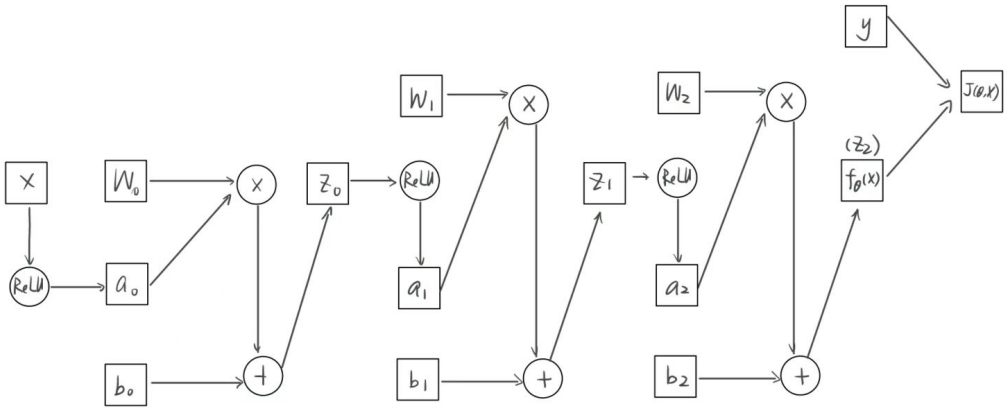
Pattern Recognition and Machine Learning: Homework 2

Qingru Hu 2020012996

April 30, 2023

Problem 1

The computing graph of the loss is as below.



Problem 2

For the 2th layer:

$$\frac{\partial J}{\partial y} = -(z_2 - y)$$
$$\frac{\partial J}{\partial z_2} = z_2 - y$$
$$\frac{\partial J}{\partial a_2} = \frac{\partial z_2}{\partial a_2}\frac{\partial J}{\partial z_2} = W_2\frac{\partial J}{\partial z_2}$$
$$\frac{\partial J}{\partial W_2} = \frac{\partial z_2}{\partial W_2}\frac{\partial J}{\partial z_2} = a_2\frac{\partial J}{\partial z_2}$$
$$\frac{\partial J}{\partial b_2} = \frac{\partial z_2}{\partial b_2}\frac{\partial J}{\partial z_2} = \frac{\partial J}{\partial z_2}$$

For the 1th layer:

$$\frac{\partial J}{\partial z_1} = \frac{\partial a_2}{\partial z_1}\frac{\partial J}{\partial a_2} = g'(z_1)\frac{\partial J}{\partial a_2} = H(z_1)\frac{\partial J}{\partial a_2}$$
$$\frac{\partial J}{\partial a_1} = \frac{\partial z_1}{\partial a_1}\frac{\partial J}{\partial z_1} = W_1\frac{\partial J}{\partial z_1}$$
$$\frac{\partial J}{\partial W_1} = \frac{\partial z_1}{\partial W_1}\frac{\partial J}{\partial z_1} = a_1\frac{\partial J}{\partial z_1}$$
$$\frac{\partial J}{\partial b_1} = \frac{\partial z_1}{\partial b_1}\frac{\partial J}{\partial z_1} = \frac{\partial J}{\partial z_1}$$

For the 0th layer:

$$\frac{\partial J}{\partial z_0} = \frac{\partial a_1}{\partial z_0}\frac{\partial J}{\partial a_1} = g'(z_0)\frac{\partial J}{\partial a_1} = H(z_0)\frac{\partial J}{\partial a_1}$$
$$\frac{\partial J}{\partial a_0} = \frac{\partial z_0}{\partial a_0}\frac{\partial J}{\partial z_0} = W_0\frac{\partial J}{\partial z_0}$$
$$\frac{\partial J}{\partial W_0} = \frac{\partial z_0}{\partial W_0}\frac{\partial J}{\partial z_0} = a_0\frac{\partial J}{\partial z_0}$$
$$\frac{\partial J}{\partial b_0} = \frac{\partial z_0}{\partial b_0}\frac{\partial J}{\partial z_0} = \frac{\partial J}{\partial z_0}$$
$$\frac{\partial J}{\partial x} = \frac{\partial a_0}{\partial x}\frac{\partial J}{\partial a_0} = g'(x)\frac{\partial J}{\partial a_0} = H(x)\frac{\partial J}{\partial a_0}$$

Problem 3

```
# Define the function for students to implement back-propagation
def compute_gradient(x, y, W0, W1, W2, b0, b1, b2, a1, a2):
    a0 = relu(x)

    z2 = np.dot(a2, W2) + b2
    pz2 = (z2 - y) / x.shape[0]
    pa2 = pz2.dot(W2.T)
    pW2 = a2.T.dot(pz2)
```

Problem 3 continued on next page...

```
    pb2 = np.sum(pz2)

    z1 = np.dot(a1, W1) + b1
    pz1 = relu_derivative(z1)*pa2
    pa1 = pz1.dot(W1.T)
    pW1 = a1.T.dot(pz1)
    pb1 = np.sum(pz1)

    z0 = np.dot(a0, W0) + b0
    pz0 = relu_derivative(z0)*(pa1)
    pW0 = a0.T.dot(pz0)
    pb0 = np.sum(pz0)

    return [pW0,pW1,pW2,pb0,pb1,pb2]
```

The difference with the gradient computed from definition is shown as below.

```
(prml) D:\github\Pattern-Recognition-and-Machine-Learning\hw9\hw9-QingruHu>python train.py
W0 diff 0.243165836705046 77
W1 diff 0.0009273427677967128
W2 diff 0.0035500176939433636
b0 diff 2.7418872085385297
b1 diff 4.1414421584969 14
b2 diff 1.2239986801887426e-11
Please make sure all the difference are sufficiently small to go on
```

The test loss is 0.118030965884446.

```
  0: loss is 7.166226621227566
1: loss is 5.436435240565046
2: loss is 4.362485487730143
3: loss is 3.5259562511192373
4: loss is 2.797636227865685
5: loss is 2.1391386688220955
6: loss is 1.554095074537603
7: loss is 1.0651384782714837
8: loss is 0.6967674140650253
9: loss is 0.4542093618541994
10: loss is 0.3156643620586208
...
90: loss is 0.09478981090562362
91: loss is 0.09442840347448693
92: loss is 0.09407267731409347
93: loss is 0.0937219001761906
94: loss is 0.09337641133499174
95: loss is 0.0930358066748061
96: loss is 0.09269980685026996
97: loss is 0.0923692779251972
98: loss is 0.0920438382813709
99: loss is 0.09172332782835861
Test loss is 0.118030965884446
```