Lab1: back-propagation

Student id / name: A113599 / 楊淨富

I. Introduction:

利用兩層 hidden layer 的 Neural network 來訓練 input 為二維的資料(range from 0.0 to 1.0) 以及其所對應到的 label 為一維的 ground truth (0 or 1)。因為資料集較小，故 training set 同時也是 testing set。

II. Experiment setups

A. Activation functions

- Sigmoid function:

```
def sigmoid(self, x, derivative=False):
    if derivative:
        return (np.exp(-x))/((np.exp(-x)+1)**2)
    return 1/(1 + np.exp(-x))
```

Sigmoid function 是一種 activation function，其輸出為 0 至 1。

derivative 若設為 True 則是 sigmoid 微分後的函數。反之則為 sigmoid function。

- Relu funtion:

```
def relu(self, x, derivative=False):
    if derivative:
        return np.where(x < 0, 0, 1)
    return np.maximum(0, x)
```

Relu 也是一種 activation function，假定輸入為 x，則輸出為 max(0, x)。

比較麻煩的是其在 0 這個點無法進行微分(因為左右兩點的斜率不同)。

B. Neural network

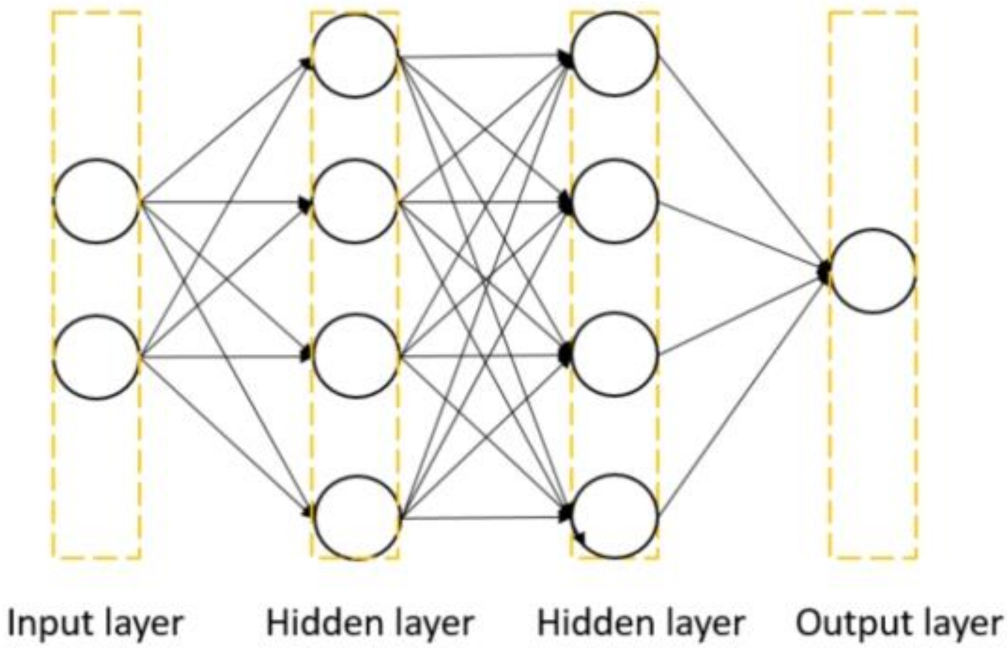


Figure 1. Two-layer neural network

- Layers:
  - Input layer: 由於 input 是二維，所以兩個 neuron。
  - Hidden layer: 要求至少兩層 hidden layers。我預設兩層(h1, h2)各 4 個 neuron。
  - Output layer: 由於 label 是 0 或 1，所以一個 neuron。

- Training process:
  1. 先隨機初始化 parameters (此 lab 即 weights)
  2. 透過 feed_forward 得出 output
  3. 進行 back_propagation(將 Loss funtion 對各 weights 做微分，算出各 weights 之 gradient)
  4. 透過 optimize 更新各 weights
  5. 重複進行 Step 2 - 4

C. Back propagation:

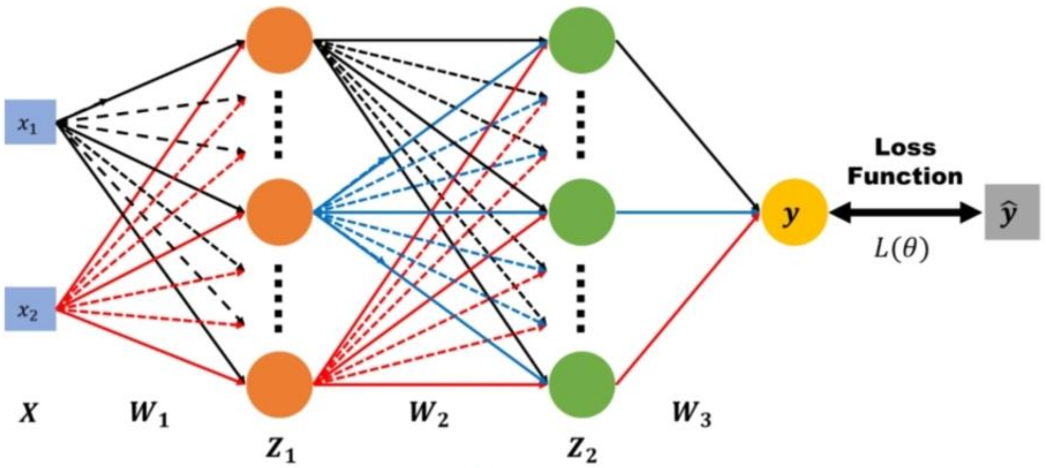


Figure 2. Forward pass

- In the figure 2, we use the following definitions for the notations:
  1. x_1, x_2 : *nerual network inputs*
  2. X : $[x_1, x_2]$
  3. y : *nerual network outputs*
  4. $\hat{y}$: *ground truth*
  5. $L(\theta)$: *loss function*
  6. W_1, W_2, W_3 : *weight matrix of network layers*

  $Z_1 = \sigma(XW_1)$ $\quad$ $Z_2 = \sigma(Z_1 W_2)$ $\quad$ $y = \sigma(Z_2 W_3)$

- 由 output 依序向前更新各 weights
- 更新方法為:

將對 Loss function 對各 weight 做偏微分，計算出各 weight 之 gradients 後，算出 new_weight = old_weight - learning_rate * weight_gradient。如下圖所示(依序更新 W3,W2,W1)

x	$\cdot$	$W1$	$H1$	$\xrightarrow{\sigma}$ $Z1$
$(100,2)$		$(2,4)$	$(100,4)$	$(100,4)$
$Z1$	$\cdot$	$W2$	$H2$	$\xrightarrow{\sigma}$ $Z2$
$(100,4)$		$(4,4)$	$(100,4)$	$(100,4)$
$Z2$	$\cdot$	$W3$	$H3$	$\xrightarrow{\sigma}$ $Z3$ (y) output $L = (y-\hat{y})^2$
$(100,4)$		$(4,1)$	$(100,1)$	$(100,1)$

$$\frac{dL}{dW3} = \frac{dL}{dZ_3} \times \frac{d\,\overset{\sigma(H_3)}{\cancel{Z_3}}}{dH_3} \times \frac{d\,\overset{Z_2\cdot W_3}{\cancel{H_3}}}{dW_3}$$

$$= \underline{2 \times (y-\hat{y}) \times (1-0)} \times \text{sig-derivate}(H_3) \times Z_2.$$

$$\frac{dL}{dW_2} = \boxed{\frac{dL}{dZ_3} \times \frac{dZ_3}{dH_3}} \times \frac{d\,\overset{Z_2\cdot W_3}{\cancel{H_3}}}{dZ_2} \times \frac{d\,\overset{\sigma(H_2)}{\cancel{Z_2}}}{dH_2} \times \frac{d\,\overset{Z_1\cdot W_2}{\cancel{H_2}}}{dW_2}.$$

$$= \underline{2 \times (y-\hat{y}) \times (1-0) \times \text{sig-derivate}(H_3)} \times W_3 \times \text{sig-derivate}(H_2) \times Z_1$$

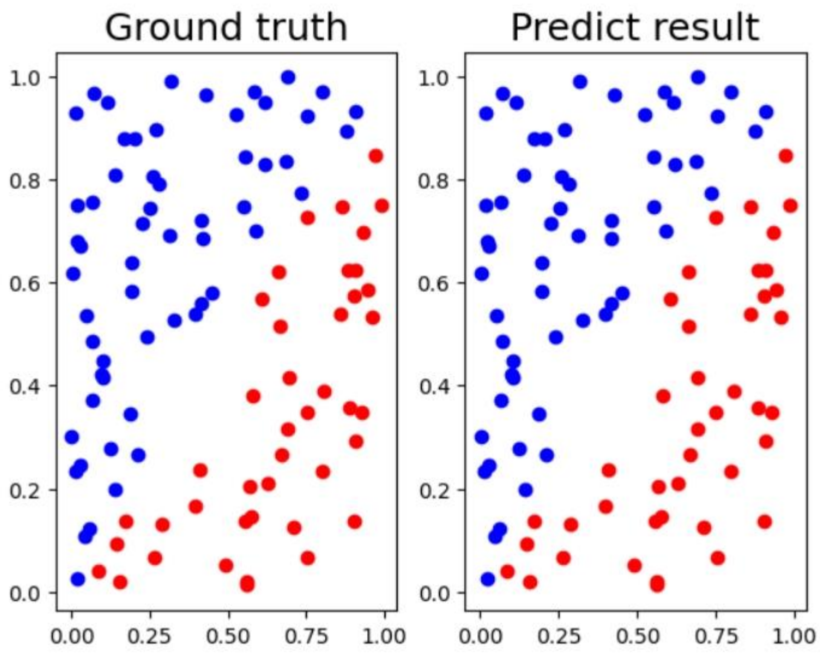
$$\frac{dL}{dW_1} = \boxed{\frac{dL}{dZ_3} \times \frac{dZ_3}{dH_3} \times \frac{dH_3}{dZ_2} \times \frac{dZ_2}{dH_2}} \times \frac{d\,\overset{Z_1\cdot W_2}{\cancel{H_2}}}{dZ_1} \times \frac{dZ_1}{dH_1} \times \frac{d\,\overset{x\cdot W_1}{\cancel{H_1}}}{dW_1}$$

$$= \underline{2 \times (y-\hat{y}) \times s\_d(H_3) \times W_3 \times s\_d(H_2)} \times W_2 \times s\_d(H_1) \times x$$

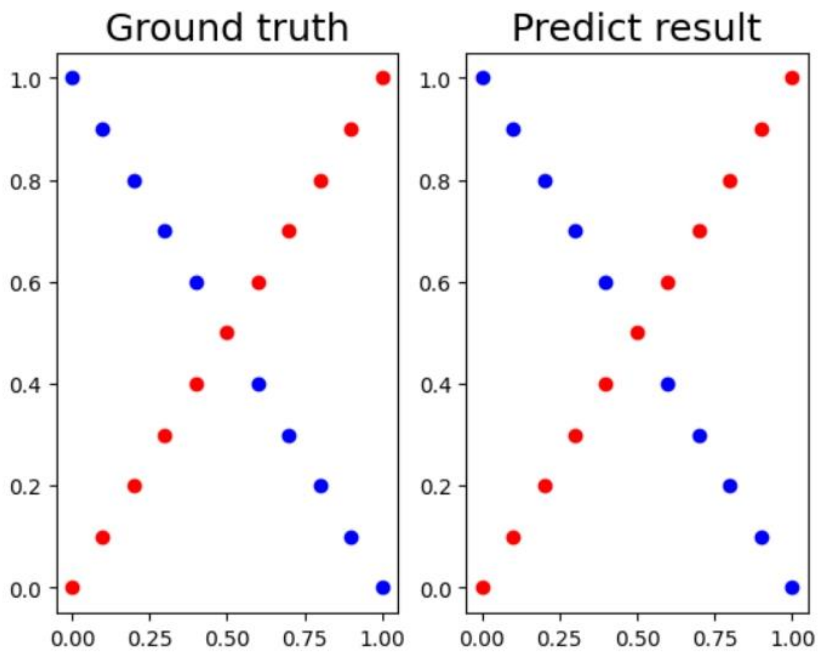
III. Results of your testing

A. Screenshot and comparison figure

- Case 1: Linear



- Case 2: XOR



B. Show the accuracy of your prediction

- Case 1: Linear

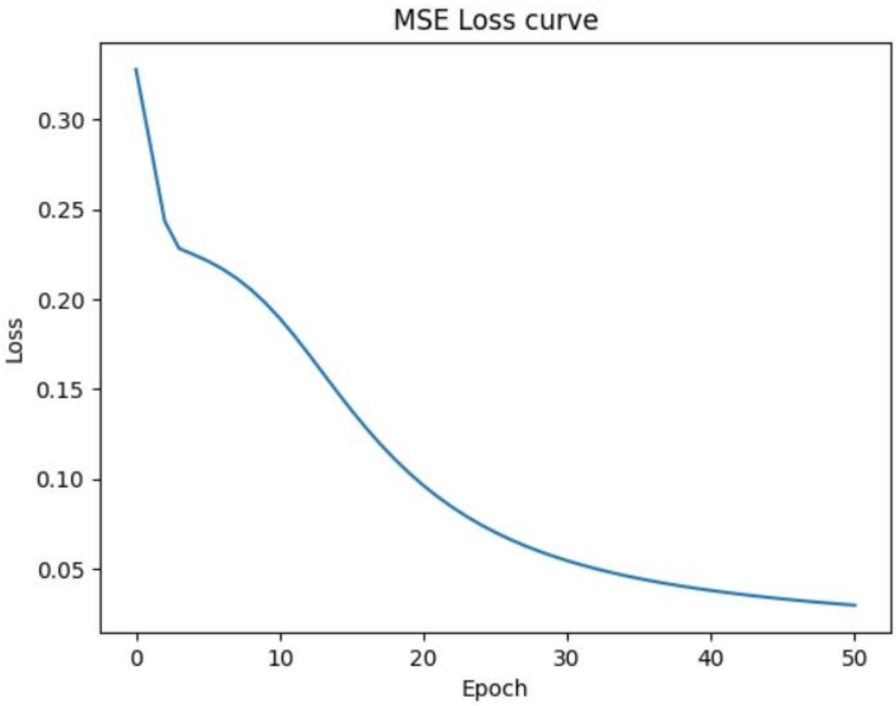
```
Epoch:   0 loss:0.327794 acc:58.00%
Epoch:   5 loss:0.221193 acc:58.00%
Epoch:  10 loss:0.189331 acc:77.00%
Epoch:  15 loss:0.138072 acc:92.00%
Epoch:  20 loss:0.096261 acc:96.00%
Epoch:  25 loss:0.070163 acc:99.00%
Epoch:  30 loss:0.054393 acc:99.00%
Epoch:  35 loss:0.044466 acc:99.00%
Epoch:  40 loss:0.037827 acc:100.00%
Epoch:  45 loss:0.033126 acc:100.00%
Epoch:  50 loss:0.029633 acc:100.00%
```

- Case 2: XOR

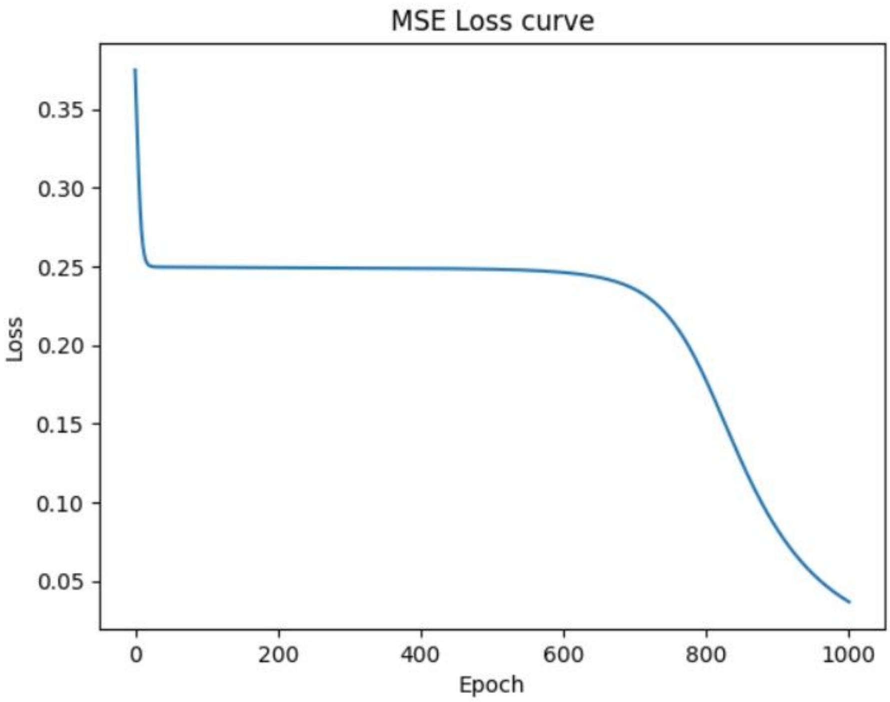
```
Epoch:   0 loss:0.374859 acc:47.62%
Epoch: 100 loss:0.249373 acc:52.38%
Epoch: 200 loss:0.249138 acc:52.38%
Epoch: 300 loss:0.248945 acc:52.38%
Epoch: 400 loss:0.248721 acc:52.38%
Epoch: 500 loss:0.248191 acc:52.38%
Epoch: 600 loss:0.246140 acc:57.14%
Epoch: 700 loss:0.235422 acc:71.43%
Epoch: 800 loss:0.177223 acc:90.48%
Epoch: 900 loss:0.082453 acc:95.24%
Epoch:1000 loss:0.036723 acc:100.00%
```

C. Learning curve(loss, epoch curve)

- Case 1: Linear



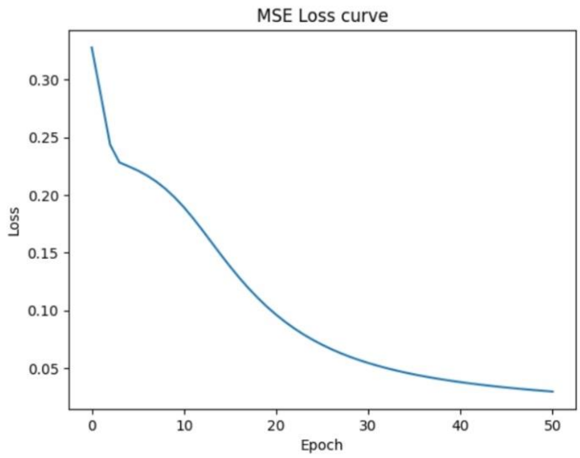
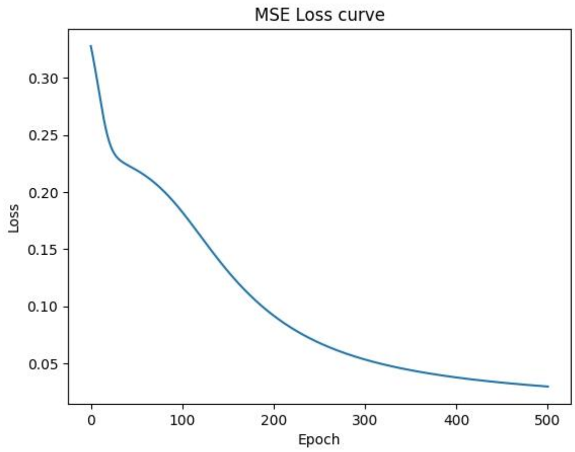
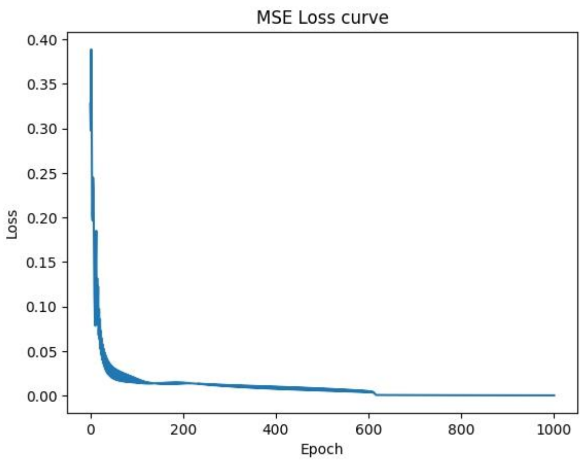
- Case 2: XOR



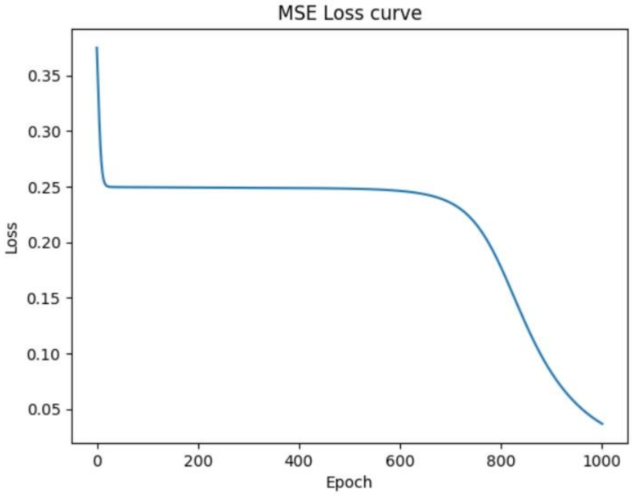
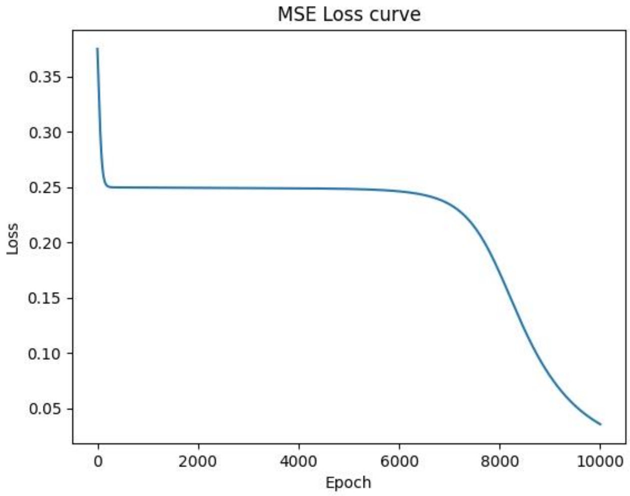
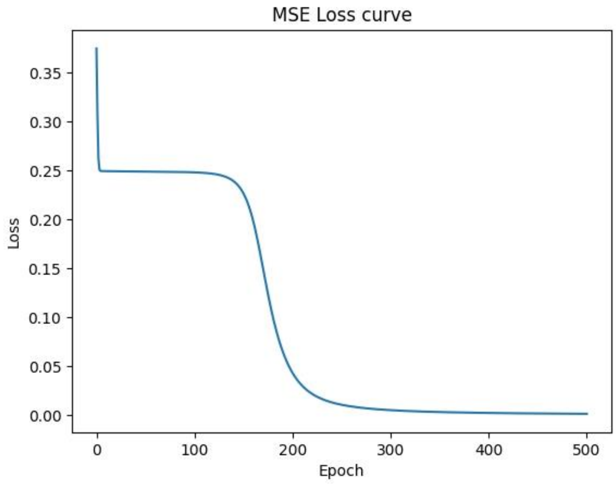
4. Discussion

A. Try different learning rates

- **Case 1: Linear**

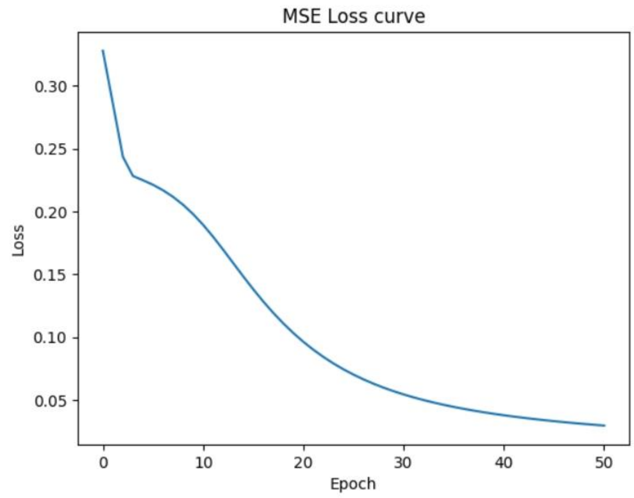
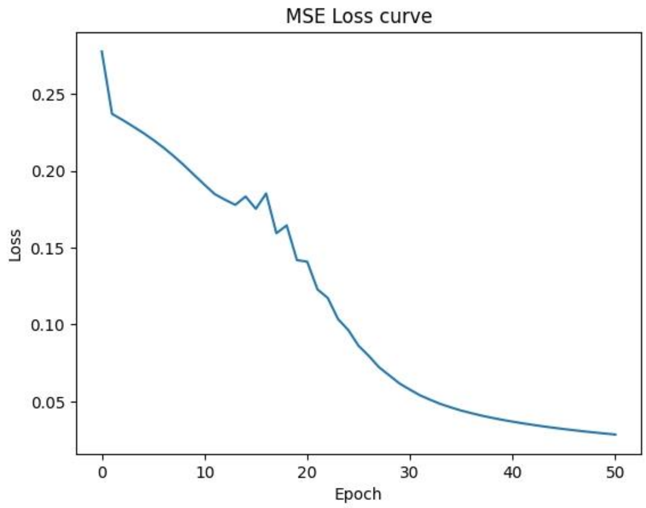
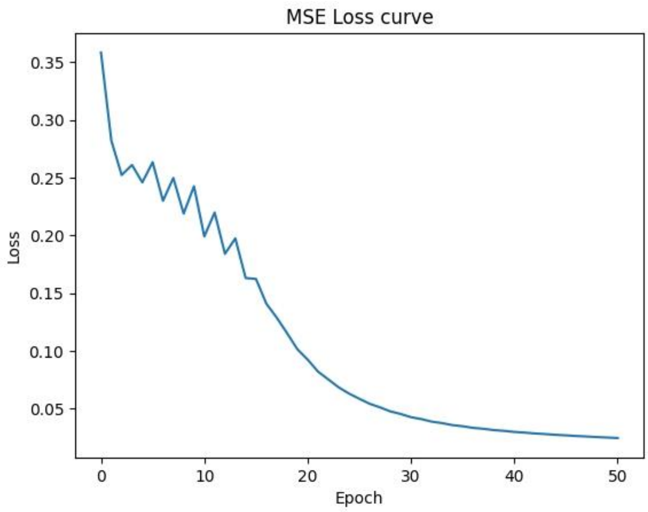
learing_rate	Epochs to acc=100%	Loss curve
0.1	38	
0.01	366	
0.5	560	

- **Case 2: XOR**

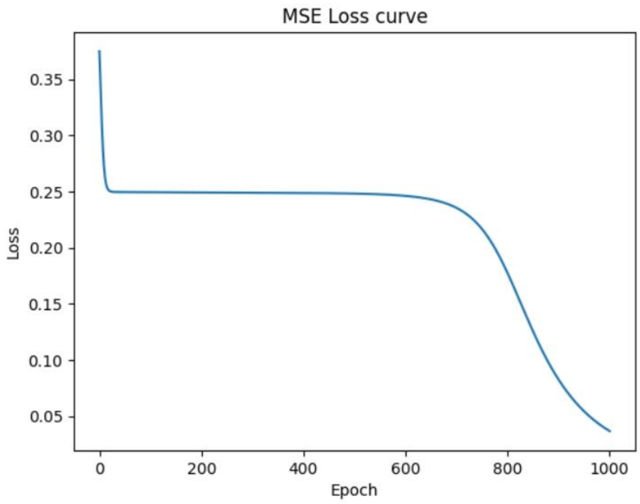
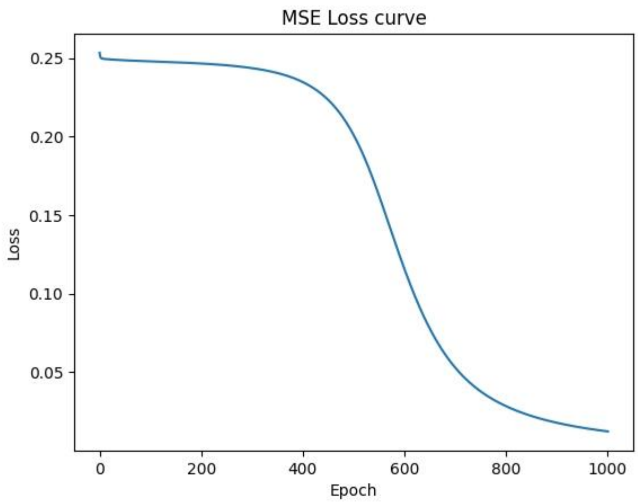
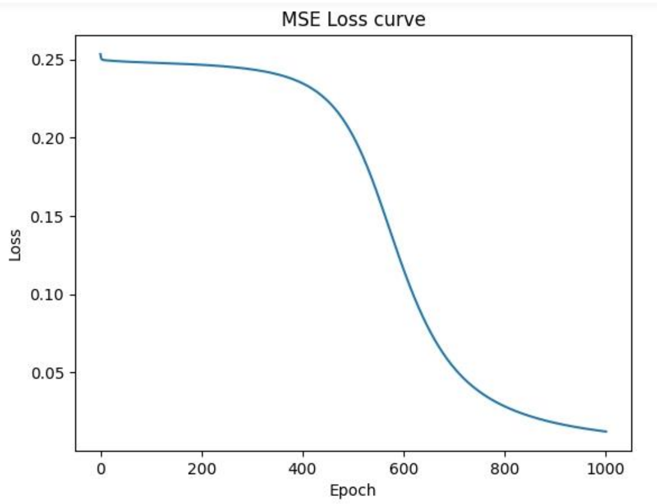
learing_rate	Epochs to acc=100%	Loss curve
0.1	916	
0.01	9100	
0.5	188	

B. Try different numbers of hidden units

- **Case 1: Linear**

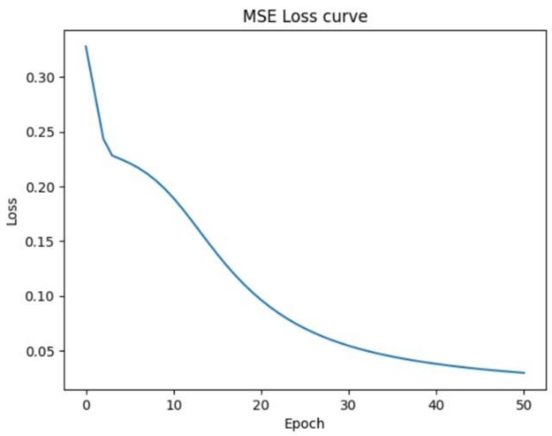
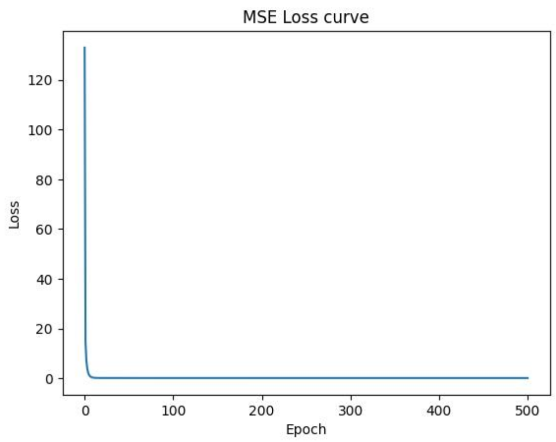
Hidden layers	Epochs to acc=100%	Loss curve
[2, 4, 4, 1]	38	
[2, 6, 6, 1]	33	
[2, 10, 10, 1]	36	

- **Case 2: XOR**

Hidden layers	Epochs to acc=100%	Loss curve
[2, 4, 4, 1]	916	
[2, 6, 6, 1]	661	
[2, 10, 10, 1]	587	

C. Try without activation functions

- **Case 1: Linear**

With sigmoid	Without sigmoid
Learning_rate = 0.1	Learning_rate = 0.0001 (if use 0.1, cannot converge)
Epoch: 0 loss:0.327794 acc:58.00% Epoch: 5 loss:0.221193 acc:58.00% Epoch: 10 loss:0.189331 acc:77.00% Epoch: 15 loss:0.138072 acc:92.00% Epoch: 20 loss:0.096261 acc:96.00% Epoch: 25 loss:0.070163 acc:99.00% Epoch: 30 loss:0.054393 acc:99.00% Epoch: 35 loss:0.044466 acc:99.00% Epoch: 40 loss:0.037827 acc:100.00% Epoch: 45 loss:0.033126 acc:100.00% Epoch: 50 loss:0.029633 acc:100.00%	Epoch: 0 loss:132.981598 acc:1.00% Epoch: 50 loss:0.131182 acc:82.00% Epoch: 100 loss:0.125672 acc:84.00% Epoch: 150 loss:0.123602 acc:84.00% Epoch: 200 loss:0.122839 acc:86.00% Epoch: 250 loss:0.122562 acc:86.00% Epoch: 300 loss:0.122462 acc:87.00% Epoch: 350 loss:0.122426 acc:87.00% Epoch: 400 loss:0.122413 acc:87.00% Epoch: 450 loss:0.122409 acc:87.00% Epoch: 500 loss:0.122407 acc:87.00%
	

- **Case 2: XOR**

<table>
<tr><td>With sigmoid</td><td>Without sigmoid</td></tr>
<tr><td>Learning_rate = 0.1</td><td>Learning_rate = 0.01

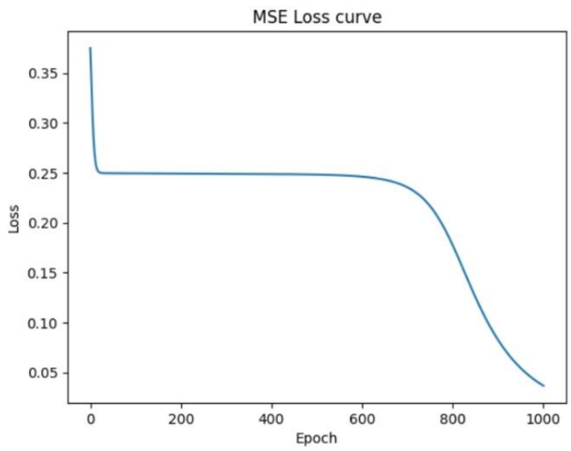
(if use 0.1, cannot converge)</td></tr>
<tr><td>

```
Epoch:   0 loss:0.374859 acc:47.62%
Epoch: 100 loss:0.249373 acc:52.38%
Epoch: 200 loss:0.249138 acc:52.38%
Epoch: 300 loss:0.248945 acc:52.38%
Epoch: 400 loss:0.248721 acc:52.38%
Epoch: 500 loss:0.248191 acc:52.38%
Epoch: 600 loss:0.246140 acc:57.14%
Epoch: 700 loss:0.235422 acc:71.43%
Epoch: 800 loss:0.177223 acc:90.48%
Epoch: 900 loss:0.082453 acc:95.24%
Epoch:1000 loss:0.036723 acc:100.00%
```

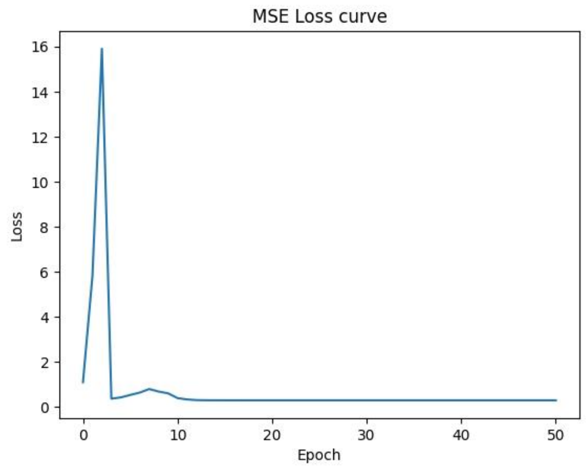
</td><td>

```
Epoch:   0 loss:1.093362 acc:52.38%
Epoch:   5 loss:0.528004 acc:52.38%
Epoch:  10 loss:0.386753 acc:66.67%
Epoch:  15 loss:0.288937 acc:33.33%
Epoch:  20 loss:0.288715 acc:33.33%
Epoch:  25 loss:0.288714 acc:33.33%
Epoch:  30 loss:0.288714 acc:33.33%
Epoch:  35 loss:0.288714 acc:33.33%
Epoch:  40 loss:0.288714 acc:33.33%
Epoch:  45 loss:0.288714 acc:33.33%
Epoch:  50 loss:0.288714 acc:33.33%
```

</td></tr>
<tr><td>



</td><td>

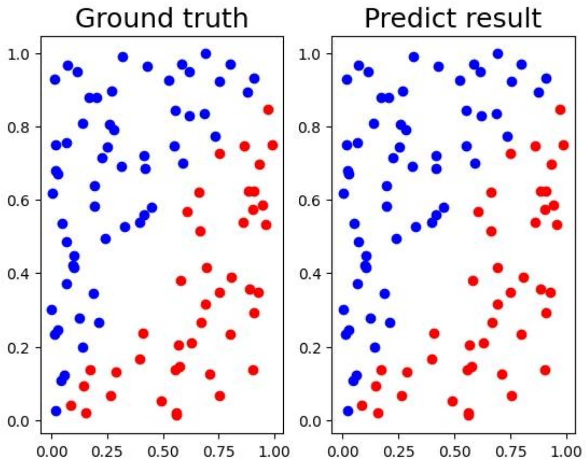
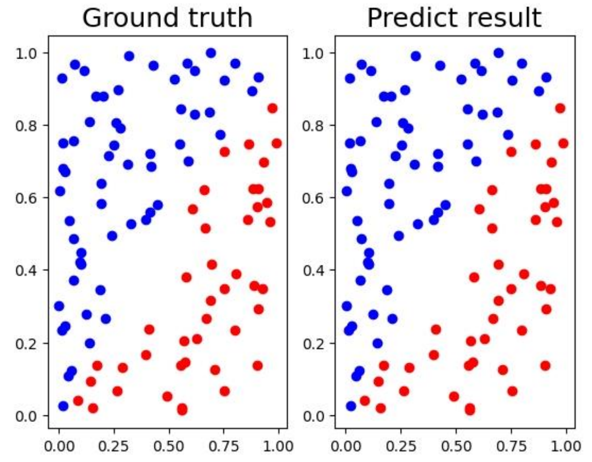
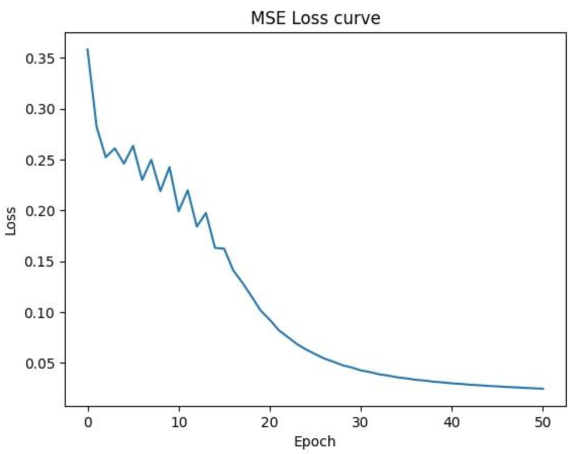
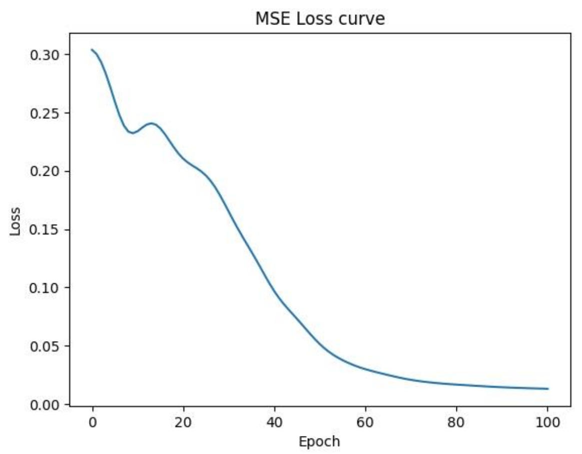


</td></tr>
</table>

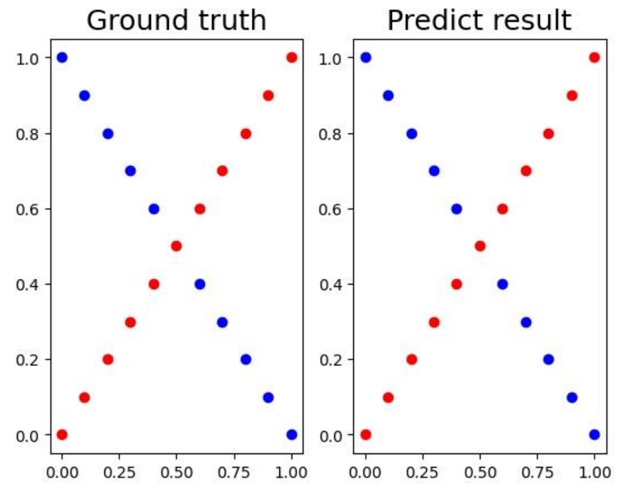
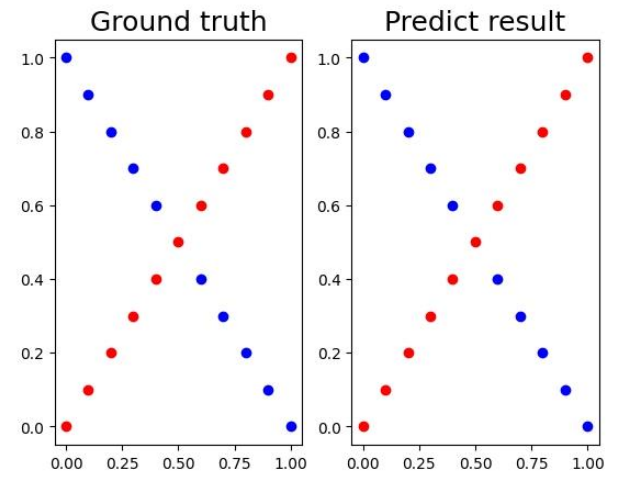
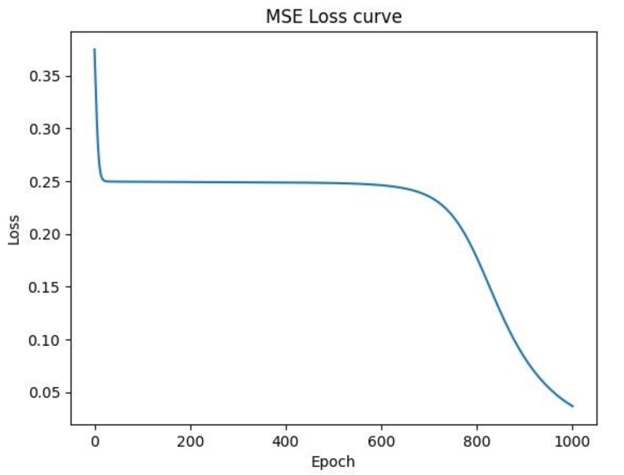
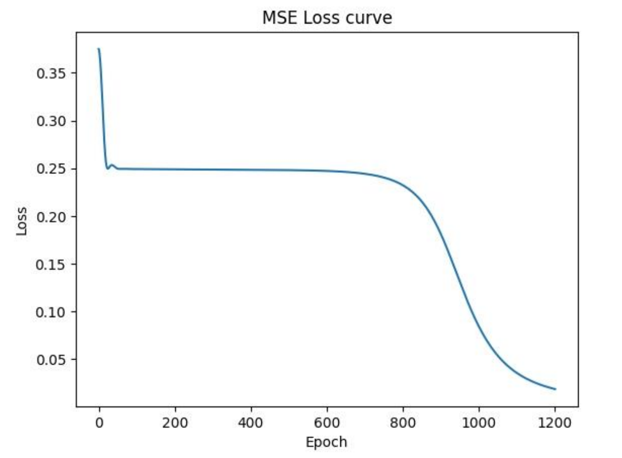
5. Extra

A. Implement different optimizers

- **Case 1: Linear**

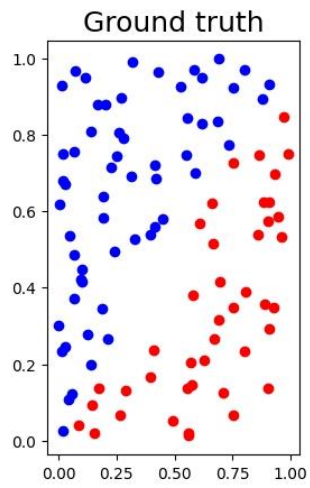
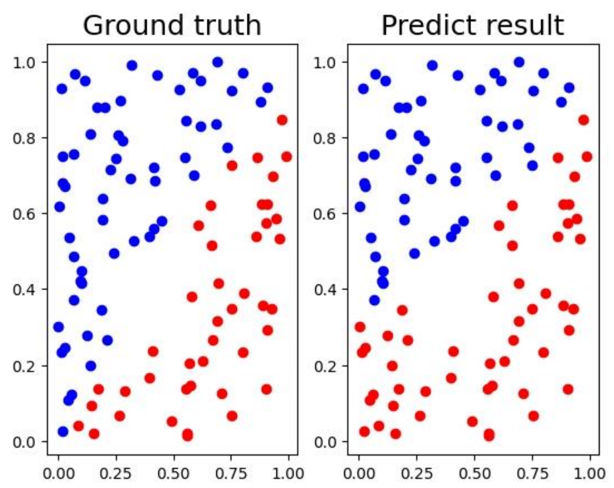
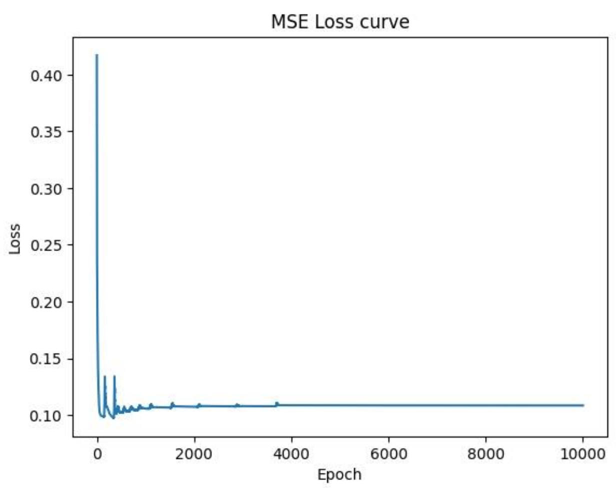
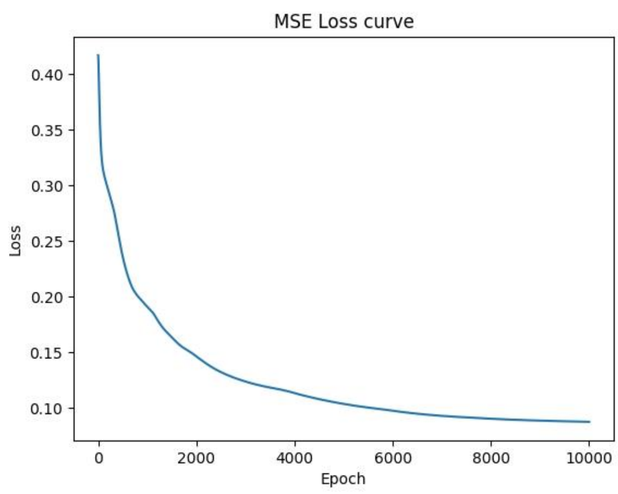
Sigmoid + SGD	**Sigmoid + momentum**
Epoch: 0 loss:0.358400 acc:58.00% Epoch: 5 loss:0.263512 acc:58.00% Epoch: 10 loss:0.199258 acc:70.00% Epoch: 15 loss:0.162447 acc:67.00% Epoch: 20 loss:0.092607 acc:90.00% Epoch: 25 loss:0.058728 acc:94.00% Epoch: 30 loss:0.042800 acc:97.00% Epoch: 35 loss:0.034942 acc:98.00% Epoch: 40 loss:0.030036 acc:100.00% Epoch: 45 loss:0.027053 acc:100.00% Epoch: 50 loss:0.024744 acc:100.00%	Epoch: 0 loss:0.303472 acc:58.00% Epoch: 10 loss:0.233734 acc:74.00% Epoch: 20 loss:0.210273 acc:71.00% Epoch: 30 loss:0.164940 acc:83.00% Epoch: 40 loss:0.096741 acc:95.00% Epoch: 50 loss:0.051386 acc:96.00% Epoch: 60 loss:0.029887 acc:98.00% Epoch: 70 loss:0.020754 acc:99.00% Epoch: 80 loss:0.016687 acc:100.00% Epoch: 90 loss:0.014425 acc:100.00% Epoch: 100 loss:0.013088 acc:100.00%
	
	

- **Case 2: XOR**

Sigmoid + SGD	**Sigmoid + momentum**
Epoch: 0 loss:0.374859 acc:47.62% Epoch: 100 loss:0.249373 acc:52.38% Epoch: 200 loss:0.249138 acc:52.38% Epoch: 300 loss:0.248945 acc:52.38% Epoch: 400 loss:0.248721 acc:52.38% Epoch: 500 loss:0.248191 acc:52.38% Epoch: 600 loss:0.246140 acc:57.14% Epoch: 700 loss:0.235422 acc:71.43% Epoch: 800 loss:0.177223 acc:90.48% Epoch: 900 loss:0.082453 acc:95.24% Epoch:1000 loss:0.036723 acc:100.00%	Epoch: 0 loss:0.374859 acc:47.62% Epoch: 120 loss:0.249060 acc:52.38% Epoch: 240 loss:0.248729 acc:52.38% Epoch: 360 loss:0.248454 acc:52.38% Epoch: 480 loss:0.248127 acc:52.38% Epoch: 600 loss:0.247180 acc:47.62% Epoch: 720 loss:0.242926 acc:66.67% Epoch: 840 loss:0.219451 acc:90.48% Epoch: 960 loss:0.120322 acc:95.24% Epoch:1080 loss:0.041479 acc:100.00% Epoch:1200 loss:0.018676 acc:100.00%
	
	

B. Implement different activation functions

- **Case 1: Linear**

<table>
<tr><th>Relu + SGD</th><th>Relu + momentum</th></tr>
<tr><td>Epoch: 0 loss:0.417119 acc:42.00%
Epoch:1000 loss:0.105709 acc:88.00%
Epoch:2000 loss:0.107252 acc:88.00%
Epoch:3000 loss:0.107806 acc:89.00%
Epoch:4000 loss:0.108537 acc:87.00%
Epoch:5000 loss:0.108462 acc:87.00%
Epoch:6000 loss:0.108420 acc:87.00%
Epoch:7000 loss:0.108394 acc:87.00%
Epoch:8000 loss:0.108376 acc:87.00%
Epoch:9000 loss:0.108364 acc:88.00%
Epoch:10000 loss:0.108354 acc:88.00%</td><td>Epoch: 0 loss:0.417119 acc:42.00%
Epoch:1000 loss:0.190789 acc:72.00%
Epoch:2000 loss:0.146076 acc:76.00%
Epoch:3000 loss:0.123402 acc:80.00%
Epoch:4000 loss:0.112995 acc:83.00%
Epoch:5000 loss:0.103259 acc:85.00%
Epoch:6000 loss:0.097123 acc:87.00%
Epoch:7000 loss:0.092489 acc:88.00%
Epoch:8000 loss:0.089913 acc:89.00%
Epoch:9000 loss:0.088215 acc:89.00%
Epoch:10000 loss:0.087129 acc:89.00%</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

- **Case 2: XOR**

Relu + SGD	**Relu + momentum**
Epoch: 0 loss:0.376679 acc:52.38% Epoch:1000 loss:0.086383 acc:90.48% Epoch:2000 loss:0.087548 acc:90.48% Epoch:3000 loss:0.085431 acc:90.48% Epoch:4000 loss:0.085886 acc:90.48% Epoch:5000 loss:0.085632 acc:90.48% Epoch:6000 loss:0.085172 acc:90.48% Epoch:7000 loss:0.088785 acc:85.71% Epoch:8000 loss:0.086329 acc:90.48% Epoch:9000 loss:0.084738 acc:90.48% Epoch:10000 loss:0.088944 acc:85.71%	Epoch: 0 loss:0.376679 acc:52.38% Epoch: 100 loss:0.195015 acc:66.67% Epoch: 200 loss:0.183786 acc:71.43% Epoch: 300 loss:0.184458 acc:71.43% Epoch: 400 loss:0.183645 acc:71.43% Epoch: 500 loss:0.182706 acc:71.43% Epoch: 600 loss:0.473566 acc:52.38% Epoch: 700 loss:0.144633 acc:80.95% Epoch: 800 loss:0.332644 acc:57.14% Epoch: 900 loss:0.202241 acc:80.95% Epoch:1000 loss:0.201889 acc:80.95%