Report of Homework 0

姓名: 吳耿暉

學號: 0556171

## 1. Introduction:

首先定義notation，$a^0$為input layer，$W^l$為每層的weight，$b^l$為每層的bias，$z^l$為每層的output，而$a^l$則是每層$z^l$經過activation function後的output，即為下一層的input，$C^r$為error，此notation參考自台大李宏毅的講義[1]，並且實作也基於其上完成。

feedforward pass:

$$z^1 = W^1 a^0 + b^1$$

$$z^l = W^l a^{l-1} + b^l,\ a^l = \sigma(z^l)$$

$$a^l = \sigma(W^l a^{l-1} + b^l)$$

backward pass:

$$\frac{\partial C^r}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l}\frac{\partial C^r}{\partial z_i^l} = a_j^{l-1}\delta_i^l$$

$$\frac{\partial C^r}{\partial b_i^l} = \frac{\partial z_i^l}{\partial b_i^l}\frac{\partial C^r}{\partial z_i^l} = \delta_i^l$$

1. output layer

$$\delta^L = \sigma(z^L) \cdot \nabla C^r(y^r)$$

2. other layers

$$\delta^l = \sigma'(z^l) \cdot (W^{l+1})^T \delta^{l+1}$$

## 2. Experiment setups:

### A. sigmoid functions:

```
def sigmoid(self, x, derivative=False):
    # you already input the sigmoid function, so use x rather than sigmoid(x)
    if derivative:
        return x * (1-x)
    return np.exp(-np.logaddexp(0, -x))
    return 1 / (np.exp(-x))
```

### B. Neural network:(Bonus done)

實作n-layer NN，詳情見code，error function採用SE

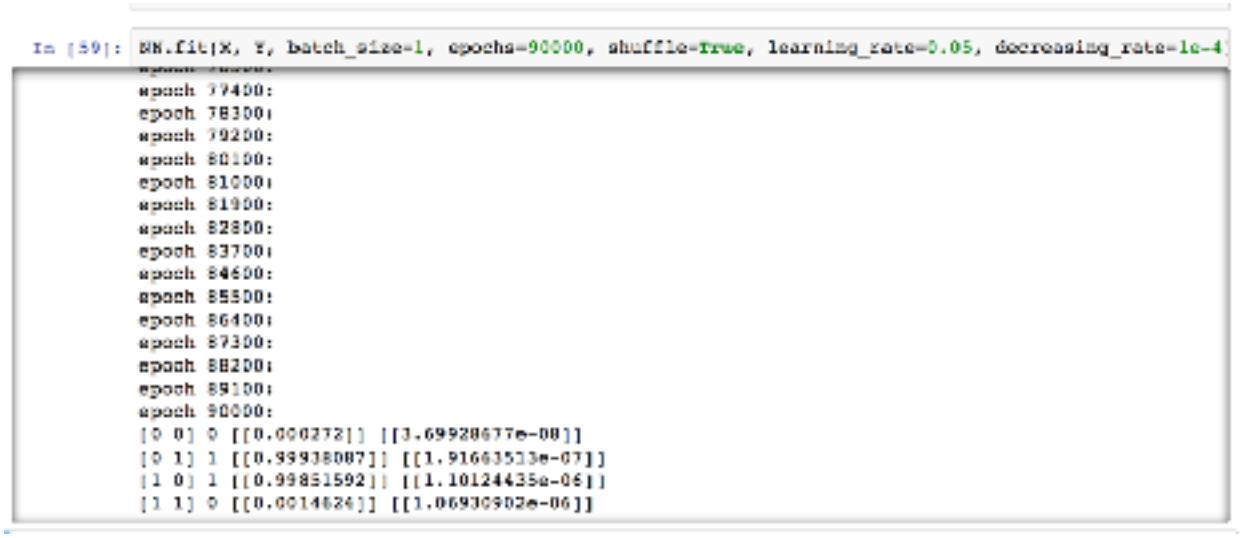
### C. Backpropagation：

如上面的notation定義中的backward pass，實作見code

# 3. Experiment setups:

## A. Results of your testing

### (1) Screenshot

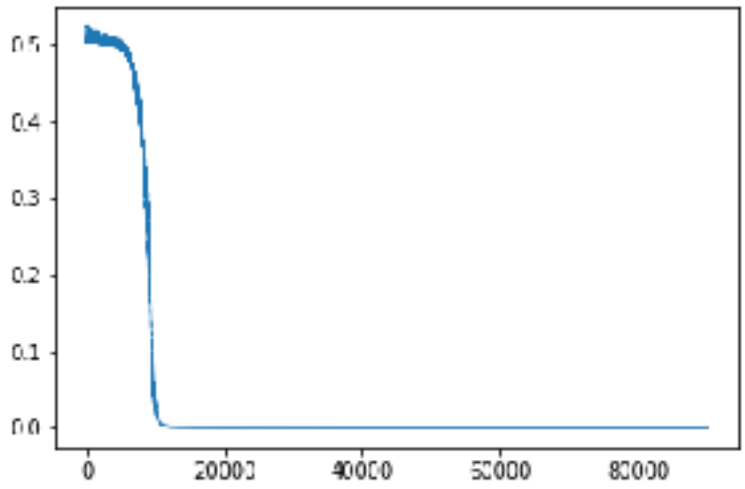



# 4. Discussion:

(1)實作上在sigmoid的部分由於有exponential項的關係，因此有可能造成數值問題[2]，所以選擇了此種實作。

(2)在此例中應用decreasing rate和shuffle似乎並沒有明顯的進一步改善。

(3)收斂過程如圖，x軸為epoch數，y軸為error




## 5. Reference:

[1] http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/DNN%20backprop.pdf

[2] https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.logaddexp.html