Perceptron Report

By Josias Moukpe

1. Training Output

Here is the output of the Network while training

Dataset size 21

Epoch # 0

error
[[1], [-1], [-1], [-1], [-1], [-1], [-1]]
error
[[-2], [2], [0], [0], [0], [0], [0]]
error
[[0], [-2], [2], [0], [0], [0], [0]]
error
[[0], [-2], [-2], [2], [0], [0], [0]]
error
[[0], [0], [0], [-2], [2], [0], [0]]
error
[[0], [0], [0], [0], [-2], [2], [0]]
error
[[0], [0], [0], [0], [-2], [0], [2]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [2], [-2], [-2], [-2], [-2], [0]]
error
[[0], [0], [2], [0], [0], [-2], [0]]
error
[[0], [0], [0], [2], [0], [0], [0]]
error
[[0], [-2], [0], [0], [2], [0], [-2]]
error
[[-2], [0], [-2], [-2], [0], [0], [0]]
error

[[-2], [0], [-2], [0], [0], [0], [2]]
error
[[2], [0], [0], [0], [0], [0], [-2]]
error
[[0], [2], [0], [0], [0], [0], [-1]]
error
[[0], [0], [0], [-2], [0], [0], [0]]
error
[[0], [-2], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [-2], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]

Epoch # 1

error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [-2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error

[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [-2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [-2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]

Epoch # 2

error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error

[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [-2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]

Epoch # 3

error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [-2], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error

[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]

Epoch # 4

error
[[0], [0], [0], [0], [0], [0], [0]]
error

[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]
error
[[0], [0], [0], [0], [0], [0], [0]]

2. Trained Weights and Biases

```
{
    "Weights": [
    [-5, -1, 1, 5, -1, -9, 3, -1, -1, 3, 9, -1, -1, -1, -1, -1, 7, 1, 7, -1, -1, -1, -1, 5, 3, 9, -1, -1, -1, -1, 1, -1, 1, -1, 3, -1,
5, 5, 9, 9, 5, -1, 3, -3, 3, -1, 3, 1, 3, 3, -3, 3, 3, -1, 1, 3, 1, 5, -3, -5, -3, 1, 9],
    [3, 3, -3, -3, -1, 3, -9, 3, -1, 3, 1, 3, -1, 7, -1, 3, 3, 1, -1, 3, 11, -1, 3, 1, 3, 1, 7, 7, -1, 3, 5, 7, 1, 11, 3, -1, 1, 1,
1, 1, 1, 11, -1, 1, 3, 3, 3, 1, 11, 3, -3, 3, 3, 3, -3, 7, 1, 1, -3, -1, -3, 1, -7],
    [-7, -3, 3, 3, 5, -3, 9, -3, 9, 5, 3, 1, 1, 5, 5, 1, 5, -1, 5, 1, 1, 5, 1, -1, 5, 3, 1, -3, 5, -7, -1, 1, -1, -3, 1, 5, -1, -1,
3, 3, -1, -3, 5, -5, 5, 1, 5, -1, 1, -3, 3, 5, 5, 1, -1, 5, -9, -5, -1, 1, -1, -1, 3],
    [5, 5, -1, -1, 1, -7, -3, 5, 1, 5, 3, 5, 5, -7, 1, 5, 5, 3, 5, 1, 5, 1, 5, 3, 1, 3, 1, 9, 1, 1, -5, -3, -1, -3, 13, 1, 3, 3, -1,
3, -1, 9, 1, -1, 5, 5, 5, -1, 5, 5, -5, 5, 5, 5, 3, -3, 3, 3, -5, -3, -5, -5, -1],
    [3, 7, 1, 5, 7, -5, 11, 3, 3, 7, 5, -1, -1, 3, -1, 7, 7, -3, 7, 3, -5, -1, 7, -3, 11, 5, -1, -1, -1, 3, 5, 15, 9, -5, 3, -1, 5,
1, 5, 5, 1, -5, -1, 1, 7, 3, -1, 1, -5, 3, -3, 7, 7, 3, -7, 3, 1, 5, -3, 7, 5, -3, 17],
    [-3, -3, -9, -1, 1, 1, 5, 1, -3, 5, 3, 5, 1, 1, -3, 1, 5, 3, 1, 9, -3, -3, 1, 3, 1, 3, 9, 1, -3, -3, -5, 1, -1, 5, 5, -3, -1, 3,
-1, 3, 7, 1, -3, 3, 5, 5, 1, 7, -3, 1, -1, 5, 5, 5, -1, 1, -5, -5, -5, 1, -1, -5, -1],
    [4, 0, -2, -10, -4, 4, 2, 2, 4, 2, -4, 10, 2, 0, 2, 4, -2, 8, -2, 2, 0, 2, 4, 2, 0, -6, 0, 2, 2, 0, 0, -2, -4, -2, 0, 2, -2, 0,
0, -4, -4, 0, -2, 2, 2, 6, 6, 0, -4, -2, 2, 2, 2, 6, 4, -4, -2, -6, -2, -4, -6, -2, -4]
    ],
  "Biases": [
    [-3],[-3],[-5],[-5],[-7],[-5],[-2]
    ]
}
```

3. Example of Correct Prediction

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☐	☐	☑	☑	☑	☐	☑
1	☐	☑	☐	☐	☐	☑	☑
2	☑	☐	☐	☐	☐	☐	☑
3	☑	☐	☐	☐	☐	☐	☐
4	☑	☐	☐	☐	☐	☐	☐
5	☑	☐	☐	☐	☐	☐	☐
6	☑	☐	☐	☐	☐	☐	☑
7	☐	☑	☐	☐	☐	☑	☐
8	☐	☐	☑	☑	☑	☐	☐

Letter

C

Here are the results from Question 2 when entering wrong data for every letter

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☐	☐	☑	☑	☐	☐	☐
1	☐	☐	☑	☐	☑	☐	☐
2	☐	☑	☐	☐	☐	☑	☐
3	☑	☐	☐	☐	☐	☑	☐
4	☑	☐	☑	☑	☑	☑	☐
5	☑	☑	☑	☑	☑	☑	☑
6	☑	☐	☐	☐	☐	☐	☑
7	☑	☑	☐	☐	☐	☐	☑
8	☑	☐	☐	☐	☐	☐	☐

output:
[[1], [1], [-1], [-1], [-1], [-1], [-1]] -> A

Testing

Input Matrix

	0	1	2	3	4	5	6
0	✓	✓	✓	✓	✓	✓	✓
1	✓	☐	☐	☐	☐	✓	✓
2	✓	☐	☐	☐	✓	✓	✓
3	✓	☐	☐	☐	✓	☐	✓
4	✓	✓	✓	✓	✓	✓	✓
5	✓	☐	☐	☐	✓	✓	☐
6	✓	☐	☐	✓	✓	✓	✓
7	✓	☐	☐	☐	☐	✓	✓
8	✓	✓	✓	✓	✓	✓	☐

output:
[[-1], [1], [-1], [-1], [-1], [-1], [-1]] -> B

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☐	☑	☑	☑	☑	☑	☑
1	☑	☐	☐	☑	☐	☑	☑
2	☑	☐	☐	☐	☐	☑	☐
3	☑	☐	☐	☐	☐	☐	☐
4	☑	☐	☐	☐	☐	☐	☐
5	☑	☐	☐	☐	☐	☐	☐
6	☑	☐	☐	☑	☐	☑	☑
7	☑	☐	☐	☑	☑	☐	☑
8	☐	☑	☑	☑	☑	☑	☐

output:
[[-1], [-1], [1], [-1], [-1], [-1], [-1]] -> C

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☑	☑	☑	☑	☑	☐	☐
1	☑	☐	☐	☐	☐	☑	☐
2	☑	☐	☐	☐	☐	☑	☑
3	☑	☐	☐	☐	☐	☑	☑
4	☐	☐	☐	☐	☐	☑	☑
5	☑	☐	☐	☐	☐	☑	☑
6	☑	☐	☐	☐	☐	☑	☑
7	☑	☐	☐	☐	☐	☑	☑
8	☑	☑	☑	☑	☑	☐	☐

output:
[[-1], [1], [-1], [1], [-1], [-1], [-1]] -> B or D

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☑	☐	☑	☑	☑	☑	☑
1	☑	☑	☑	☑	☑	☐	☑
2	☑	☐	☐	☐	☐	☐	☐
3	☑	☐	☐	☐	☐	☐	☐
4	☑	☑	☑	☑	☑	☑	☐
5	☑	☑	☐	☐	☐	☐	☐
6	☑	☑	☐	☐	☐	☐	☐
7	☑	☐	☐	☐	☑	☐	☐
8	☑	☑	☑	☑	☑	☑	☑

output:
[[-1], [-1], [-1], [-1], [1], [-1], [-1]] -> E

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☐	☐	☑	☑	☑	☑	☑
1	☐	☐	☐	☐	☐	☑	☑
2	☐	☐	☐	☐	☐	☑	☐
3	☐	☐	☐	☐	☐	☑	☐
4	☑	☑	☑	☐	☐	☑	☐
5	☐	☑	☐	☐	☐	☑	☐
6	☐	☑	☐	☐	☐	☑	☐
7	☐	☑	☑	☐	☐	☑	☐
8	☐	☐	☐	☑	☑	☐	☐

output:
[[-1], [-1], [-1], [-1], [-1], [1], [-1]] -> J

Testing

Input Matrix

	0	1	2	3	4	5	6
0	☑	☐	☐	☐	☑	☑	☐
1	☑	☐	☐	☑	☑	☐	☐
2	☑	☐	☑	☑	☐	☐	☐
3	☑	☑	☑	☑	☐	☐	☐
4	☑	☑	☐	☑	☐	☐	☐
5	☑	☐	☐	☐	☑	☐	☐
6	☑	☐	☐	☐	☑	☐	☐
7	☑	☐	☐	☐	☐	☑	☐
8	☑	☐	☐	☐	☐	☐	☑

output:
[[-1], [-1], [-1], [-1], [-1], [-1], [1]] -> K

4. Missing Data

Now here is the output when missing data is manually introduced into the input matrix

A with 5 random missing pixels
output:
[[1], [-1], [-1], [-1], [-1], [-1], [-1]]   ->   A

B with 5 random missing pixels
output:
[[-1], [1], [-1], [-1], [-1], [-1], [-1]]   ->   B

C with 5 random missing pixels
output:
[[-1], [-1], [1], [-1], [-1], [-1], [-1]]   ->   C

D with 5 random missing pixels
output:
[[-1], [-1], [-1], [1], [-1], [-1], [-1]]   ->   D

E with 5 random missing pixels
output:
[[-1], [-1], [-1], [-1], [1], [-1], [-1]]   ->   E

J with 5 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [1], [-1]]   ->   J

K with 5 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [-1], [1]]   ->   K

A with 10 random missing pixels
output:
[[1], [-1], [-1], [-1], [-1], [-1], [-1]]   ->   A

B with 10 random missing pixels
output:
[[-1], [1], [-1], [-1], [-1], [-1], [-1]]   ->   B

C with 10 random missing pixels
output:
[[-1], [-1], [1], [-1], [-1], [-1], [-1]]   ->   C

D with 10 random missing pixels
output:
[[-1], [-1], [-1], [1], [-1], [-1], [-1]]   ->   D

E with 10 random missing pixels
output:
[[-1], [-1], [-1], [-1], [1], [-1], [-1]]   ->   E

J with 10 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [1], [-1]]   ->   J

K with 10 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [-1], [1]]   ->   K

A with 15 random missing pixels
output:
[[1], [-1], [-1], [-1], [-1], [-1], [-1]]   ->   A

B with 15 random missing pixels
output:
[[-1], [1], [-1], [-1], [-1], [-1], [-1]]   ->   B

C with 15 random missing pixels
output:
[[-1], [-1], [1], [-1], [-1], [-1], [-1]]   ->   C

D with 15 random missing pixels
output:

[[-1], [-1], [-1], [1], [-1], [-1], [-1]]   ->   D

E15 with 15 random missing pixels
output:
[[-1], [-1], [-1], [-1], [1], [-1], [-1]]   ->   E

J with 15 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [1], [-1]]   ->   J

K with 15 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [-1], [1]]   ->   K

A with 20 random missing pixels
output:
[[1], [-1], [-1], [-1], [-1], [-1], [-1]]   ->   A

B with 20 random missing pixels
output:
[[-1], [1], [-1], [-1], [-1], [-1], [-1]]   ->   B

C with 20 random missing pixels
output:
[[-1], [-1], [1], [-1], [-1], [-1], [-1]]   ->   C

D with 20 random missing pixels
output:
[[-1], [-1], [-1], [1], [-1], [-1], [-1]]   ->   D

E with 20 random missing pixels
output:
[[-1], [-1], [-1], [-1], [1], [-1], [-1]]   ->   E

J with 20 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [1], [-1]]   ->   J

K with 20 random missing pixels
output:
[[-1], [-1], [-1], [-1], [-1], [-1], [1]]   ->   K

5. Ease of Identification

The hardest letters to identify are J, D, K in that order
The easiest letters to identify are C, A, B, E in that order