

## 一、展示權值矩陣 W1 與 W2 之初始值、訓練變化、最終值

initialize W1: [[-0.32344333 0.95318594][ 0.42953142 0.04554452]]

initialize W2: [[-0.99459151 0.28564619][ 0.23852008 -1.9424014]]

final W1: [[-7.26064143 7.51015781][-7.0499215 6.74125435]]

final W2: [[-7.26064143 7.51015781][-7.0499215 6.74125435]]

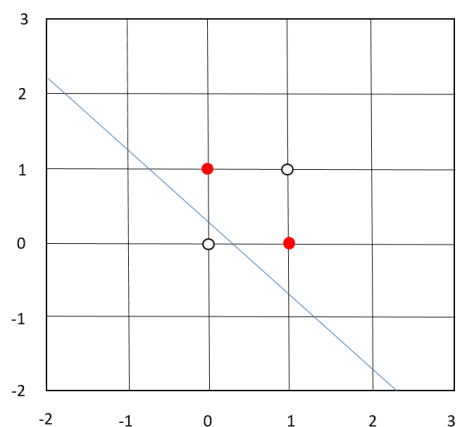
訓練變化：

W1: [[-0.32350346 0.95314663] [ 0.42900835 0.04511581]]  
W2: [[-0.32350346 0.95314663] [ 0.42900835 0.04511581]]  
W1: [[-0.35315989 0.98963714] [ 0.282622 0.09711236]]  
W2: [[-0.35315989 0.98963714] [ 0.282622 0.09711236]]  
W1: [[-0.42553045 1.06160455] [ 0.21244504 0.17067605]]  
W2: [[-0.42553045 1.06160455] [ 0.21244504 0.17067605]]  
W1: [[-0.55774084 1.19187385] [ 0.16241042 0.24207431]]  
W2: [[-0.55774084 1.19187385] [ 0.16241042 0.24207431]]  
W1: [[-0.68635132 1.3208844 ] [ 0.13929123 0.29180403]]  
W2: [[-0.68635132 1.3208844 ] [ 0.13929123 0.29180403]]  
W1: [[-0.88557726 1.52080517] [ 0.11981572 0.35744587]]  
W2: [[-0.88557726 1.52080517] [ 0.11981572 0.35744587]]  
W1: [[-1.20126904 1.83249069] [ 0.10237452 0.45441817]]  
W2: [[-1.20126904 1.83249069] [ 0.10237452 0.45441817]]  
W1: [[-1.55841828 2.17975595] [ 0.08657278 0.56661926]]  
W2: [[-1.55841828 2.17975595] [ 0.08657278 0.56661926]]  
W1: [[-1.99599328 2.60697436] [ 0.06213501 0.71726726]]  
W2: [[-1.99599328 2.60697436] [ 0.06213501 0.71726726]]  
W1: [[-2.97495047 3.59904059] [-0.08924161 1.12167962]]  
W2: [[-2.97495047 3.59904059] [-0.08924161 1.12167962]]  
W1: [[-3.51064088 4.1498537 ] [-0.4288582 1.32337367]]  
W2: [[-3.51064088 4.1498537 ] [-0.4288582 1.32337367]]  
W1: [[-4.43488828 4.94816151] [-2.37804503 2.20786365]]  
W2: [[-4.43488828 4.94816151] [-2.37804503 2.20786365]]  
W1: [[-4.8319147 5.252825 ] [-3.26810892 3.0039793 ]]  
W2: [[-4.8319147 5.252825 ] [-3.26810892 3.0039793 ]]  
W1: [[-5.04542216 5.41906248] [-3.70524374 3.42020938]]  
W2: [[-5.04542216 5.41906248] [-3.70524374 3.42020938]]  
W1: [[-5.37202644 5.68449807] [-4.31881254 4.01731447]]

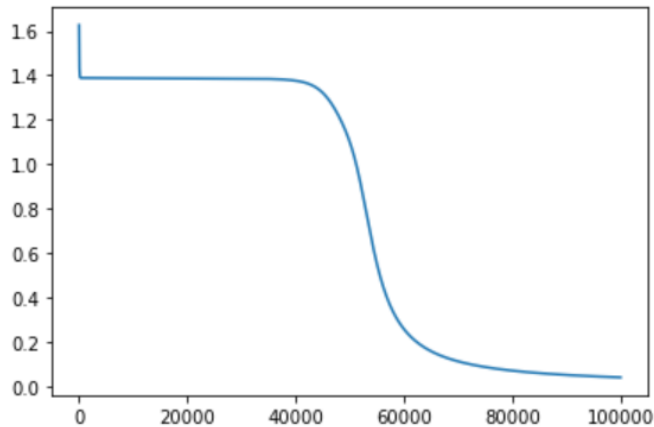
W2: [[-5.37202644 5.68449807] [-4.31881254 4.01731447]]  
W1: [[-5.50695896 5.79931703] [-4.55492152 4.24955203]]  
W2: [[-5.50695896 5.79931703] [-4.55492152 4.24955203]]  
W1: [[-5.69449828 5.96437057] [-4.8684583 4.55939013]]  
W2: [[-5.69449828 5.96437057] [-4.8684583 4.55939013]]  
W1: [[-5.8253247 6.08321205] [-5.07813049 4.76735909]]  
W2: [[-5.8253247 6.08321205] [-5.07813049 4.76735909]]  
W1: [[-5.97680976 6.22435713] [-5.31261072 5.00057716]]  
W2: [[-5.97680976 6.22435713] [-5.31261072 5.00057716]]  
W1: [[-6.07051974 6.31340236] [-5.45358911 5.14109658]]  
W2: [[-6.07051974 6.31340236] [-5.45358911 5.14109658]]  
W1: [[-6.13862845 6.37887267] [-5.55424238 5.24154996]]  
W2: [[-6.13862845 6.37887267] [-5.55424238 5.24154996]]  
W1: [[-6.20069676 6.43904394] [-5.64471252 5.33192579]]  
W2: [[-6.20069676 6.43904394] [-5.64471252 5.33192579]]  
W1: [[-6.26461281 6.5014735 ] [-5.73668203 5.42387714]]  
W2: [[-6.26461281 6.5014735 ] [-5.73668203 5.42387714]]  
W1: [[-6.31726995 6.55323498] [-5.81158025 5.49881463]]  
W2: [[-6.31726995 6.55323498] [-5.81158025 5.49881463]]  
W1: [[-6.45493288 6.68979991] [-6.00389937 5.69143216]]  
W2: [[-6.45493288 6.68979991] [-6.00389937 5.69143216]]  
W1: [[-6.51345791 6.74833684] [-6.08422409 5.77195442]]  
W2: [[-6.51345791 6.74833684] [-6.08422409 5.77195442]]  
W1: [[-6.55476274 6.78980035] [-6.14042871 5.82831879]]  
W2: [[-6.55476274 6.78980035] [-6.14042871 5.82831879]]  
W1: [[-6.60548101 6.84086893] [-6.20891345 5.89701916]]  
W2: [[-6.60548101 6.84086893] [-6.20891345 5.89701916]]  
W1: [[-6.68444974 6.9206897 ] [-6.31443327 6.00290893]]  
W2: [[-6.68444974 6.9206897 ] [-6.31443327 6.00290893]]  
W1: [[-6.73588833 6.97286156] [-6.38247161 6.07120497]]  
W2: [[-6.73588833 6.97286156] [-6.38247161 6.07120497]]  
W1: [[-6.77157188 7.00912738] [-6.42936188 6.11827944]]  
W2: [[-6.77157188 7.00912738] [-6.42936188 6.11827944]]  
W1: [[-6.80246164 7.04056588] [-6.46975467 6.15883424]]  
W2: [[-6.80246164 7.04056588] [-6.46975467 6.15883424]]  
W1: [[-6.82365155 7.06215468] [-6.49735955 6.18655122]]  
W2: [[-6.82365155 7.06215468] [-6.49735955 6.18655122]]  
W1: [[-6.85743922 7.09661378] [-6.54120476 6.23057616]]

W2:  $\begin{bmatrix} -6.85743922 & 7.09661378 \\ -6.54120476 & 6.23057616 \end{bmatrix}$   
W1:  $\begin{bmatrix} -6.89692295 & 7.13693329 \\ -6.59218122 & 6.28176312 \end{bmatrix}$   
W2:  $\begin{bmatrix} -6.89692295 & 7.13693329 \\ -6.59218122 & 6.28176312 \end{bmatrix}$   
W1:  $\begin{bmatrix} -6.96788876 & 7.20952451 \\ -6.68312332 & 6.37308073 \end{bmatrix}$   
W2:  $\begin{bmatrix} -6.96788876 & 7.20952451 \\ -6.68312332 & 6.37308073 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.01538978 & 7.25819065 \\ -6.74352634 & 6.43372998 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.01538978 & 7.25819065 \\ -6.74352634 & 6.43372998 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.04412671 & 7.28765867 \\ -6.77989267 & 6.47024224 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.04412671 & 7.28765867 \\ -6.77989267 & 6.47024224 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.08794108 & 7.33262155 \\ -6.83509157 & 6.52565812 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.08794108 & 7.33262155 \\ -6.83509157 & 6.52565812 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.10998491 & 7.35525725 \\ -6.86275264 & 6.55342544 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.10998491 & 7.35525725 \\ -6.86275264 & 6.55342544 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.14769584 & 7.39400023 \\ -6.90990594 & 6.60075539 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.14769584 & 7.39400023 \\ -6.90990594 & 6.60075539 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.16449778 & 7.41126926 \\ -6.9308482 & 6.62177409 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.16449778 & 7.41126926 \\ -6.9308482 & 6.62177409 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.19296341 & 7.44053552 \\ -6.96623639 & 6.65728837 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.19296341 & 7.44053552 \\ -6.96623639 & 6.65728837 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.22151622 & 7.46990211 \\ -7.00161891 & 6.69279277 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.22151622 & 7.46990211 \\ -7.00161891 & 6.69279277 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.24871692 & 7.49788702 \\ -7.03522183 & 6.72650724 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.24871692 & 7.49788702 \\ -7.03522183 & 6.72650724 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.26060802 & 7.51012343 \\ -7.04988035 & 6.74121307 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.26061637 & 7.51013203 \\ -7.04989064 & 6.74122339 \end{bmatrix}$   
W1:  $\begin{bmatrix} -7.26064143 & 7.51015781 \\ -7.0499215 & 6.74125435 \end{bmatrix}$   
W2:  $\begin{bmatrix} -7.26064143 & 7.51015781 \\ -7.0499215 & 6.74125435 \end{bmatrix}$

## 二、繪出上題中權值矩陣 W1 所對應的切割直線



### 三、列出 MSE 的相對變化



### 四、變化過程如何呈現以及用幾個階段呈現，請各別發揮

(1) 定義 Sigmoid 函數

```
def sigmoid(z):  
    return 1 / (1 + np.exp(-z))
```

(2) 初始化神經網絡參數（權重、偏差）並定義模型超參數（迭代次數、學習率）

```
def initializeParameters(inputFeatures, neuronsInHiddenLayers, outputFeatures):  
    W1 = np.random.randn(neuronsInHiddenLayers, inputFeatures)  
    W2 = np.random.randn(outputFeatures, neuronsInHiddenLayers)  
    b1 = np.zeros((neuronsInHiddenLayers, 1))  
    b2 = np.zeros((outputFeatures, 1))  
  
    parameters = {"W1": W1, "b1": b1,  
                  "W2": W2, "b2": b2}  
    #print("initialize W1:", W1)  
    #print("initialize W2:", W2)  
    return parameters
```

(3) 前向傳播

```
def forwardPropagation(X, Y, parameters):  
    m = X.shape[1]  
    W1 = parameters["W1"]  
    W2 = parameters["W2"]  
    b1 = parameters["b1"]  
    b2 = parameters["b2"]  
  
    Z1 = np.dot(W1, X) + b1  
    A1 = sigmoid(Z1)  
    Z2 = np.dot(W2, A1) + b2  
    A2 = sigmoid(Z2)  
  
    cache = (Z1, A1, W1, b1, Z2, A2, W2, b2)  
    logprobs = np.multiply(np.log(A2), Y) + np.multiply(np.log(1 - A2), (1 - Y))  
    cost = -np.sum(logprobs) / m  
    #print(W1)  
    return cost, cache, A2
```

#### (4)反向傳播

```
def backwardPropagation(X, Y, cache):
    m = X.shape[1]
    (Z1, A1, W1, b1, Z2, A2, W2, b2) = cache

    dZ2 = A2 - Y
    dW2 = np.dot(dZ2, A1.T) / m
    db2 = np.sum(dZ2, axis = 1, keepdims = True)

    dA1 = np.dot(W2.T, dZ2)
    dZ1 = np.multiply(dA1, A1 * (1 - A1))
    dW1 = np.dot(dZ1, X.T) / m
    db1 = np.sum(dZ1, axis = 1, keepdims = True) / m

    gradients = {"dZ2": dZ2, "dW2": dW2, "db2": db2,
                  "dZ1": dZ1, "dW1": dW1, "db1": db1}
    return gradients
```

#### (5)更新權重

```
def updateParameters(parameters, gradients, learningRate):
    parameters["W1"] = parameters["W1"] - learningRate * gradients["dW1"]
    parameters["W2"] = parameters["W2"] - learningRate * gradients["dW2"]
    parameters["b1"] = parameters["b1"] - learningRate * gradients["db1"]
    parameters["b2"] = parameters["b2"] - learningRate * gradients["db2"]
    #print("W1:", W1)
    #print("W2:", W2)
    return parameters
```

#### (6)訓練學習模型

```
# Model to Learn the XOR truth table
X = np.array([[0, 0, 1, 1], [0, 1, 0, 1]]) # XOR input
Y = np.array([[0, 1, 1, 0], [1, 0, 0, 1]]) # XOR output

# Define model parameters
neuronsInHiddenLayers = 2
inputFeatures = X.shape[0]
outputFeatures = Y.shape[0]
parameters = initializeParameters(inputFeatures, neuronsInHiddenLayers, outputFeatures)
epoch = 100000
learningRate = 0.01
losses = np.zeros((epoch, 1))
#print("initialize W1:", W1)
#print("initialize W2:", W2)

for i in range(epoch):
    losses[i, 0], cache, A2 = forwardPropagation(X, Y, parameters)
    gradients = backwardPropagation(X, Y, cache)
    parameters = updateParameters(parameters, gradients, learningRate)
```

#### (7)繪製 MSE

```
plt.figure()
plt.plot(losses)
plt.xlabel("EPOCHS")
plt.ylabel("Loss value")
plt.show()
```

#### (8)測試模型性能

```
X = np.array([[1, 1, 0, 0], [0, 1, 0, 1]]) # XOR input
cost, _, A2 = forwardPropagation(X, Y, parameters)
prediction = (A2 > 0.5) * 1.0
# print(A2)
print(prediction)
```