COMP9444 Neural Networks and Deep Learning

Assignment 1

Term 2, 2024

Submitted by

zID: z5365051

Name: Weiyuan (Felix) Li

Part 1: Japanese Character Recognition

1. **Answer question 1**

```
[[767.   5.   8.  14.  30.  63.   2.  62.  32.  17.]
[  6. 666. 110.  18.  27.  22.  60.  14.  26.  51.]
[  8.  58. 692.  27.  28.  21.  46.  36.  48.  36.]
[  5.  37.  59. 756.  16.  55.  14.  17.  30.  11.]
[ 61.  52.  80.  21. 621.  20.  31.  38.  20.  56.]
[  8.  29. 122.  16.  18. 727.  27.   8.  33.  12.]
[  5.  23. 146.  10.  25.  24. 724.  21.   9.  13.]
[ 16.  30.  25.  11.  85.  16.  55. 621.  91.  50.]
[ 10.  42.  94.  40.   8.  30.  45.   6. 701.  24.]
[  8.  52.  90.   3.  55.  31.  19.  30.  40. 672.]]
```

Test set: Average loss: 1.0085, **Accuracy: 6947/10000 (69%)**

2. **Answer question 2**

```
[[848.   5.   1.   5.  32.  26.   4.  41.  32.   6.]
 [  6. 822.  31.   2.  18.   7.  63.   6.  17.  28.]
 [  8.  12. 836.  39.  13.  17.  26.  10.  24.  15.]
 [  3.   7.  33. 918.   2.  14.   7.   2.   5.   9.]
 [ 36.  27.  20.   3. 823.   5.  32.  15.  22.  17.]
 [  7.   7.  81.  11.  16. 823.  32.   1.  15.   7.]
 [  3.   7.  49.   9.  13.   7. 894.   8.   1.   9.]
 [ 19.  19.  17.   2.  19.  12.  27. 818.  25.  42.]
 [ 10.  23.  26.  52.   4.   8.  29.   4. 837.   7.]
 [  1.  14.  56.   4.  32.   6.  20.  12.  14. 841.]]
```

Hidden layer node count: 128
Test set: Average loss: 0.5133, **Accuracy: 8460/10000 (85%)**

3. **Answer question 3**

```
[[952.   3.   4.   0.  21.   2.   2.  12.   1.   3.]
 [  3. 921.   9.   1.  17.   0.  32.   4.   4.   9.]
 [ 10.  22. 863.  42.   8.   9.  21.  13.   4.   8.]
 [  2.   3.  27. 943.   6.   6.   3.   4.   2.   4.]
 [ 20.  15.   4.   5. 909.   2.  17.  15.   6.   7.]
 [  4.  19.  37.   4.   2. 906.  18.   6.   3.   1.]
 [  2.  13.  16.   2.   6.   2. 955.   3.   0.   1.]
 [  7.  11.   1.   1.   7.   0.   9. 943.   3.  18.]
 [  5.  24.   8.   5.  13.   5.   6.   4. 930.   0.]
 [  9.  13.  12.   2.   4.   1.   1.   4.   2. 952.]]
```

Convolution layer: (1, 16, (5,5)), Pool; (16, 256, (5,5)), Pool; Hidden node: 4096
Test set: Average loss: 0.2628, **Accuracy: 9274/10000 (93%)**

4. Answer question 4

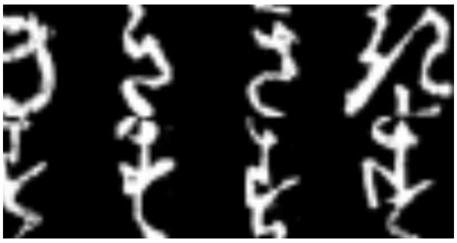
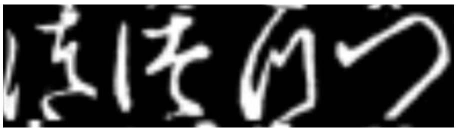
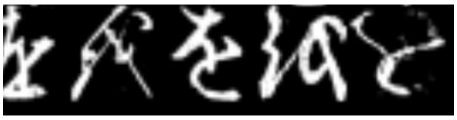
a. As expected, the accuracy of a single linear network is the worst, comparing to a two-layer linear network which improves to 85%. Finally with convolution the third network achieves the most accuracy at 93%.

b. With the implementation in kuzu.py, we have
   - i. Linear: 28 * 28 + 10 = 794
   - ii. 2-Layer FC: (28 * 28 * 128 + 128 * 10) + 128 + 10 = 101770
   - iii. 2-Layer Conv + Linear:
     (5 * 5 * 1 + 1) * 16 + (5 * 5 * 16 + 1) * 256 + (256 * 4 * 4 + 1) * 10
     = 144042

c. With each model we have the most likely confuses
   - i. Linear
     1. 0 = 5
     2. 1 = 2
     3. 2 = 1
     4. 3 = 2
     5. 4 = 0
     6. 5 = 2
     7. 6 = 2
     8. 7 = 4
     9. 8 = 2
     10. 9 = 4
   - ii. 2-Layer FC
     1. 0 = 7
     2. 1 = 6
     3. 2 = 3
     4. 3 = 2
     5. 4 = 0
     6. 5 = 2
     7. 6 = 2
     8. 7 = 6
     9. 8 = 3
     10. 9 = 2
   - iii. Convolution
     1. 0 = 3
     2. 1 = 5
     3. 2 = 3
     4. 3 = 2
     5. 4 = 0
     6. 5 = 2
     7. 6 = 2
     8. 7 = 9

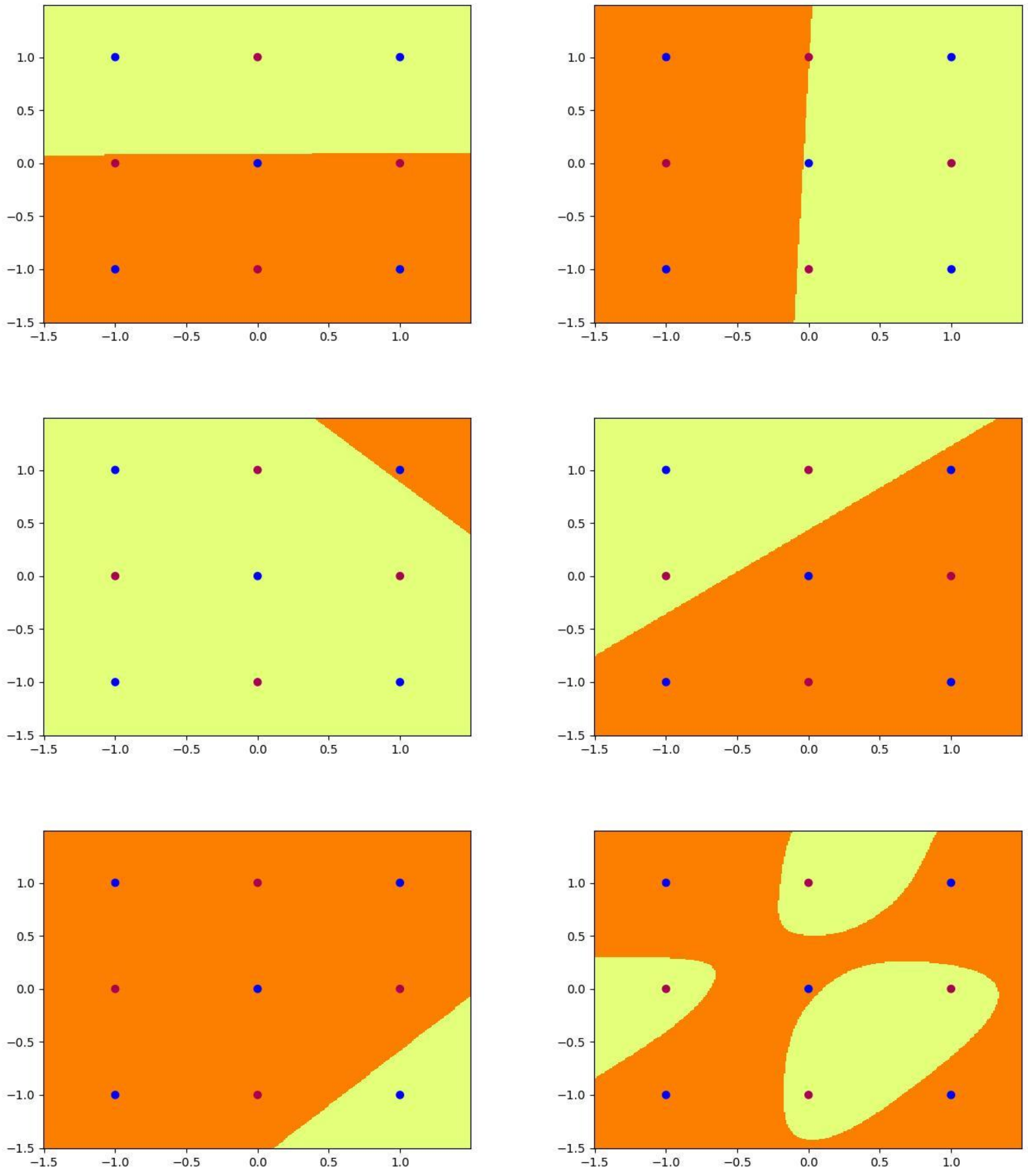
9. 8 = 2
10. 9 = 2

iv. The confusion mainly comes from the malformed handwriting which makes them similar to that of another digit. For example, a consistent confusion in the above three models are 2 (su) and 3 (tsu):

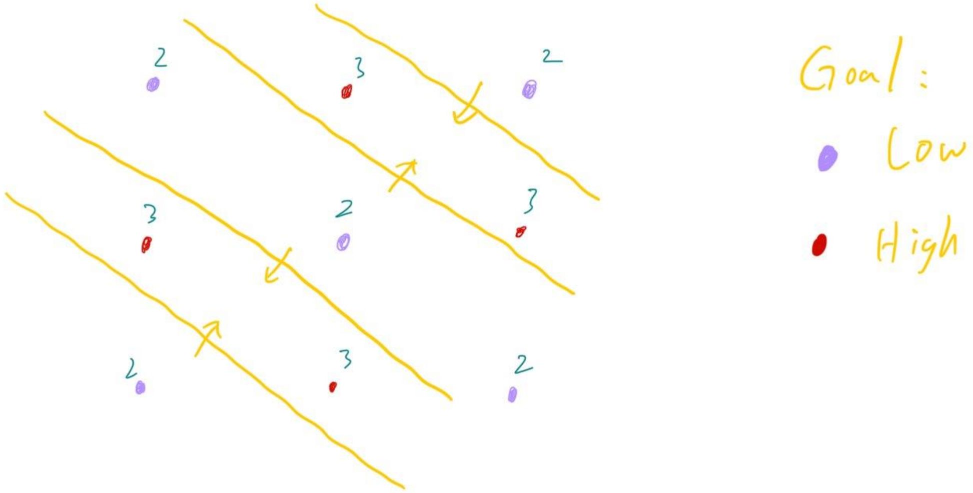
1. 
    a. The second finals become indistinguishable
2. Another consistent one is mistaking 4 (na) to 0 (o)
3.  (na)
4.  (o)
5. For both of above cases, the indistinguishability comes in to the ancient font of kuzushiji instead of modern hiragana. The network hence not only has to distinguish between difference in MNISTs, but also the kuzushiji and hiragana. A viable way to do this of course is to increase the convolution layer to extract more features and adding more fully connected layers after to merge them.

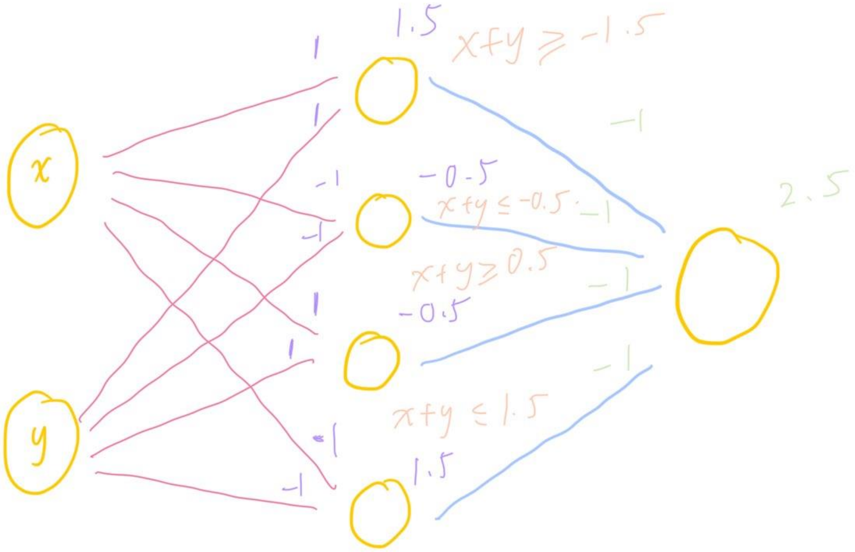
Part 2: Multi-Layer Perceptron

1. **Answer question 1**


1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0

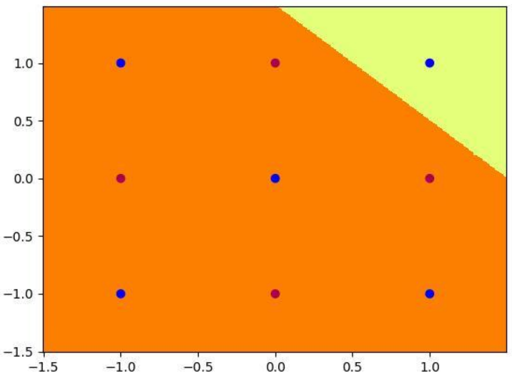
2. Answer question 2

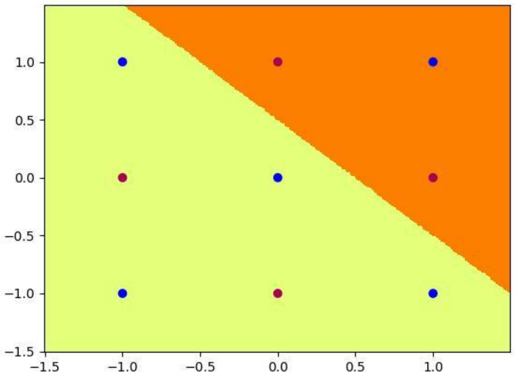


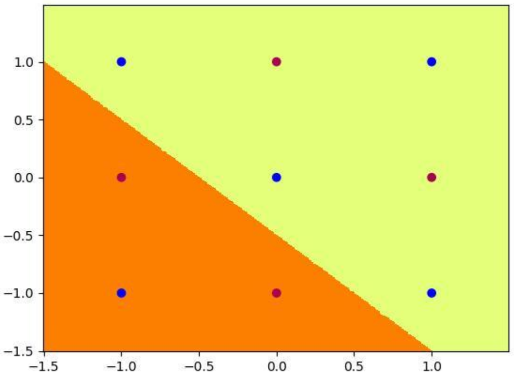


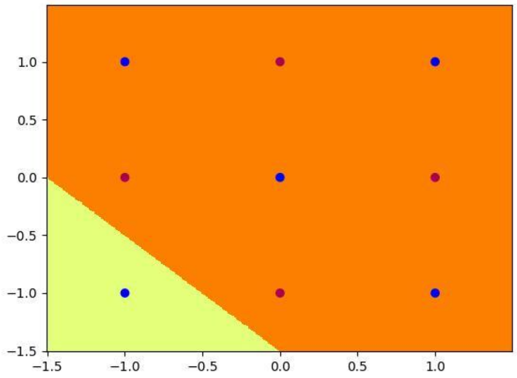
Data (position)	Hidden unit activation	Output activation
[-1 -1] =>	[False True False True]	True
[-1 1] =>	[True False False True]	True
[0 0] =>	[True False False True]	True
[1 -1] =>	[True False False True]	True
[1 1] =>	[True False True False]	True
[-1 0] =>	[True True False True]	False
[0 -1] =>	[True True False True]	False
[0 1] =>	[True False True True]	False
[1 0] =>	[True False True True]	False

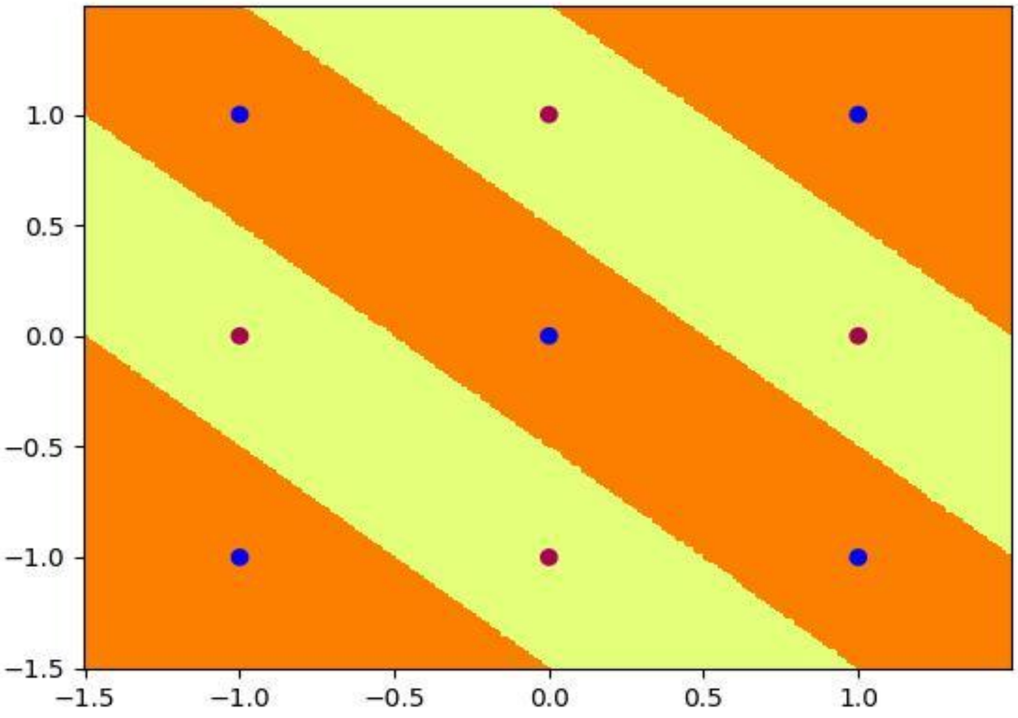
3. **Answer question 3**





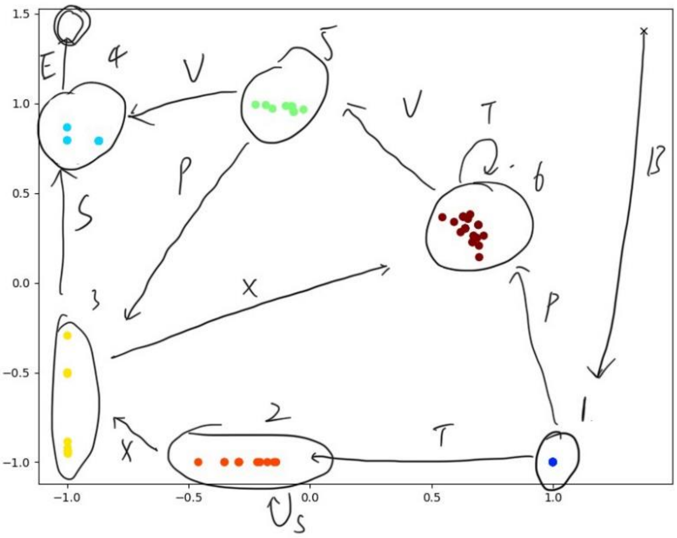




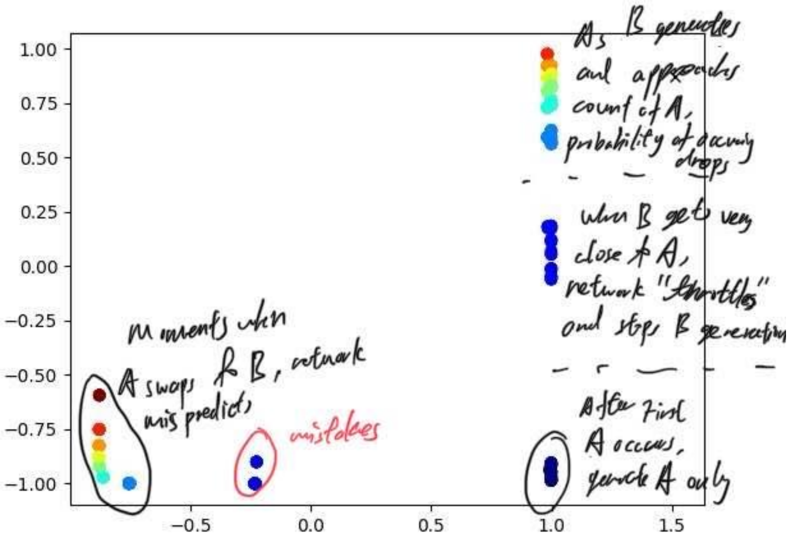


Part 3: Hidden Unit Dynamics for Recurrent Networks

1. Answer question 1



2. Answer question 2



As B generates and approaches count of A, probability of accuracy drops

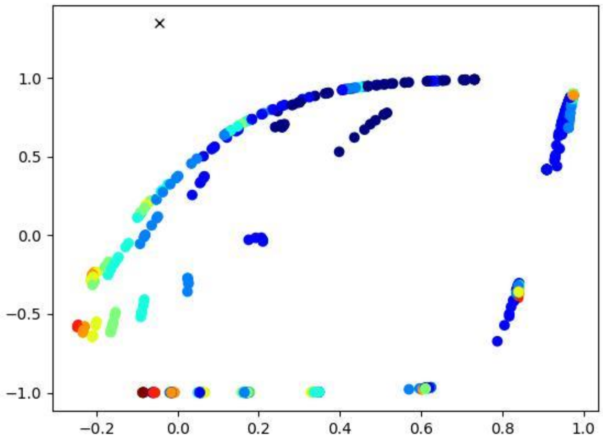
when B gets very close to A, network "throttles" and stops B generation

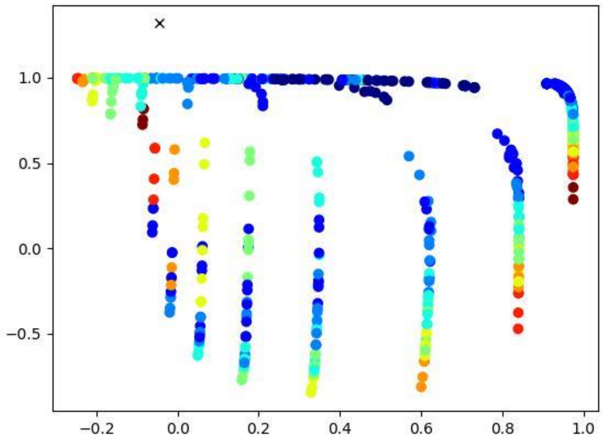
After first A occurs, generate A only

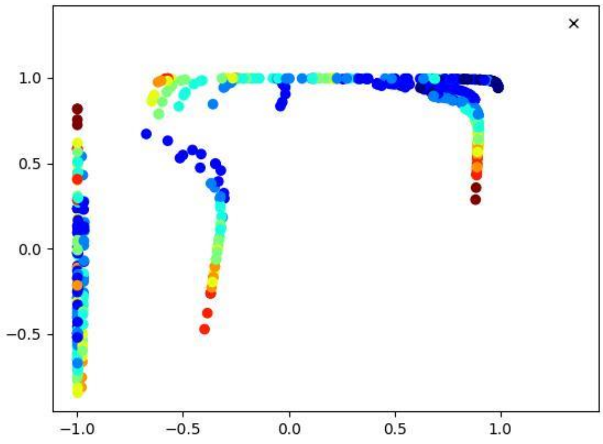
moments when A swaps to B, network mispredicts

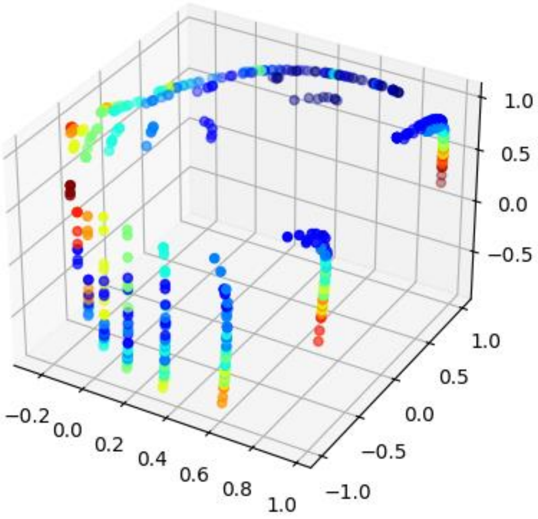
mistakes

3. Answer question 3
   We have top, front, right and finally 3D view.


×
1.0
0.5
0.0
−0.5
−1.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


×
1.0
0.5
0.0
−0.5
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


×
1.0
0.5
0.0
−0.5
−1.0
−0.5
0.0
0.5
1.0



4. Answer question 4