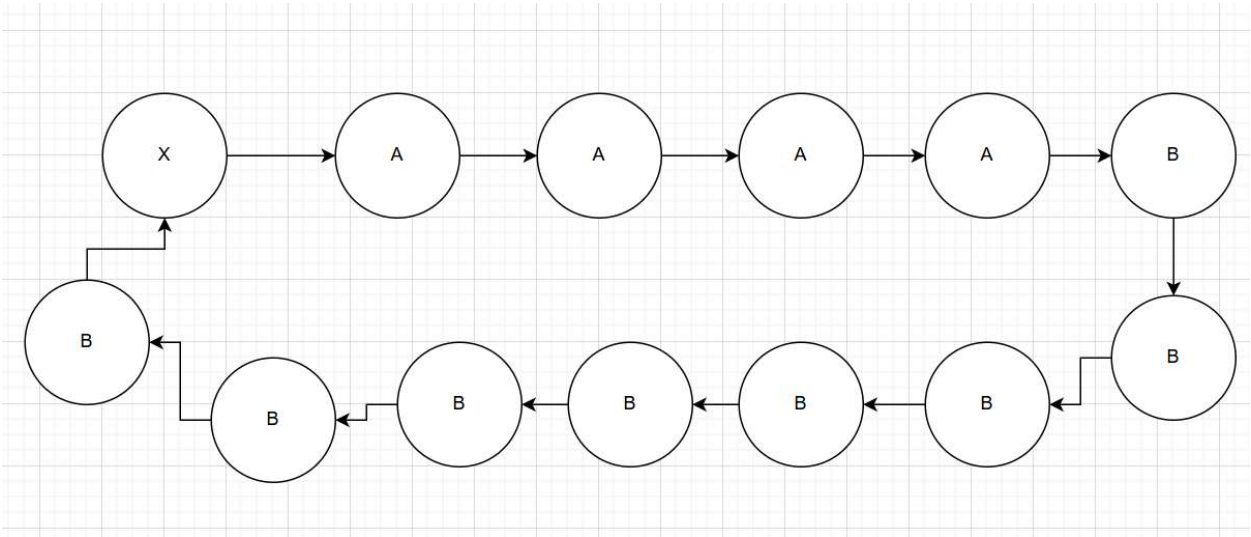
Next prediction AABBBB



next prediction AABBBB



C.2



C.3

When reading in A, this is the value encoded for n; upon encountering the first B, it is known how many B's are encoded based on the number of A's. Before the last B, the network already knows how many B's are left, thus it can make an accurate prediction; after reading the last B, the network 'knows' that the sequence has ended, and will predict the starting A of the next sequence.

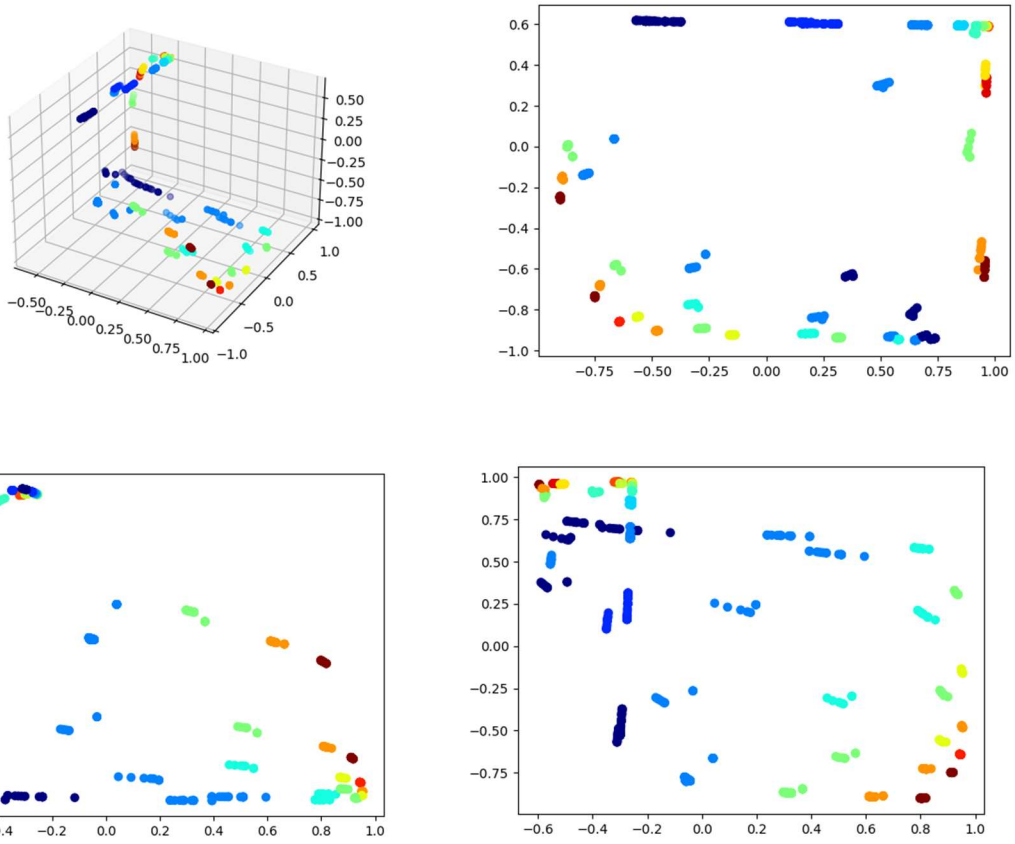
C.4

```
symbol= AAAABBBBBBBBCCCCCCCCCCCCAABBBBCCCCCCAAAABBBBBBBBCCCCCCCCCCCCAAABBBBBBCCCCCCCCCAAABBBBBBCCCCCCCCCA
hidden activations and output probabilities:
A [ 0.04 -0.66  0.04] [ 0.76  0.23  0.01]
A [ 0.37 -0.85 -0.05] [ 0.57  0.43  0.  ]
A [ 0.66 -0.89 -0.16] [ 0.27  0.73  0.  ]
A [ 0.82 -0.9  -0.26] [ 0.13  0.87  0.  ]
B [ 0.92 -0.75 -0.74] [ 0.01  0.99  0.  ]
B [ 0.95 -0.64 -0.86] [ 0.  1.  0.]
B [ 0.95 -0.48 -0.9 ] [ 0.  1.  0.]
B [ 0.95 -0.16 -0.92] [ 0.  1.  0.]
B [ 0.94  0.3  -0.94] [ 0.  1.  0.]
B [ 0.83  0.57 -0.95] [ 0.    0.99  0.01]
B [ 0.39  0.65 -0.95] [ 0.    0.88  0.12]
B [-0.37  0.72 -0.95] [ 0.    0.09  0.91]
C [-0.59  0.95 -0.64] [ 0.  0.  1.]
C [-0.55  0.96  0.26] [ 0.  0.  1.]
```

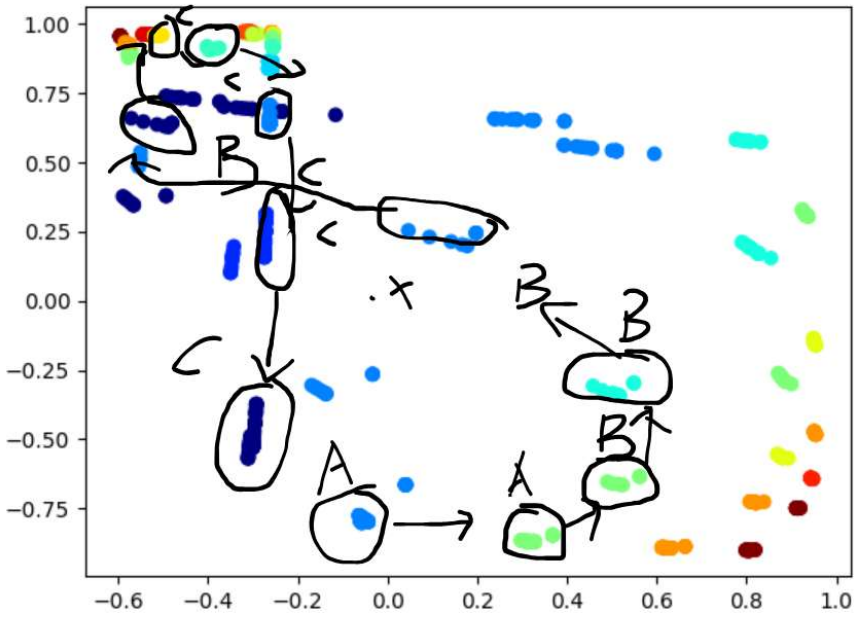
```
C [-0.59  0.94 -0.51] [ 0.  0.  1.]
C [-0.51  0.96  0.38] [ 0.  0.  1.]
C [-0.29  0.96  0.59] [ 0.  0.  1.]
C [-0.26  0.95  0.59] [ 0.  0.  1.]
C [-0.26  0.92  0.59] [ 0.  0.  1.]
C [-0.26  0.86  0.59] [ 0.  0.  1.]
C [-0.26  0.69  0.6 ] [ 0.  0.  1.]
C [-0.27  0.29  0.6 ] [ 0.03  0.    0.96]
C [-0.29 -0.4   0.61] [ 0.85  0.01  0.14]
epoch: 50000
 loss: 0.0157
```

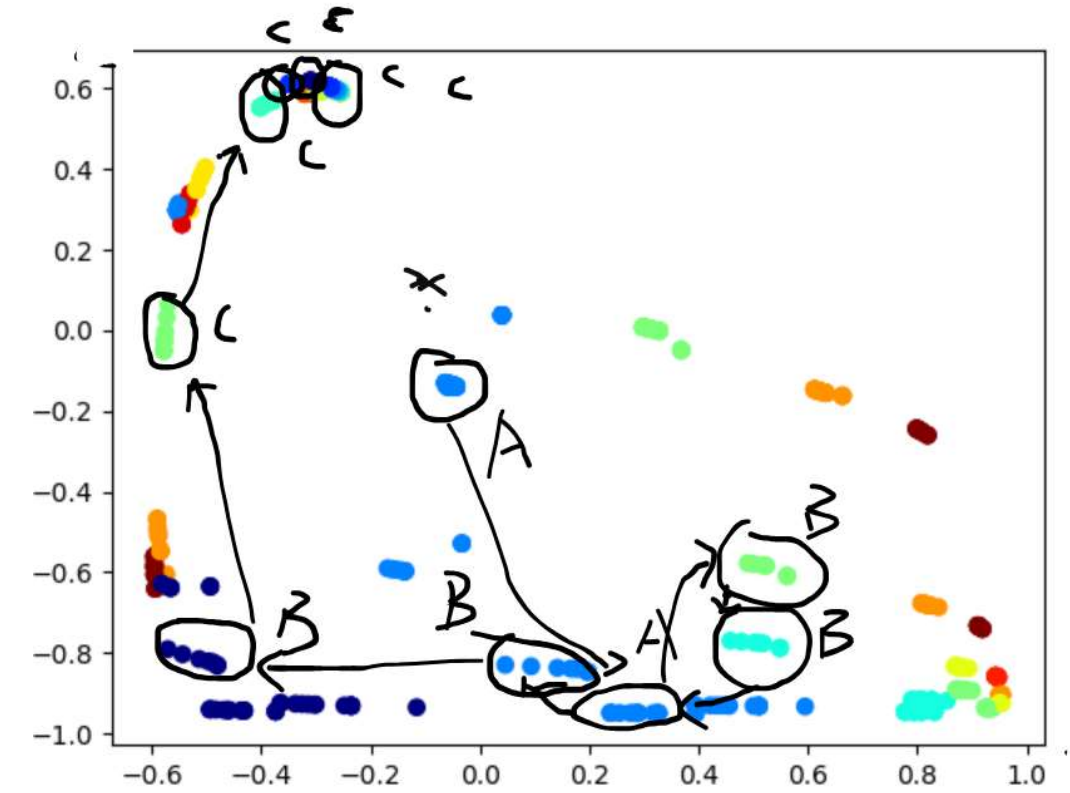
List one prediction as exampleAABBBBCCCCCC

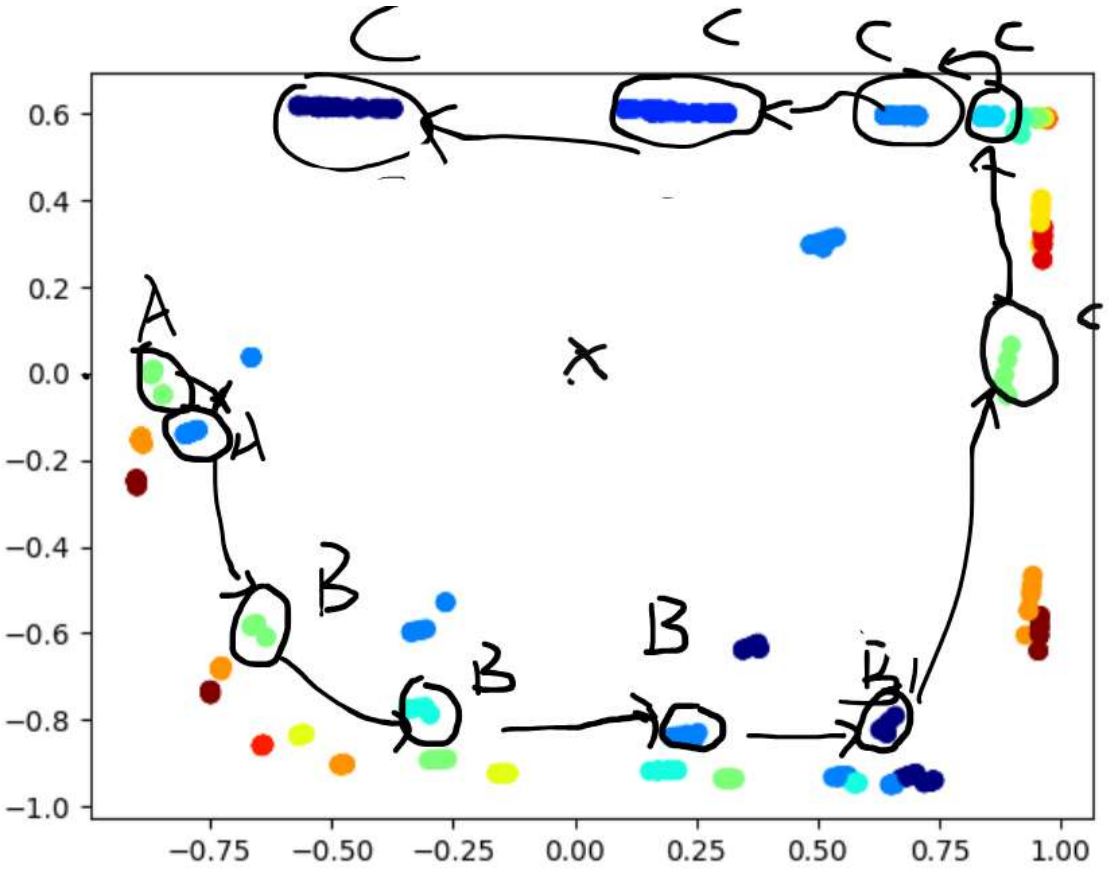
```
A [-0.05 -0.8  -0.14] [ 0.71  0.28  0.01]
A [ 0.33 -0.87 -0.  ] [ 0.66  0.34  0.  ]
B [ 0.52 -0.66 -0.58] [ 0.04  0.96  0.  ]
B [ 0.51 -0.34 -0.78] [ 0.01  0.99  0.  ]
B [ 0.17  0.2  -0.84] [ 0.    0.93  0.07]
B [-0.5   0.63 -0.82] [ 0.    0.05  0.95]
C [-0.58  0.88 -0.02] [ 0.  0.  1.]
C [-0.4   0.91  0.56] [ 0.  0.  1.]
C [-0.27  0.84  0.6 ] [ 0.  0.  1.]
C [-0.26  0.64  0.6 ] [ 0.  0.  1.]
C [-0.27  0.16  0.6 ] [ 0.08  0.01  0.91]
C [-0.3  -0.52  0.62] [ 0.93  0.01  0.06]
```



(c) .5




0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

During the A-phase, hidden unit activations gradually increase in a smooth trajectory. At the B-phase, the hidden state switches sharply to another region, indicating a change in mode. In the C-phase, the hidden state follows an extended path. These trajectories encode position and phase within the sequence.

The LSTM memorizes the count of A's during the A-phase using its internal state . It then uses this information to count off exactly 2n B's and 3n C's,. After the final C, it returns to the A-region to begin a new sequence, demonstrating that it has learned the correct phase structure.