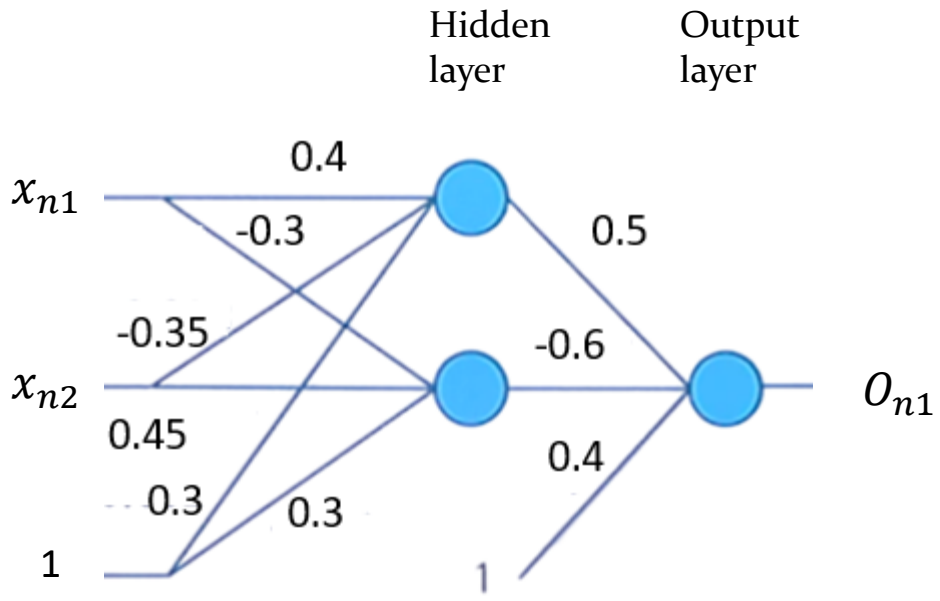
3. Neural Network Programming requirements

Requirement (1)

- XOR implementation using Error Back Propagation
  - Hidden nodes : 2
  - Learning rate : 0.02
  - Activation function : sigmoid

x_{n1}	x_{n2}	t_{n1}
1	1	0
1	0	1
0	1	1
0	0	0



Requirement (1)

- XOR outputs

[100000번 학습 결과]

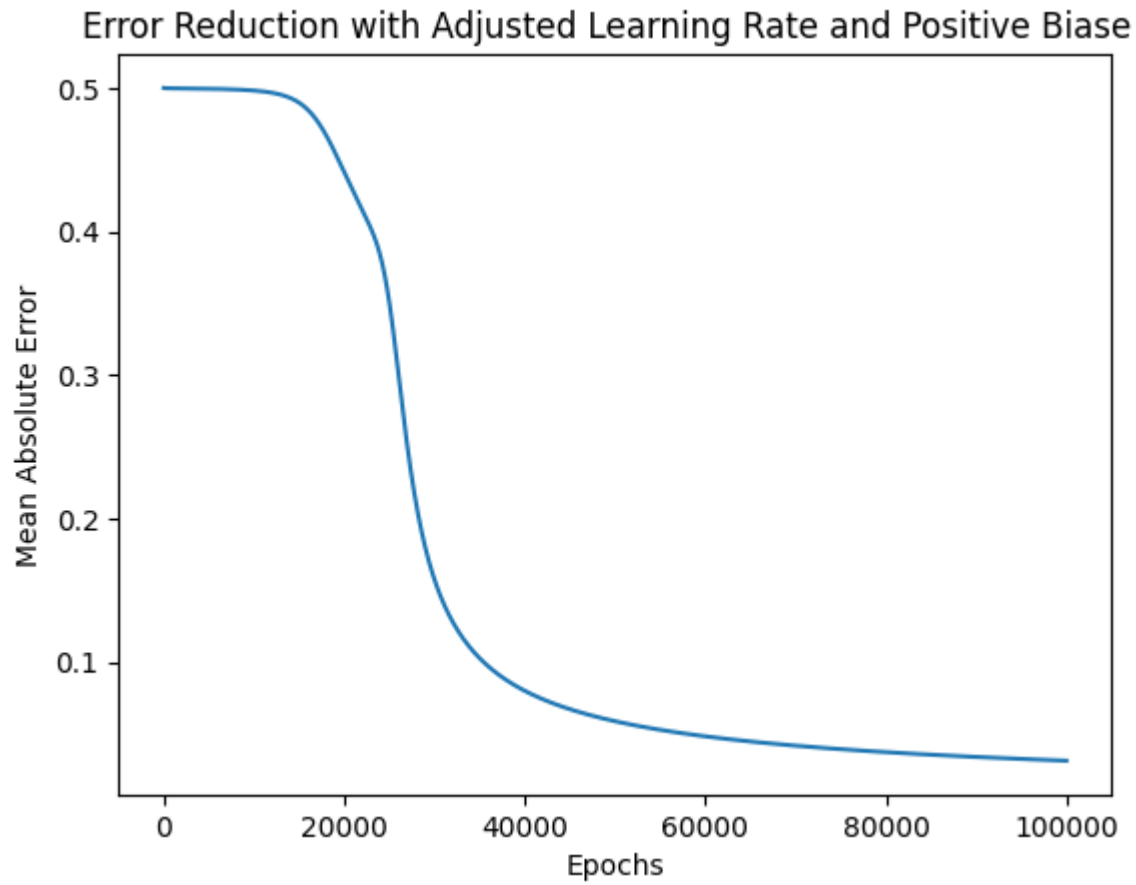
```
Final Outputs :
 [[0.02803873]
 [0.96470059]
 [0.96991792]
 [0.03143102]]

Final Weights (Input to Hidden):
 [[-5.24760033 -6.01692012]
 [ 5.00682797  6.04416126]
 [-2.70746378  3.10550889]]

Final Weights (Hidden to Output):
 [[ 8.56046799]
 [-8.02659253]
 [ 3.71905948]]
```

Requirement (1)

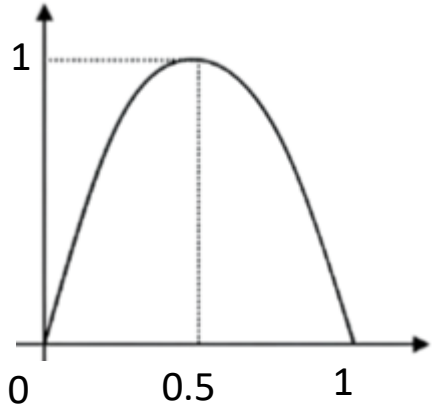
- XOR outputs using matplotlib



Requirement (2)

- Quadratic function implementation using Error back propagation
  - Hidden nodes : 4
  - Iteration(epoch) : 500,000
  - Learning rate : 0.7
  - Activation function : sigmoid
  - Initial weight : random.uniform()

$$f(x) = -4x^2+4x$$

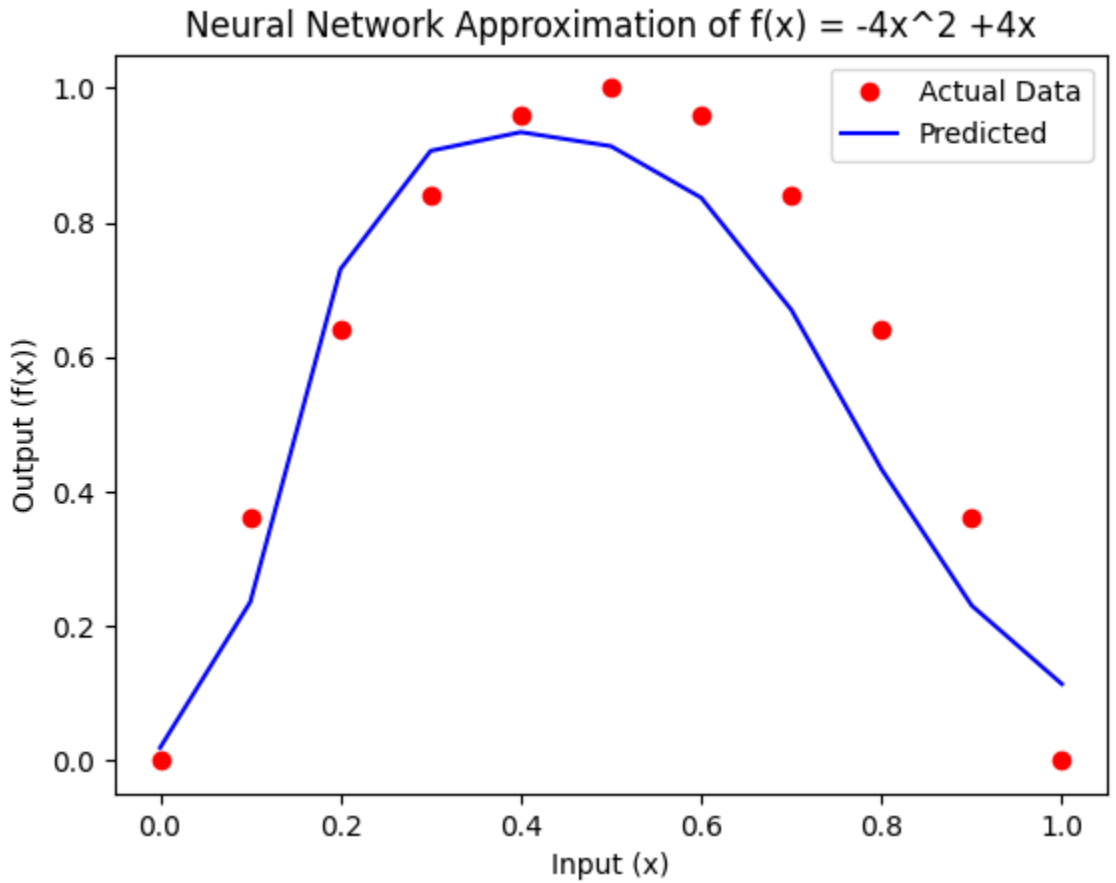


(training data)

input	output
0.00	0.00
0.10	0.36
0.20	0.64
0.30	0.84
0.40	0.96
0.50	1.00
0.60	0.96
0.70	0.84
0.80	0.64
0.90	0.36
1.00	0.00

Requirement (2)

- **Outputs**




```
Iteration 0, Loss: 0.215449448665259
Iteration 100000, Loss: 0.013803495708764884
Iteration 200000, Loss: 0.013925151227826211
Iteration 300000, Loss: 0.013940912238430452
Iteration 400000, Loss: 0.013943030059185152
```