## 1. Introduction (20%)
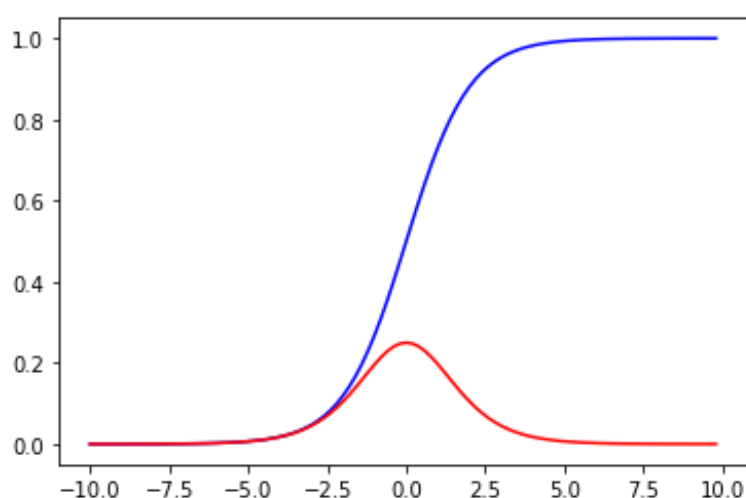
這次作業是實作backpropagation, 建立一個2個hidden layers的neural network, input為generate linear和generate XOR, 透過forward和backpropagation來classify input data, 用cross entropy來當loss function.

## 2. Experiment setups (30%)

(a) Sigmoid functions



```python
def sigmoid(x):
    return 1.0/(1.0 + np.exp(-x))

def derivative_sigmoid(x):
    return np.multiply(x, 1.0 - x)
```

(b) Neural network

我使用2個hidden layers的neural network, 每層10個neuron, loss function使用cross entropy, learning rate = 0.1

**Cross Entropy:**

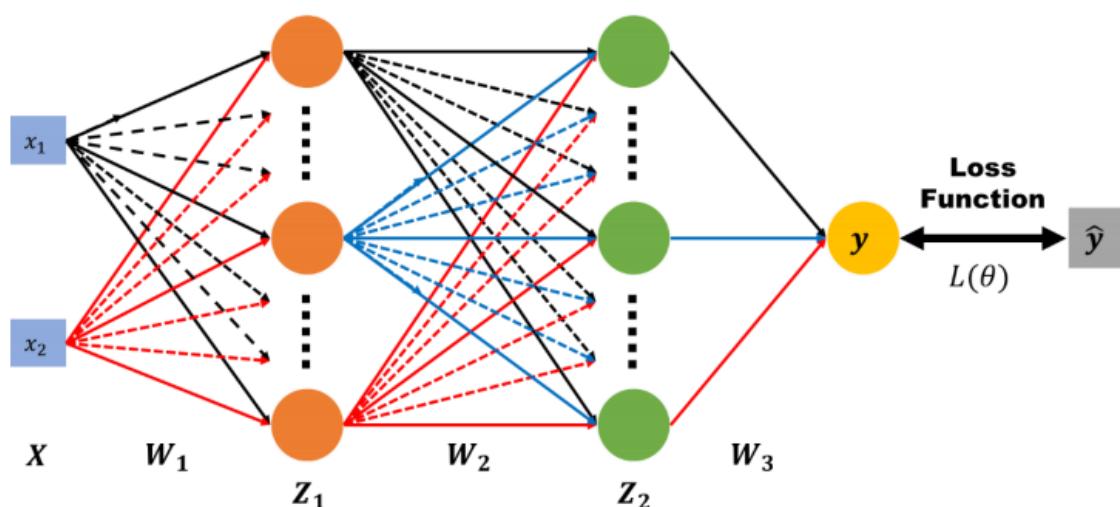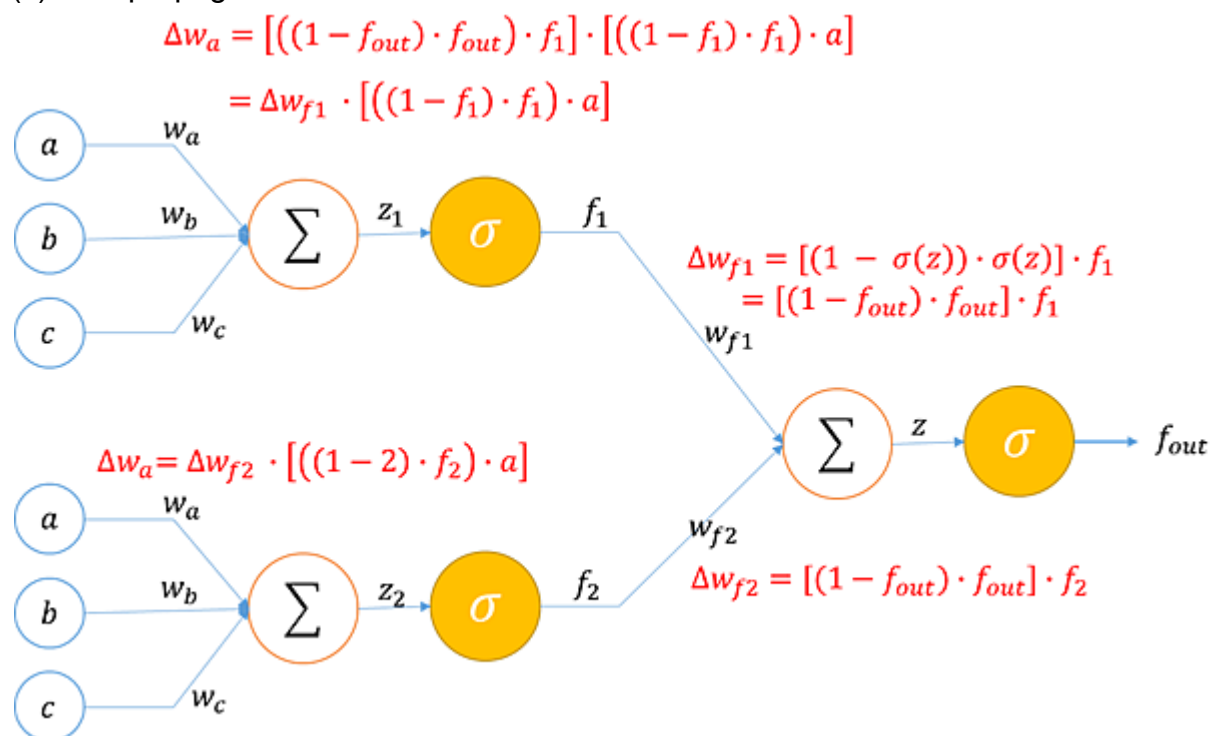$$-(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$
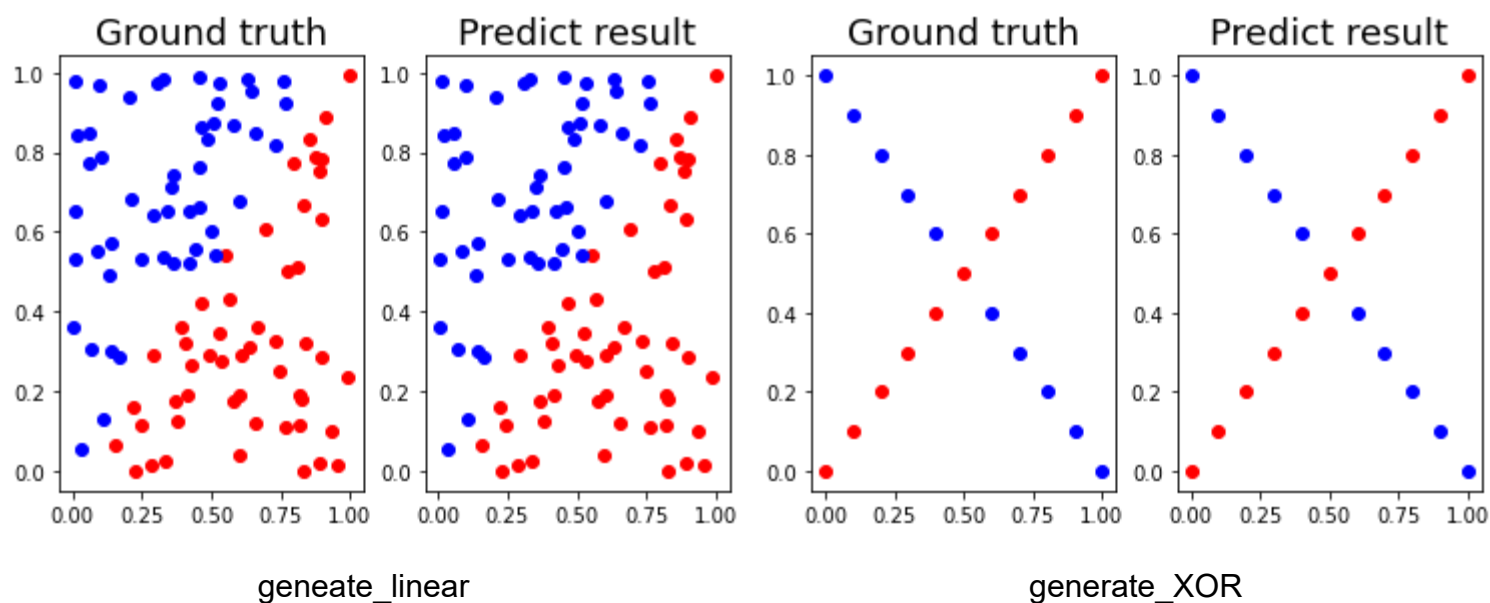


**Figure 2. Forward pass**

(c) Backpropagation

$$\Delta w_a = \left[\left((1-f_{out})\cdot f_{out}\right)\cdot f_1\right]\cdot\left[\left((1-f_1)\cdot f_1\right)\cdot a\right]$$

$$= \Delta w_{f1}\cdot\left[\left((1-f_1)\cdot f_1\right)\cdot a\right]$$



$$\Delta w_{f1} = \left[(1-\sigma(z))\cdot\sigma(z)\right]\cdot f_1$$
$$= \left[(1-f_{out})\cdot f_{out}\right]\cdot f_1$$

$$\Delta w_a = \Delta w_{f2}\cdot\left[\left((1-2)\cdot f_2\right)\cdot a\right]$$

$$\Delta w_{f2} = \left[(1-f_{out})\cdot f_{out}\right]\cdot f_2$$

## 3. Results of your testing (20%)

(a)Screenshot and comparison figure



geneate_linear                         generate_XOR

(b) Show the accuracy of your prediction
Both geneate_linear and geneate_XOR accuracy on training and testing are 100%.

geneate_linear:

```
Epochs 0: loss=0.90480 accuracy=49.00%
Epochs 500: loss=0.31900 accuracy=99.00%
Epochs 1000: loss=0.13974 accuracy=99.00%
Epochs 1500: loss=0.08767 accuracy=100.00%
Epochs 2000: loss=0.06433 accuracy=100.00%
Epochs 2500: loss=0.05115 accuracy=100.00%
Epochs 3000: loss=0.04265 accuracy=100.00%
Epochs 3500: loss=0.03669 accuracy=100.00%
Epochs 4000: loss=0.03225 accuracy=100.00%
Epochs 4500: loss=0.02879 accuracy=100.00%
Epochs 5000: loss=0.02601 accuracy=100.00%
Epochs 5500: loss=0.02372 accuracy=100.00%
Epochs 6000: loss=0.02179 accuracy=100.00%
Epochs 6500: loss=0.02013 accuracy=100.00%
Epochs 7000: loss=0.01869 accuracy=100.00%
Epochs 7500: loss=0.01743 accuracy=100.00%
Epochs 8000: loss=0.01631 accuracy=100.00%
Epochs 8500: loss=0.01531 accuracy=100.00%
Epochs 9000: loss=0.01440 accuracy=100.00%
Epochs 9500: loss=0.01358 accuracy=100.00%
Epochs 10000: loss=0.01283 accuracy=100.00%
training finished
```

```
start testing:
[[2.32371574e-05 9.99996562e-01 1.49132778e-06 9.39193535e-01
  7.41157943e-06 3.60473294e-06 1.35447086e-06 5.04175721e-02
  1.87262864e-06 2.90045284e-06 2.16303537e-05 9.99995933e-01
  9.81669381e-01 1.74629946e-06 6.79898924e-06 3.98961447e-04
  9.99992005e-01 1.04539650e-05 1.15324411e-03 1.30103195e-05
  9.99987409e-01 2.30128908e-04 9.99995925e-01 2.68339460e-01
  2.28149135e-06 9.99312861e-01 9.99998230e-01 2.22728993e-06
  8.33228929e-01 1.92418046e-06 5.24316592e-05 4.36979006e-06
  9.99998142e-01 3.90563977e-06 9.99160490e-01 9.99786370e-01
  2.35328153e-01 2.54314786e-06 9.99857772e-01 2.61258854e-03
  9.99997930e-01 1.58791069e-06 9.99937898e-01 2.57071630e-06
  9.99454549e-01 9.90136666e-03 9.98864861e-01 9.25625834e-01
  9.99988485e-01 1.90797434e-05 1.22944369e-02 9.96877211e-01
  9.99995177e-01 9.99994489e-01 1.20507829e-06 3.86558705e-06
  9.99644410e-01 9.10146661e-01 9.99994200e-01 1.19294907e-05
  9.99996052e-01 9.15384356e-01 9.98678324e-01 1.69960656e-05
  9.65162730e-01 9.99998161e-01 9.99861654e-01 9.86600890e-06
  8.71828101e-03 4.93913932e-04 1.24512341e-06 5.73821507e-06
  3.57870463e-04 9.99846806e-01 9.99997760e-01 9.94949825e-01
  1.98461061e-04 9.99992098e-01 1.95089709e-04 1.27762872e-03
  9.99201109e-01 3.96726793e-05 1.07504682e-05 9.99733877e-01
  9.99958923e-01 9.99993244e-01 9.99992395e-01 9.99998087e-01
  1.44122427e-04 9.99866223e-01 9.99997884e-01 9.99998249e-01
  1.42159001e-06 2.69583740e-05 9.99983440e-01 9.81027008e-01
  5.48383634e-03 9.54497662e-01 9.99972663e-01 1.49382744e-06]]
loss=0.01215 accuracy=100.00%
testing finished
```

geneate_XOR:

```
Epochs 0: loss=0.68826 accuracy=52.38%
Epochs 500: loss=0.67891 accuracy=66.67%
Epochs 1000: loss=0.66545 accuracy=66.67%
Epochs 1500: loss=0.63324 accuracy=71.43%
Epochs 2000: loss=0.54627 accuracy=85.71%
Epochs 2500: loss=0.40995 accuracy=85.71%
Epochs 3000: loss=0.29888 accuracy=90.48%
Epochs 3500: loss=0.22602 accuracy=90.48%
Epochs 4000: loss=0.17505 accuracy=90.48%
Epochs 4500: loss=0.13524 accuracy=95.24%
Epochs 5000: loss=0.10302 accuracy=100.00%
Epochs 5500: loss=0.07789 accuracy=100.00%
Epochs 6000: loss=0.05924 accuracy=100.00%
Epochs 6500: loss=0.04581 accuracy=100.00%
Epochs 7000: loss=0.03620 accuracy=100.00%
Epochs 7500: loss=0.02925 accuracy=100.00%
Epochs 8000: loss=0.02412 accuracy=100.00%
Epochs 8500: loss=0.02026 accuracy=100.00%
Epochs 9000: loss=0.01729 accuracy=100.00%
Epochs 9500: loss=0.01495 accuracy=100.00%
Epochs 10000: loss=0.01308 accuracy=100.00%
training finished
```

```
start testing:
[[0.00437284 0.99960899 0.00608885 0.99954114 0.00893606 0.99933218
  0.0127496  0.9980946  0.01644745 0.93081042 0.01839657 0.01775664
  0.94375973 0.01514851 0.99924859 0.01186132 0.99951256 0.00885679
  0.99934882 0.00650625 0.99899655]]
loss=0.01155 accuracy=100.00%
testing finished
```
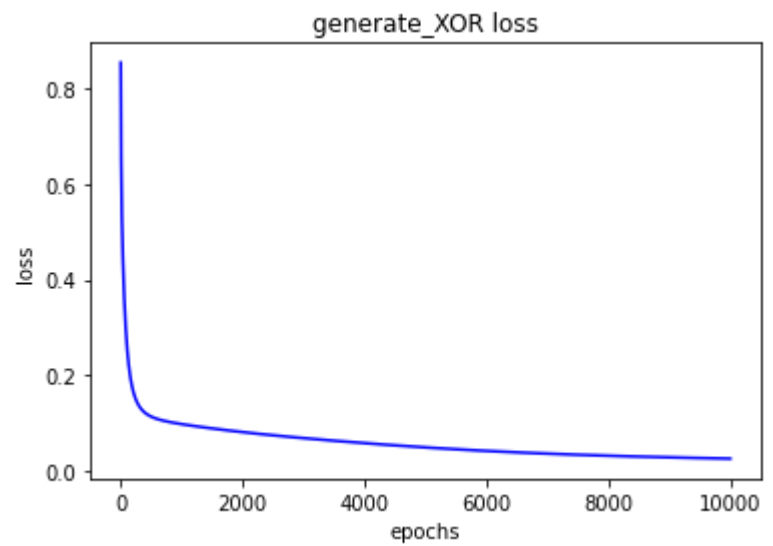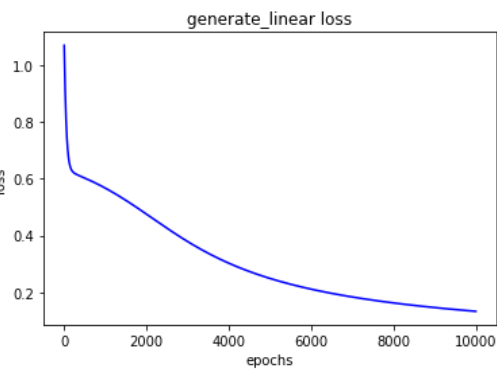
(c) Learning curve (loss, epoch curve)

geneate_linear:

geneate_XOR:

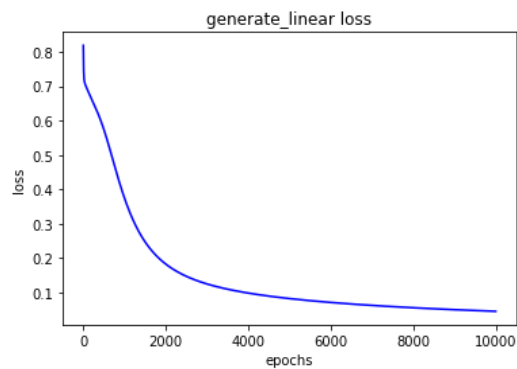### generate_linear loss



### generate_XOR loss
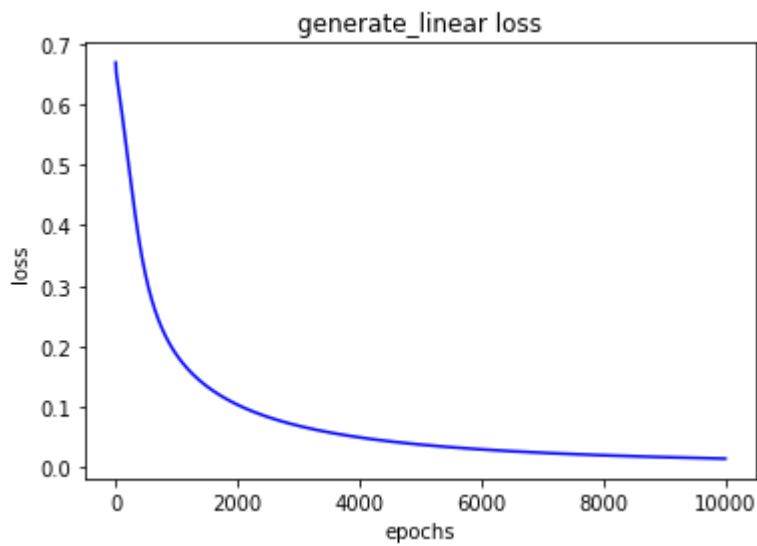


## 4. Discussion (30%)

(a) Try different learning rates on `genetate_linear`

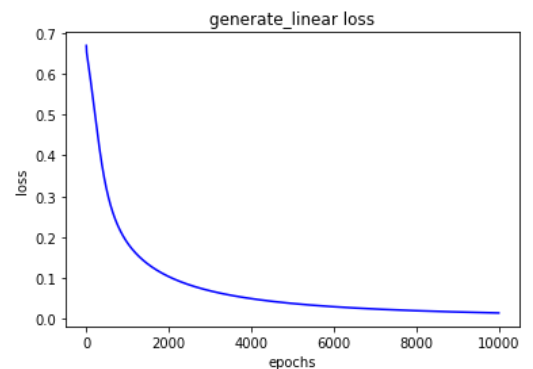learning rate=0.01,

learning rate=0.05,

learning rate=0.1

### generate_linear loss



### generate_linear loss



### generate_linear loss
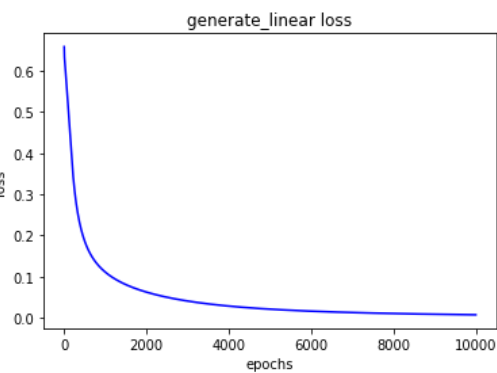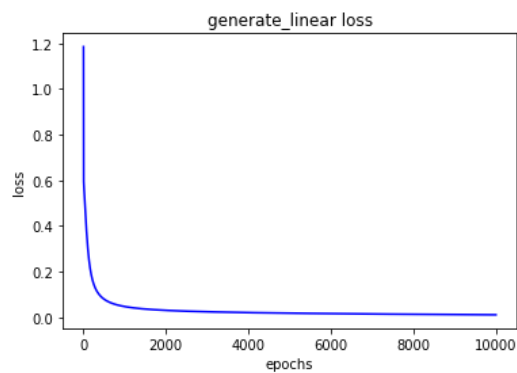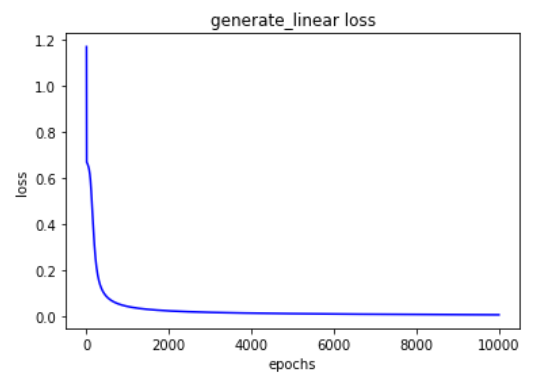


learning rate=0.2,

learning rate=0.3,

learning rate=0.5

### generate_linear loss
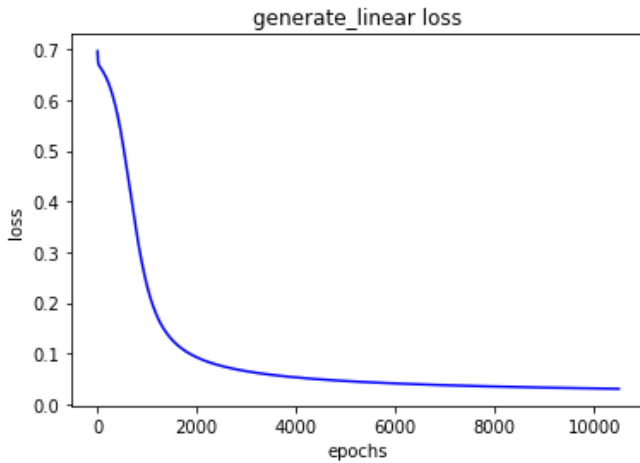


### generate_linear loss



### generate_linear loss

## (b) Try different numbers of hidden units on `genetate_linear`

### (1)2個hidden layers, 每層5個hidden units



```
Epochs 0: loss=0.69667 accuracy=44.00%
Epochs 500: loss=0.51966 accuracy=90.00%
Epochs 1000: loss=0.23240 accuracy=97.00%
Epochs 1500: loss=0.12703 accuracy=98.00%
Epochs 2000: loss=0.09225 accuracy=98.00%
Epochs 2500: loss=0.07519 accuracy=99.00%
Epochs 3000: loss=0.06486 accuracy=99.00%
Epochs 3500: loss=0.05783 accuracy=99.00%
Epochs 4000: loss=0.05268 accuracy=99.00%
Epochs 4500: loss=0.04872 accuracy=99.00%
Epochs 5000: loss=0.04556 accuracy=99.00%
Epochs 5500: loss=0.04296 accuracy=99.00%
Epochs 6000: loss=0.04079 accuracy=99.00%
Epochs 6500: loss=0.03894 accuracy=99.00%
Epochs 7000: loss=0.03734 accuracy=99.00%
Epochs 7500: loss=0.03594 accuracy=99.00%
Epochs 8000: loss=0.03471 accuracy=99.00%
Epochs 8500: loss=0.03360 accuracy=99.00%
Epochs 9000: loss=0.03261 accuracy=99.00%
Epochs 9500: loss=0.03172 accuracy=99.00%
Epochs 10000: loss=0.03090 accuracy=99.00%
```

### (2)2個hidden layers, 每層10個hidden units
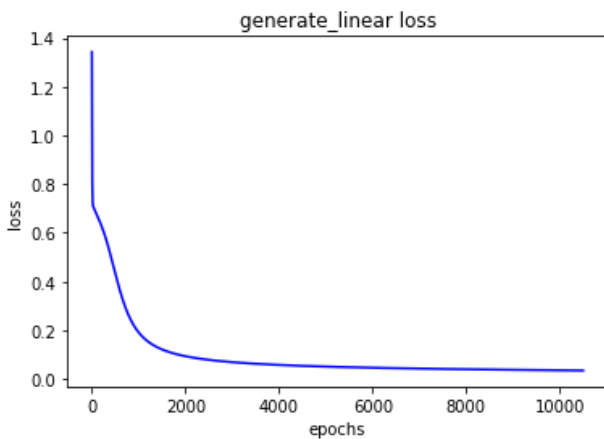


```
Epochs 0: loss=1.47746 accuracy=50.00%
Epochs 500: loss=0.42814 accuracy=95.00%
Epochs 1000: loss=0.12886 accuracy=100.00%
Epochs 1500: loss=0.06351 accuracy=100.00%
Epochs 2000: loss=0.04130 accuracy=100.00%
Epochs 2500: loss=0.03039 accuracy=100.00%
Epochs 3000: loss=0.02394 accuracy=100.00%
Epochs 3500: loss=0.01967 accuracy=100.00%
Epochs 4000: loss=0.01664 accuracy=100.00%
Epochs 4500: loss=0.01437 accuracy=100.00%
Epochs 5000: loss=0.01261 accuracy=100.00%
Epochs 5500: loss=0.01120 accuracy=100.00%
Epochs 6000: loss=0.01005 accuracy=100.00%
Epochs 6500: loss=0.00909 accuracy=100.00%
Epochs 7000: loss=0.00828 accuracy=100.00%
Epochs 7500: loss=0.00758 accuracy=100.00%
Epochs 8000: loss=0.00697 accuracy=100.00%
Epochs 8500: loss=0.00644 accuracy=100.00%
Epochs 9000: loss=0.00598 accuracy=100.00%
Epochs 9500: loss=0.00556 accuracy=100.00%
Epochs 10000: loss=0.00519 accuracy=100.00%
```

### (3)2個hidden layers, 每層15個hidden units
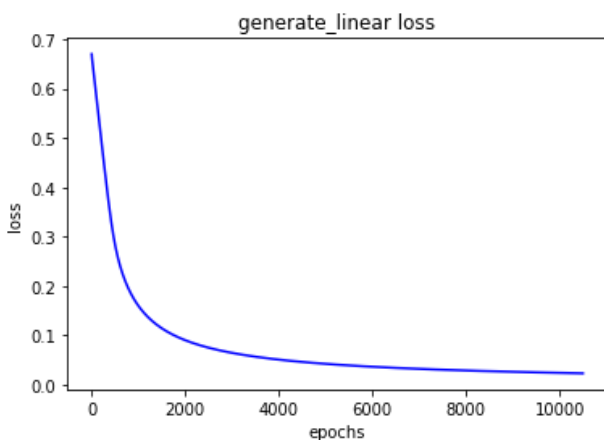


```
Epochs 0: loss=0.96956 accuracy=50.00%
Epochs 500: loss=0.15120 accuracy=99.00%
Epochs 1000: loss=0.09233 accuracy=99.00%
Epochs 1500: loss=0.06869 accuracy=100.00%
Epochs 2000: loss=0.05531 accuracy=100.00%
Epochs 2500: loss=0.04648 accuracy=100.00%
Epochs 3000: loss=0.04014 accuracy=100.00%
Epochs 3500: loss=0.03534 accuracy=100.00%
Epochs 4000: loss=0.03157 accuracy=100.00%
Epochs 4500: loss=0.02852 accuracy=100.00%
Epochs 5000: loss=0.02601 accuracy=100.00%
Epochs 5500: loss=0.02390 accuracy=100.00%
Epochs 6000: loss=0.02211 accuracy=100.00%
Epochs 6500: loss=0.02057 accuracy=100.00%
Epochs 7000: loss=0.01923 accuracy=100.00%
Epochs 7500: loss=0.01805 accuracy=100.00%
Epochs 8000: loss=0.01702 accuracy=100.00%
Epochs 8500: loss=0.01609 accuracy=100.00%
Epochs 9000: loss=0.01527 accuracy=100.00%
Epochs 9500: loss=0.01452 accuracy=100.00%
Epochs 10000: loss=0.01385 accuracy=100.00%
```

(4)2個hidden layers, 每層20個hidden units
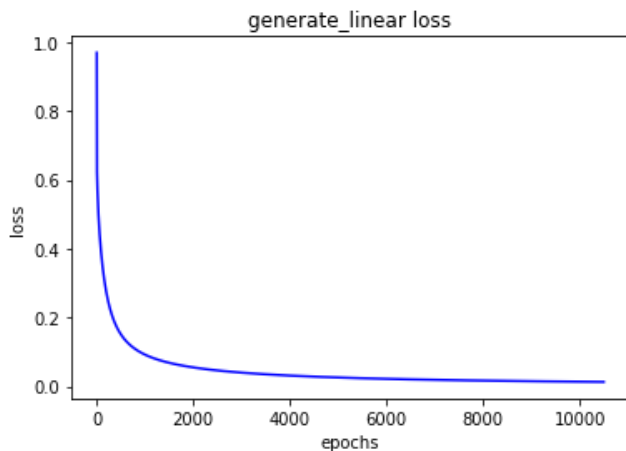


```
Epochs 0: loss=0.67017 accuracy=64.00%
Epochs 500: loss=0.28284 accuracy=96.00%
Epochs 1000: loss=0.15906 accuracy=100.00%
Epochs 1500: loss=0.11364 accuracy=100.00%
Epochs 2000: loss=0.08949 accuracy=100.00%
Epochs 2500: loss=0.07432 accuracy=100.00%
Epochs 3000: loss=0.06382 accuracy=100.00%
Epochs 3500: loss=0.05609 accuracy=100.00%
Epochs 4000: loss=0.05015 accuracy=100.00%
Epochs 4500: loss=0.04543 accuracy=100.00%
Epochs 5000: loss=0.04159 accuracy=100.00%
Epochs 5500: loss=0.03840 accuracy=100.00%
Epochs 6000: loss=0.03571 accuracy=100.00%
Epochs 6500: loss=0.03340 accuracy=100.00%
Epochs 7000: loss=0.03139 accuracy=100.00%
Epochs 7500: loss=0.02964 accuracy=100.00%
Epochs 8000: loss=0.02809 accuracy=100.00%
Epochs 8500: loss=0.02670 accuracy=100.00%
Epochs 9000: loss=0.02545 accuracy=100.00%
Epochs 9500: loss=0.02433 accuracy=100.00%
Epochs 10000: loss=0.02330 accuracy=100.00%
```

(c) Anything you want to share

backpropagation有個地方是要取 sigmoid 的微分, 一開始以為 `derivative_sigmoid(x) = x*(1-x)` 我一開始input x進去算sigmoid 的微分發現梯度會爆炸, 後來上網查發現input x要先取 sigmoid 之後再input 到 sigmoid 的微分函式才是對的, 所以要特別注意~

**5. Extra (10%)**

這次datasets相對簡單, 甚至 `geneate_XOR` 只有11個點, 所以epoch不用太長, 也不需要用到複雜的最佳化函數(eg. adam) 也可以有很好的效果, 像我這次實作大概1000 epochs 就可以到 training 跟 testing accuracy 100%