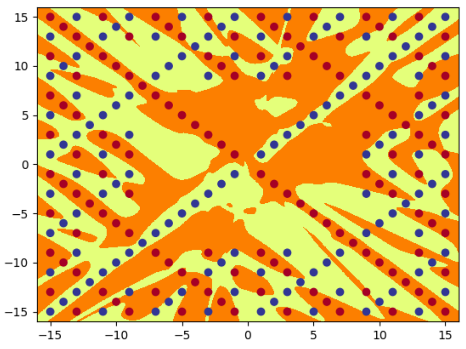
9444 Homework 1

Part 1 (1), (2)

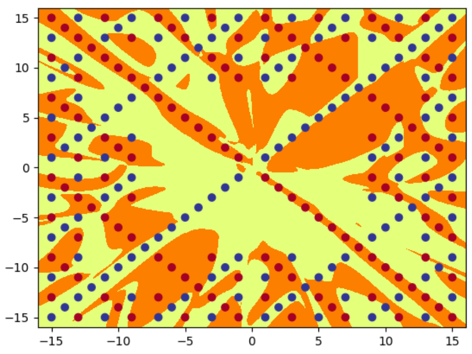


This is **out_full3_30.png**. I would choose 30 hidden nodes with accuracy 100% and loss 0.0031 finished in 24000 epochs.

Total number of independent parameters is

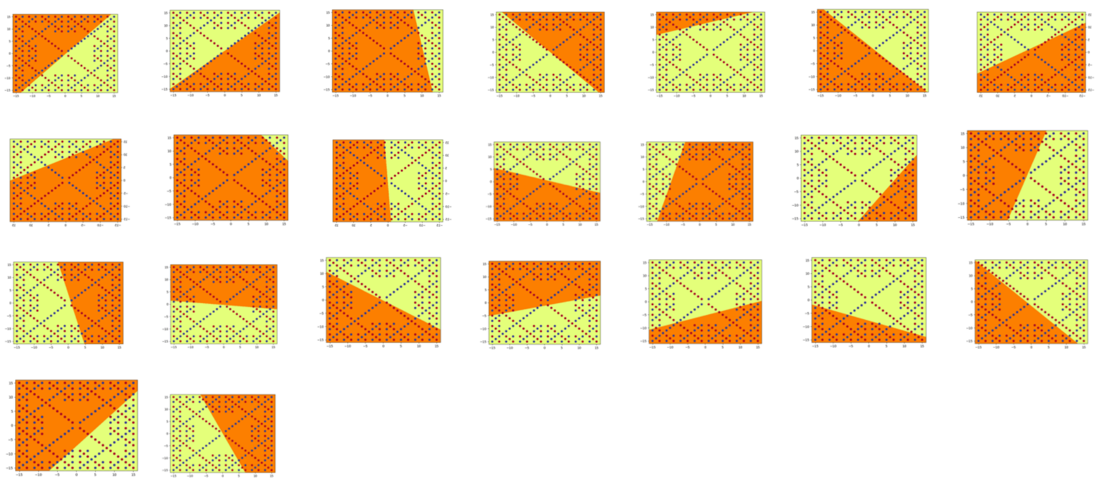
$$(2+1)\times 30+(30+1)\times 30+(30+1)\times 1=1051$$

(3), (4)

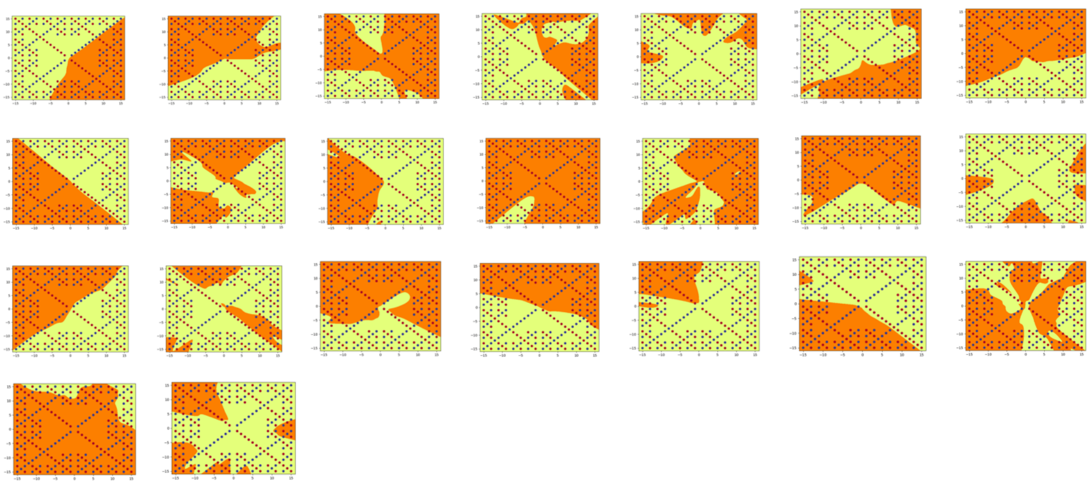


This is **out_full4_23.png** for Full4Net. I would choose 23 hidden nodes. Training finishes in 189500 epoch and loss is 0.0007.

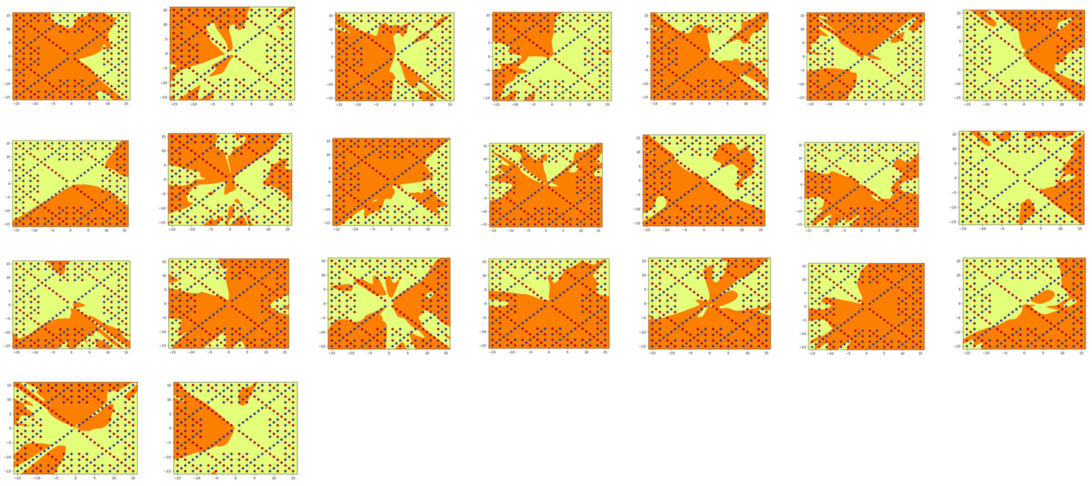
Full 4 Layer 1



Full 4 layer 2



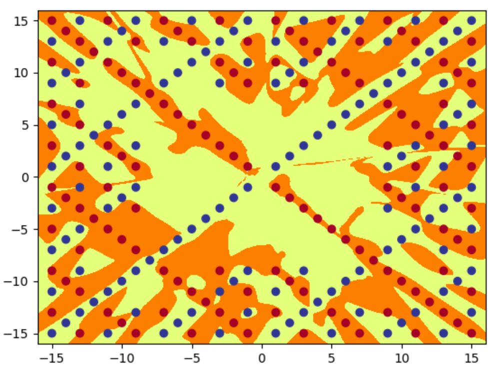
Full4Net Layer 3



Total number of independent parameters is

$$(2+1)\times 23+(23+1)\times 23+(23+1)\times 23+(23+1)\times 1=1197$$

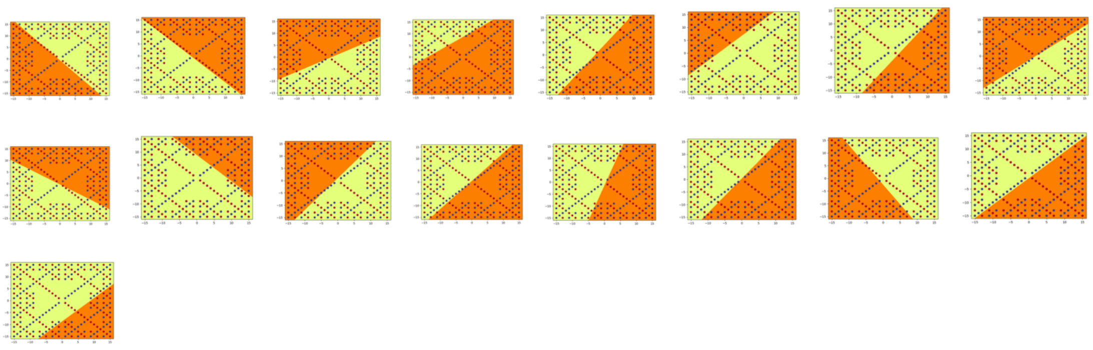
(5), (6)



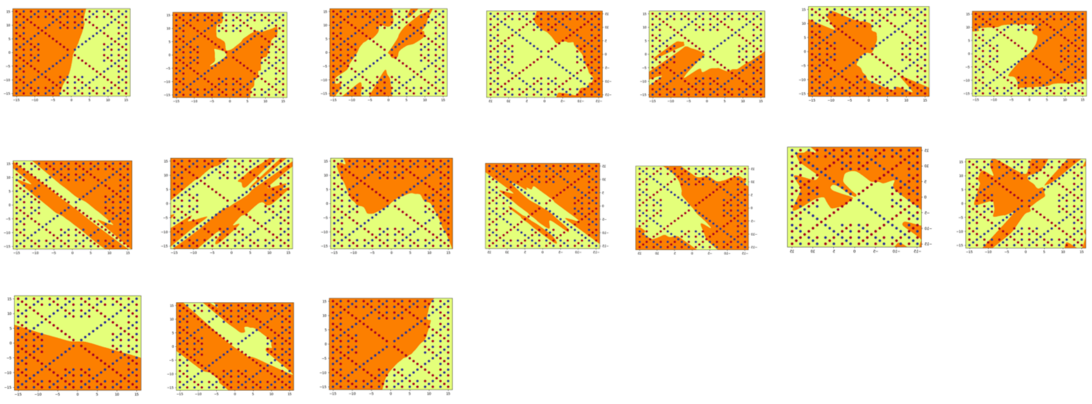
out_dense_17.png

Training finish in 85000 epochs.

DenseNet Layer 1



DenseNet Layer 2



Total number of independent parameters is

$$(2+1) \times 17 + (17+2+1) \times 17 + (17 * 2 + 2 + 1) \times 1 = 428$$

(7)

a.

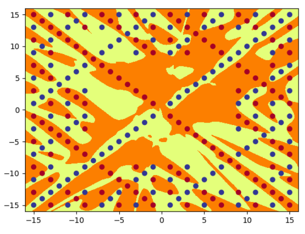
	Full3Net	Full4Net	DenseNet
Hidden nodes	30	23	17
Independent Parameter	1051	1197	428
Epoch	24000	189500	85000

b.

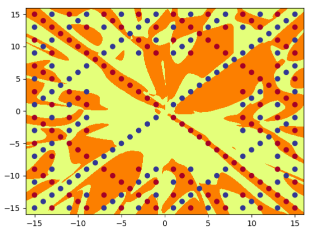
From the plots we can find that the first hidden layer can only learn linear separable functions, Layer 2 and later layers can learn convex features or more complicated functions.

c.

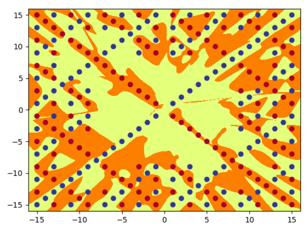
All networks distinguish dots successfully with 100% accuracy. There is no large difference. DenseNet looks more sophisticated than others.



Full3Net

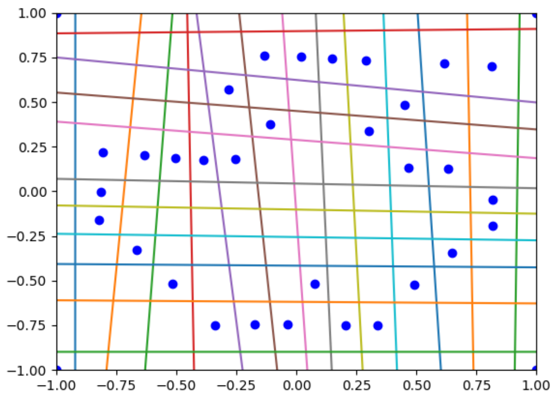


Full4Net



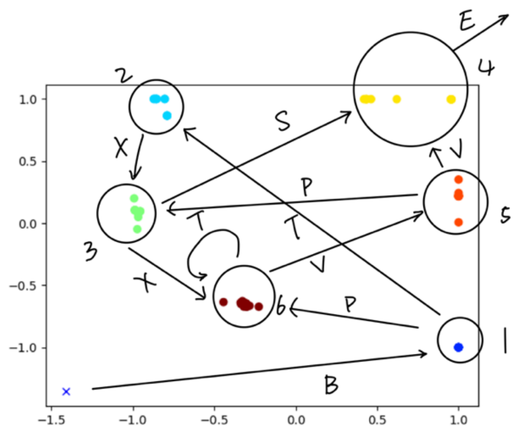
DenseNet

Part 2

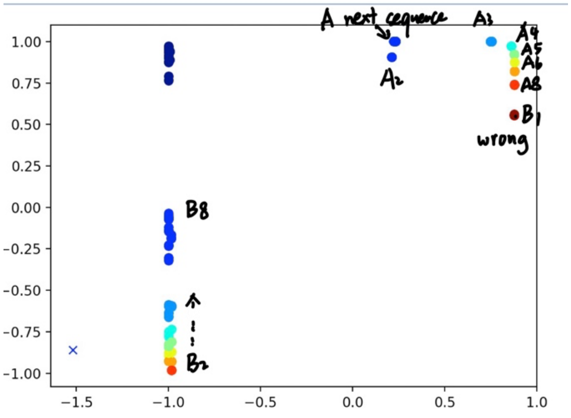


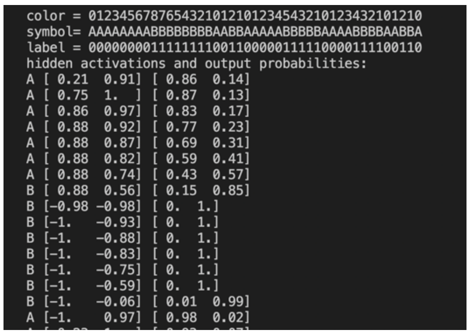
Part 3

(1)



(2)





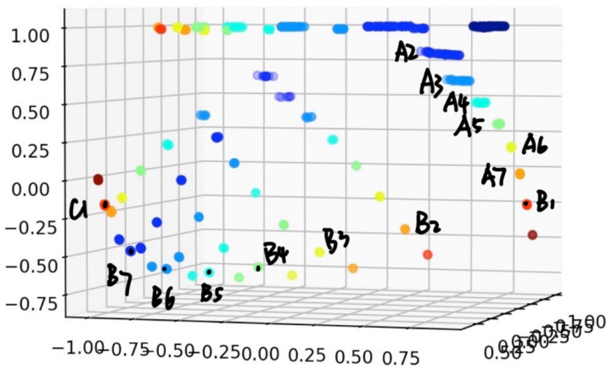
```
color = 0123456787654321012101234543210123432101210
symbol= AAAAAAAABBBBBBBBAABBAAAAABBBBBAAAABBBBAABBA
label = 0000000011111111001100000111110000111100110
hidden activations and output probabilities:
A [ 0.21  0.91] [ 0.86  0.14]
A [ 0.75  1.  ] [ 0.87  0.13]
A [ 0.86  0.97] [ 0.83  0.17]
A [ 0.88  0.92] [ 0.77  0.23]
A [ 0.88  0.87] [ 0.69  0.31]
A [ 0.88  0.82] [ 0.59  0.41]
A [ 0.88  0.74] [ 0.43  0.57]
B [ 0.88  0.56] [ 0.15  0.85]
B [-0.98 -0.98] [ 0.  1.]
B [-1.   -0.93] [ 0.  1.]
B [-1.   -0.88] [ 0.  1.]
B [-1.   -0.83] [ 0.  1.]
B [-1.   -0.75] [ 0.  1.]
B [-1.   -0.59] [ 0.  1.]
B [-1.   -0.06] [ 0.01  0.99]
A [-1.    0.97] [ 0.98  0.02]
```

(3)

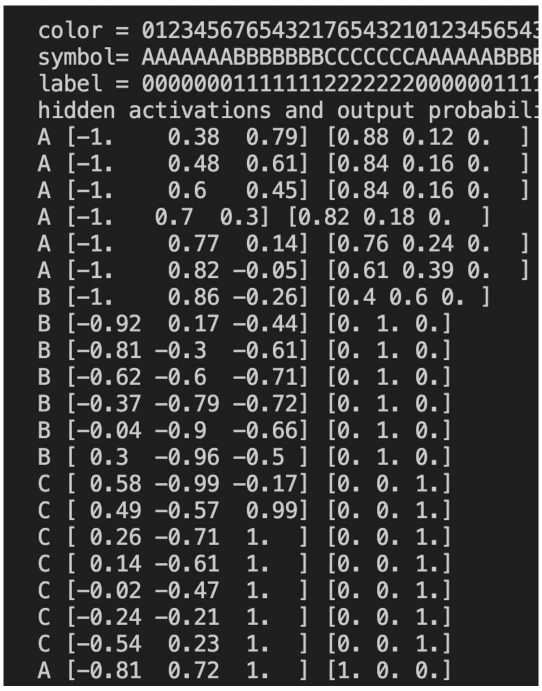
The sequence of symbols is labelled in the jet colormap plot.

It could be seen in the plot that in AnBn task, first B is unpredictable, the followed B and the initial A in the next sequence is predictable. Firstly, it starts with cross with initial weights, numbers of A are counted from 1 to 8 from dark blue point downwards to red. The probability of next character to be A is decreasing. When first B occurred, probability of A is larger than B so that we cannot predict first B occurred and the first B activations is plotted in A region with dark red. Then the remaining B is repelled to the other side with negative activations. The expected number of B is 8-1 = 7 and all of them should have probability p(B) = 1. Then after counting down expected number of Bs with color getting back to dark blue color (1) again, next symbol should be A which is predictable and plotted with color 0.

(4)



(5)

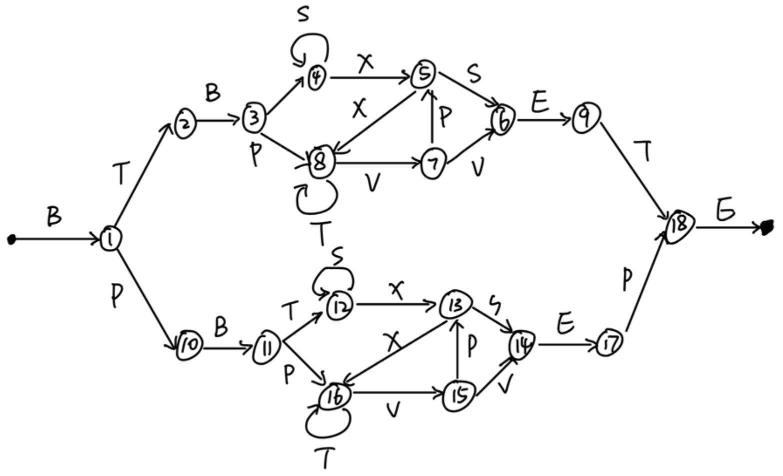


```
color = 0123456765432176543210123456543
symbol= AAAAAAABBBBBBBCCCCCCCAAAAAABBBB
label = 0000000111111122222220000001111
hidden activations and output probabil
A [-1.    0.38  0.79] [0.88 0.12 0.  ]
A [-1.    0.48  0.61] [0.84 0.16 0.  ]
A [-1.    0.6   0.45] [0.84 0.16 0.  ]
A [-1.   0.7  0.3] [0.82 0.18 0.  ]
A [-1.    0.77  0.14] [0.76 0.24 0.  ]
A [-1.    0.82 -0.05] [0.61 0.39 0.  ]
B [-1.    0.86 -0.26] [0.4 0.6 0. ]
B [-0.92  0.17 -0.44] [0. 1. 0.]
B [-0.81 -0.3  -0.61] [0. 1. 0.]
B [-0.62 -0.6  -0.71] [0. 1. 0.]
B [-0.37 -0.79 -0.72] [0. 1. 0.]
B [-0.04 -0.9  -0.66] [0. 1. 0.]
B [ 0.3  -0.96 -0.5 ] [0. 1. 0.]
C [ 0.58 -0.99 -0.17] [0. 0. 1.]
C [ 0.49 -0.57  0.99] [0. 0. 1.]
C [ 0.26 -0.71  1.  ] [0. 0. 1.]
C [ 0.14 -0.61  1.  ] [0. 0. 1.]
C [-0.02 -0.47  1.  ] [0. 0. 1.]
C [-0.24 -0.21  1.  ] [0. 0. 1.]
C [-0.54  0.23  1.  ] [0. 0. 1.]
A [-0.81  0.72  1.  ] [1. 0. 0.]
```

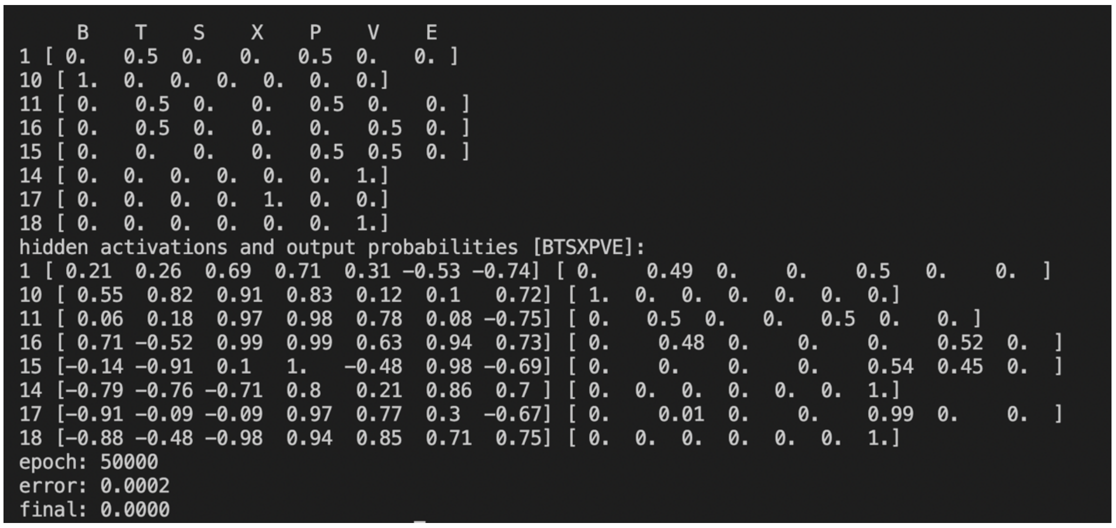
The probability of current symbol to be A is decreasing, and p(B) is increasing when the count of A is increasing. For AnBnCn task, first B is unpredictable, but the following B, following C and the first A in the next sequence is predictable.
Similar as AnBn task, the network counts the number of A and move to B region gradually with color plotted from dark blue to orange. When the first B unpredictable occurred, count down the expected number B and move towards C region with color back to blue again gradually. At the same time, probability of B is all 1 since all Bs are predictable. After that the network should

predict that the next symbol is C and color changes from blue to red immediately. Finally, if the count of C reaches 0, the network would predict next to be A and start again until finish. So that the probability in the last line of A in the next sequence is 1 which is correct.

(6)



For embedded reber grammar, it must remember which transition (T or P) was chosen after B, and which node between 2 and 10 occurred, then retain this information while it is processing the transitions in one of the identical RG boxes, to correctly predict the symbol T or P occurring before the final E and remember whether it is on edge 9 or 18.
The task is accomplished because LSTM module has a cell state to help information flow through the units without being altered by few linear interactions. One of the context units is assigned the task of retaining the knowledge of the initial T or P. Besides, each unit has an input, output and a forget gate which can add or remove the information to the cell state. The forget gate forgets information from previous cell which ratio is close to zero.
In this way, the context units can determine which units could change their values frequently while others preserve their state, until circumstances cause the gates to change the values.
Here is one of my training results with number of hidden nodes = 7. After reaching 1, it goes to node 10 by P, and from the last line the probability for 17-18 edge to be P is 0.99 which means that it retains info correctly. Thus, the training is successful.



```
     B    T    S    X    P    V    E
1 [ 0.   0.5  0.   0.   0.5  0.   0. ]
10 [ 1.  0.  0.  0.  0.  0.  0.]
11 [ 0.   0.5  0.   0.   0.5  0.   0. ]
16 [ 0.   0.5  0.   0.   0.   0.5  0. ]
15 [ 0.   0.   0.   0.   0.5  0.5  0. ]
14 [ 0.  0.  0.  0.  0.  0.  1.]
17 [ 0.  0.  0.  0.  1.  0.  0.]
18 [ 0.  0.  0.  0.  0.  0.  1.]
hidden activations and output probabilities [BTSXPVE]:
1 [ 0.21  0.26  0.69  0.71  0.31 -0.53 -0.74] [ 0.    0.49  0.    0.    0.5   0.    0.  ]
10 [ 0.55  0.82  0.91  0.83  0.12  0.1   0.72] [ 1.  0.  0.  0.  0.  0.  0.]
11 [ 0.06  0.18  0.97  0.98  0.78  0.08 -0.75] [ 0.   0.5  0.   0.   0.5  0.   0. ]
16 [ 0.71 -0.52  0.99  0.99  0.63  0.94  0.73] [ 0.    0.48  0.    0.    0.    0.52  0.  ]
15 [-0.14 -0.91  0.1   1.   -0.48  0.98 -0.69] [ 0.    0.    0.    0.    0.54  0.45  0.  ]
14 [-0.79 -0.76 -0.71  0.8   0.21  0.86  0.7 ] [ 0.  0.  0.  0.  0.  0.  1.]
17 [-0.91 -0.09 -0.09  0.97  0.77  0.3  -0.67] [ 0.    0.01  0.    0.    0.99  0.    0.  ]
18 [-0.88 -0.48 -0.98  0.94  0.85  0.71  0.75] [ 0.  0.  0.  0.  0.  0.  1.]
epoch: 50000
error: 0.0002
final: 0.0000
```