Homework 11/14 (4-4 Backpropagation)

4B315021 詹家緯

main.py

```
import numpy as np

def backpropagation(input, target, bias, weight):
    print("\n#### 1. Forward Pass:")
    print("\n##### A. Hidden Unit")
    print("\n###### i. net")
    net = [0] * 10
    a = [0] * 10
    for i in range(len(input)):
        a[i] = input[i]
    for i in range(2, 5 + 1):  # 2, 3, 4, 5 : hidden num
        for j in range(0, 1 + 1):  # 0, 1 : input num
            # net_i = a_0*a_0i + a_1*a_1i + b_i
            net[i] += input[j] * weight[j][i]
        net[i] += bias[i]
        # Round the result to six decimal places
```

```
        # net[i] = round(net[i], 6)
        print("net({}) = {:.6f}".format(i, net[i]))
    print("\n###### ii. a")
    for i in range(2, 5 + 1):  # 2, 3, 4, 5 : hidden num
        a[i] = (1 + np.exp(-1 * net[i])) ** -1
        # Round the result to six decimal places
        # a[i] = round(a[i], 6)
        print("a({}) = {:.6f}".format(i, a[i]))
    print("\n##### B. Output Unit")
    print("\n###### i. net")
    for i in range(6, 9 + 1):  # 6, 7, 8, 9 : output num
        for j in range(2, 5 + 1):  # 2, 3, 4, 5 : hidden num
            # net_i = a_2*a_2i + a_3*a_3i + a_4*a_4i + a_5*a_5i + b_i
            net[i] += a[j] * weight[j][i]
        net[i] += bias[i]
        # Round the result to six decimal places
        # net[i] = round(net[i], 6)
        print("net({}) = {:.6f}".format(i, net[i]))
    print("\n###### ii. a")
    for i in range(6, 9 + 1):  # 6, 7, 8, 9 : output num
        a[i] = (1 + np.exp(-1 * net[i])) ** -1
        # Round the result to six decimal places
        # a[i] = round(a[i], 6)
        print("a({}) = {:.6f}".format(i, a[i]))

    print("\n#### 2. Backward Pass:")
    delta = [0] * 10
    print("\n##### A. Output Unit")
    for i in range(6, 9 + 1):  # 6, 7, 8, 9 : output num
        delta[i] = (target[i - 6] - a[i]) * ( a[i] * (1 - a[i] ))  # 1-6 = 0, 1, 2,
3 : target num
        # Round the result to six decimal places
        # delta[i] = round(delta[i], 6)
        print("d({}) = {:.6f}".format(i, delta[i]))
    print("\n##### B. Hidden Unit")
    for i in range(2, 5 + 1):  # 2, 3, 4, 5 : hidden num
        for j in range(6, 9 + 1):  # 6, 7, 8, 9 : output num
            delta[i] += delta[j] * weight[i][j]
        delta[i] *= a[i] * (1 - a[i])
        # Round the result to six decimal places
        # delta[i] = round(delta[i], 6)
        print("d({}) = {:.6f}".format(i, delta[i]))

    print("\n#### 3. Change of Weights and Biases:")
    print("\n##### A. Weights")
    d_weight = [[0] * 10] * 6
    weight_new = [[0] * 10] * 6
    for i in range(0, 1+1): # 0, 1 : input num
        for j in range(2, 5+1): # 2, 3, 4, 5 : hidden num
            d_weight[i][j] = 0.20 * delta[j] * a[i]
            weight_new[i][j] = weight[i][j] + d_weight[i][j]
            # Round the result to six decimal places
            # d_weight[i][j] = round(d_weight[i][j], 6)
            print("d_w({}-{}) = {:.6f}, w_new({}-{}) = {:.6f}".format(i, j,
d_weight[i][j], i, j, weight_new[i][j]))
    for i in range(2, 5+1): # 2, 3, 4, 5 : hidden num
        for j in range(6, 9+1): # 6, 7, 8, 9 : output num
```

```
            d_weight[i][j] = 0.20 * delta[j] * a[i]
            weight_new[i][j] = weight[i][j] + d_weight[i][j]
            # Round the result to six decimal places
            # d_weight[i][j] = round(d_weight[i][j], 6)
            print("d_w({}-{}) = {:.6f}, w_new({}-{}) = {:.6f}".format(i, j,
d_weight[i][j], i, j, weight_new[i][j]))
    print("\n##### B. Biases")
    d_bias = [0] * 10
    bias_new = [0] * 10
    for i in range(2, 9+1): # 2 , ..., 9 : biases num
        d_bias[i] = 0.20 * delta[i]
        bias_new[i] = bias[i] + d_bias[i]
        # Round the result to six decimal places
        # d_bias[i] = round(d_bias[i], 6)
        print("d_b({}) = {:.6f}, b_new({}) = {:.6f}".format(i, d_bias[i], i,
bias_new[i]))
    return 0

if __name__ == "__main__":
    input = (0.017322, 1.480488)
    target = (0.494200, 0.495051, 0.494171, 0.501720)
    # b_2 = bias[2] etc...
    bias = (0, 0, -0.444700, 0.410733, 0.358089, -0.005783, 0.094012, -0.058550,
-0.055376, -0.158925)
    # w_0-2 = weight[0][2] etc...
    weight = [
        [0, 0, 0.121845, 0.474700, 0.194113, 0.318567, 0, 0, 0, 0],
        [0, 0, -0.384945, 0.131458, 0.187948, 0.117237, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0.069170, 0.326563, -0.106006, -0.189261],
        [0, 0, 0, 0, 0, 0, -0.088916, 0.360866, 0.264275, -0.165883],
        [0, 0, 0, 0, 0, 0, -0.432951, 0.046312, -0.455687, -0.068896],
        [0, 0, 0, 0, 0, 0, -0.270775, 0.323694, 0.128620, -0.490692],
    ]
    backpropagation(input, target, bias, weight)
```

Output

1. Forward Pass:

A. Hidden Unit

i. net

net(2) = -1.012496 net(3) = 0.613578 net(4) = 0.639706 net(5) = 0.173303

ii. a

a(2) = 0.266492 a(3) = 0.648757 a(4) = 0.654687 a(5) = 0.543218

B. Output Unit

i. net

net(6) = -0.375777 net(7) = 0.468747 net(8) = -0.140639 net(9) = -0.628637

ii. a

a(6) = 0.407146 a(7) = 0.615087 a(8) = 0.464898 a(9) = 0.347820

2. Backward Pass:

A. Output Unit

d(6) = 0.021013 d(7) = -0.028419 d(8) = 0.007282 d(9) = 0.034911

B. Hidden Unit

d(2) = -0.002972 d(3) = -0.003644 d(4) = -0.003648 d(5) = -0.007713

3. Change of Weights and Biases:

A. Weights

d_w(0-2) = -0.000010, w_new(0-2) = 0.121835 d_w(0-3) = -0.000013, w_new(0-3) = 0.474687 d_w(0-4) = -0.000013, w_new(0-4) = 0.194100 d_w(0-5) = -0.000027, w_new(0-5) = 0.318540 d_w(1-2) = -0.000880, w_new(1-2) = -0.385825 d_w(1-3) = -0.001079, w_new(1-3) = 0.130379 d_w(1-4) = -0.001080, w_new(1-4) = 0.186868 d_w(1-5) = -0.002284, w_new(1-5) = 0.114953 d_w(2-6) = 0.001120, w_new(2-6) = 0.070290 d_w(2-7) = -0.001515, w_new(2-7) = 0.325048 d_w(2-8) = 0.000388, w_new(2-8) = -0.105618 d_w(2-9) = 0.001861, w_new(2-9) = -0.187400 d_w(3-6) = 0.002726, w_new(3-6) = -0.086190 d_w(3-7) = -0.003687, w_new(3-7) = 0.357179 d_w(3-8) = 0.000945, w_new(3-8) = 0.265220 d_w(3-9) = 0.004530, w_new(3-9) = -0.161353 d_w(4-6) = 0.002751, w_new(4-6) = -0.430200 d_w(4-7) = -0.003721, w_new(4-7) = 0.042591 d_w(4-8) = 0.000954, w_new(4-8) = -0.454733 d_w(4-9) = 0.004571, w_new(4-9) = -0.064325 d_w(5-6) = 0.002283, w_new(5-6) = -0.268492 d_w(5-7) = -0.003088, w_new(5-7) = 0.320606 d_w(5-8) = 0.000791, w_new(5-8) = 0.129411 d_w(5-9) = 0.003793, w_new(5-9) = -0.486899

B. Biases

d_b(2) = -0.000594, b_new(2) = -0.445294 d_b(3) = -0.000729, b_new(3) = 0.410004
d_b(4) = -0.000730, b_new(4) = 0.357359 d_b(5) = -0.001543, b_new(5) = -0.007326
d_b(6) = 0.004203, b_new(6) = 0.098215 d_b(7) = -0.005684, b_new(7) = -0.064234
d_b(8) = 0.001456, b_new(8) = -0.053920 d_b(9) = 0.006982, b_new(9) = -0.151943