Machine Learning Homework 1

Liao-Jia-Wei, 309652008

March 21, 2022

Problem.

Given the 2-layer feedforward network in Figure 1. and an output. Compute the following value.

1. outputs of the model
2. loss value
3. gradient of all the parameters
4. all the parameters after updating by gradient descent with a 0.5 learning rate

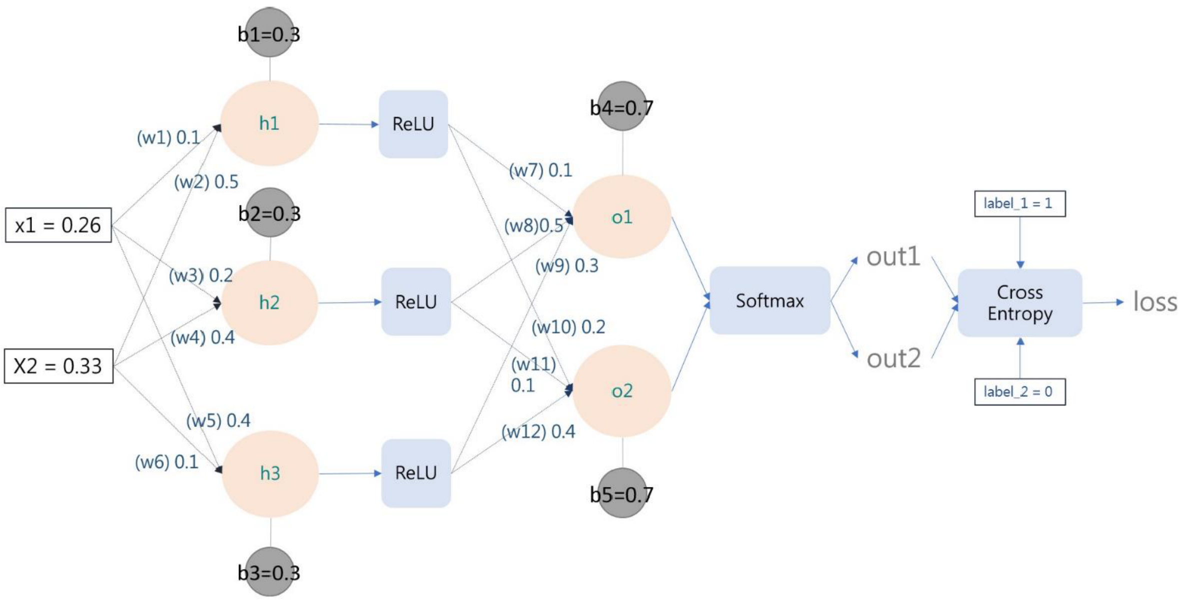

Figure 1. 2-layer feedforward network

Solution: Let $y_1 = \text{out}_1, y_2 = \text{out}_2, t_1 = \text{label}_1, t_2 = \text{label}_2$

- Forward:

$$\begin{aligned}
h_1 &= w_1x_1 + w_2x_2 + b_1 \\
h_2 &= w_3x_1 + w_4x_2 + b_2 \\
h_3 &= w_5x_1 + w_6x_2 + b_3 \\
a_1 &= \mathrm{ReLU}(h_1) \\
a_2 &= \mathrm{ReLU}(h_2) \\
a_3 &= \mathrm{ReLU}(h_3) \\
o_1 &= w_7a_1 + w_8a_2 + w_9a_3 + b_4 \\
o_2 &= w_{10}a_1 + w_{11}a_2 + w_{12}a_3 + b_5 \\
[y_1, y_2] &= \mathrm{Softmax}(o_1, o_2) \\
\mathcal{L}(y, t) &= -t_1 \log_e y_1 - t_2 \log_e y_2
\end{aligned}$$

- Backward:

  Since $\log_e x = \frac{1}{x}$,

$$\frac{\partial \mathcal{L}}{\partial y_1} = -\frac{t_1}{y_1}, \quad \frac{\partial \mathcal{L}}{\partial y_2} = -\frac{t_2}{y_2}$$

  Notice that

$$\begin{aligned}
\frac{\partial}{\partial x_1}\left(\frac{e^{x_1}}{e^{x_1}+e^{x_2}}\right) &= \frac{e^{x_1}(e^{x_1}+e^{x_2}) - (e^{x_1})^2}{(e^{x_1}+e^{x_2})^2} = \left(\frac{e^{x_1}}{e^{x_1}+e^{x_2}}\right)\left(1 - \frac{e^{x_1}}{e^{x_1}+e^{x_2}}\right) \\
\frac{\partial}{\partial x_2}\left(\frac{e^{x_1}}{e^{x_1}+e^{x_2}}\right) &= \frac{-e^{x_1}e^{x_2}}{(e^{x_1}+e^{x_2})^2} = -\left(\frac{e^{x_1}}{e^{x_1}+e^{x_2}}\right)\left(\frac{e^{x_2}}{e^{x_1}+e^{x_2}}\right)
\end{aligned}$$

  Hence

$$\frac{\partial y_1}{\partial o_1} = y_1(1-y_1), \quad \frac{\partial y_1}{\partial o_2} = -y_1y_2, \quad \frac{\partial y_2}{\partial o_1} = -y_1y_2, \quad \frac{\partial y_2}{\partial o_2} = y_2(1-y_2)$$

  By Chain rule,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial o_1} &= \frac{\partial \mathcal{L}}{\partial y_1}\frac{\partial y_1}{\partial o_1} + \frac{\partial \mathcal{L}}{\partial y_2}\frac{\partial y_2}{\partial o_1} = -t_1(1-y_1) + t_2y_1 \\
\frac{\partial \mathcal{L}}{\partial o_2} &= \frac{\partial \mathcal{L}}{\partial y_1}\frac{\partial y_1}{\partial o_2} + \frac{\partial \mathcal{L}}{\partial y_2}\frac{\partial y_2}{\partial o_2} = t_1y_2 - t_2(1-y_2)
\end{aligned}$$

  Since $\frac{\partial o_1}{\partial w_7} = a_1$, $\frac{\partial o_1}{\partial w_8} = a_2$, $\frac{\partial o_1}{\partial w_9} = a_3$, and $\frac{\partial o_1}{\partial b_4} = 1$,

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_7} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial w_7} = [-t_1(1-y_1)+t_2y_1]a_1 \\
\frac{\partial \mathcal{L}}{\partial w_8} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial w_8} = [-t_1(1-y_1)+t_2y_1]a_2 \\
\frac{\partial \mathcal{L}}{\partial w_9} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial w_9} = [-t_1(1-y_1)+t_2y_1]a_3 \\
\frac{\partial \mathcal{L}}{\partial b_4} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial b_4} = [-t_1(1-y_1)+t_2y_1]
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{10}} &= \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial w_{10}} = [t_1y_2-t_2(1-y_2)]a_1 \\
\frac{\partial \mathcal{L}}{\partial w_{11}} &= \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial w_{11}} = [t_1y_2-t_2(1-y_2)]a_2 \\
\frac{\partial \mathcal{L}}{\partial w_{12}} &= \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial w_{12}} = [t_1y_2-t_2(1-y_2)]a_3 \\
\frac{\partial \mathcal{L}}{\partial b_5} &= \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial b_5} = [t_1y_2-t_2(1-y_2)]
\end{aligned}
$$

By Chain rule again,

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial a_1} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial a_1} + \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial a_1} = [-t_1(1-y_1)+t_2y_1]w_7 + [t_1y_2-t_2(1-y_2)]w_{10} \\
\frac{\partial \mathcal{L}}{\partial a_2} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial a_2} + \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial a_2} = [-t_1(1-y_1)+t_2y_1]w_8 + [t_1y_2-t_2(1-y_2)]w_{11} \\
\frac{\partial \mathcal{L}}{\partial a_3} &= \frac{\partial \mathcal{L}}{\partial o_1}\frac{\partial o_1}{\partial a_3} + \frac{\partial \mathcal{L}}{\partial o_2}\frac{\partial o_2}{\partial a_3} = [-t_1(1-y_1)+t_2y_1]w_9 + [t_1y_2-t_2(1-y_2)]w_{12}
\end{aligned}
$$

Notice that

$$
\frac{\partial a_1}{\partial h_1} = u(h_1), \quad \frac{\partial a_2}{\partial h_2} = u(h_2), \quad \frac{\partial a_3}{\partial h_3} = u(h_3)
$$

where $u(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \le 0 \end{cases}$ be a step function.

Finally, we have

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_1} &= \frac{\partial \mathcal{L}}{\partial a_1}\frac{\partial a_1}{\partial h_1}\frac{\partial h_1}{\partial w_1} = \{[-t_1(1-y_1)+t_2y_1]w_7 + [t_1y_2-t_2(1-y_2)]w_{10}\}u(h_1)x_1 \\
\frac{\partial \mathcal{L}}{\partial w_2} &= \frac{\partial \mathcal{L}}{\partial a_1}\frac{\partial a_1}{\partial h_1}\frac{\partial h_1}{\partial w_2} = \{[-t_1(1-y_1)+t_2y_1]w_7 + [t_1y_2-t_2(1-y_2)]w_{10}\}u(h_1)x_2 \\
\frac{\partial \mathcal{L}}{\partial b_1} &= \frac{\partial \mathcal{L}}{\partial a_1}\frac{\partial a_1}{\partial h_1}\frac{\partial h_1}{\partial b_1} = \{[-t_1(1-y_1)+t_2y_1]w_7 + [t_1y_2-t_2(1-y_2)]w_{10}\}u(h_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_3} &= \frac{\partial \mathcal{L}}{\partial a_2}\frac{\partial a_2}{\partial h_2}\frac{\partial h_2}{\partial w_3} = \{[-t_1(1-y_1)+t_2y_1]w_8 + [t_1y_2 - t_2(1-y_2)]w_{11}\}u(h_2)x_1 \\
\frac{\partial \mathcal{L}}{\partial w_4} &= \frac{\partial \mathcal{L}}{\partial a_2}\frac{\partial a_2}{\partial h_2}\frac{\partial h_2}{\partial w_4} = \{[-t_1(1-y_1)+t_2y_1]w_8 + [t_1y_2 - t_2(1-y_2)]w_{11}\}u(h_2)x_2 \\
\frac{\partial \mathcal{L}}{\partial b_2} &= \frac{\partial \mathcal{L}}{\partial a_2}\frac{\partial a_2}{\partial h_2}\frac{\partial h_2}{\partial b_2} = \{[-t_1(1-y_1)+t_2y_1]w_8 + [t_1y_2 - t_2(1-y_2)]w_{11}\}u(h_2)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_5} &= \frac{\partial \mathcal{L}}{\partial a_3}\frac{\partial a_3}{\partial h_3}\frac{\partial h_3}{\partial w_5} = \{[-t_1(1-y_1)+t_2y_1]w_9 + [t_1y_2 - t_2(1-y_2)]w_{12}\}u(h_3)x_1 \\
\frac{\partial \mathcal{L}}{\partial w_6} &= \frac{\partial \mathcal{L}}{\partial a_3}\frac{\partial a_3}{\partial h_3}\frac{\partial h_3}{\partial w_6} = \{[-t_1(1-y_1)+t_2y_1]w_9 + [t_1y_2 - t_2(1-y_2)]w_{12}\}u(h_3)x_2 \\
\frac{\partial \mathcal{L}}{\partial b_3} &= \frac{\partial \mathcal{L}}{\partial a_3}\frac{\partial a_3}{\partial h_3}\frac{\partial h_3}{\partial b_3} = \{[-t_1(1-y_1)+t_2y_1]w_9 + [t_1y_2 - t_2(1-y_2)]w_{12}\}u(h_3)
\end{aligned}
$$

- Update parameter by gradient descent:

  Given learning rate $\gamma > 0$. For $\theta \in \{w_1, ..., w_{12}, b_1, ..., b_5\}$,

$$
\theta^{(t+1)} = \theta^{(t)} - \gamma \cdot \frac{\partial \mathcal{L}}{\partial \theta^{(t)}}
$$

Result

parameter	initial	gradient	update outputs
out1			0.53
out2			0.47
loss			0.64
w_1	0.1	0.01	0.09
w_2	0.5	0.02	0.49
w_3	0.2	-0.05	0.22
w_4	0.4	-0.06	0.43
w_5	0.4	0.01	0.39
w_6	0.1	0.02	0.09
w_7	0.1	-0.23	0.22
w_8	0.5	-0.23	0.61
w_9	0.3	-0.21	0.40
w_{10}	0.2	0.23	0.08
w_{11}	0.1	0.23	-0.01
w_{12}	0.4	0.21	0.30
b_1	0.3	0.05	0.28
b_2	0.3	-0.19	0.39
b_3	0.3	0.05	0.28
b_4	0.7	-0.47	0.94
b_5	0.7	0.47	0.46

Code implement by Python

```
import math


x1, x2 = 0.26, 0.33
t1, t2 = 1, 0
lr = 0.5

w1, w2 = 0.1, 0.5
w3, w4 = 0.2, 0.4
w5, w6 = 0.4, 0.1

w7,  w8,  w9  = 0.1, 0.5, 0.3
w10, w11, w12 = 0.2, 0.1, 0.4

b1, b2, b3 = 0.3, 0.3, 0.3
b4, b5     = 0.7, 0.7


e = math.e
Relu = lambda x: x if x>0 else 0
dRelu = lambda x: 1 if x>0 else 0
Softmax = lambda x1, x2: (e**x1 / (e**x1 + e**x2), e**x2 / (e**x1 + e**x2))
CrossEntropy = lambda x1, x2, y1, y2: -y1*math.log(x1) - y2*math.log(x2)


def GD(p, grad, lr=0.5):
    return p - lr*grad


if __name__ == '__main__':
    for i in range(1):

        ## Forward
        h1 = w1*x1 + w2*x2 + b1
        h2 = w3*x1 + w4*x2 + b2
        h3 = w5*x1 + w6*x2 + b3

        a1 = Relu(h1)
        a2 = Relu(h2)
        a3 = Relu(h3)

        o1 = w7*a1 + w8*a2 + w9*a3 + b4
        o2 = w10*a1 + w11*a2 + w12*a3 + b5

        y1, y2 = Softmax(o1, o2)
        loss = CrossEntropy(y1, y2, t1, t2)


        ## Backward
```

```
        dL_o1 = (-t1/y1) * (y1*(1-y1)) + (-t2/y2) * (-y1*y2)
        dL_o2 = (-t1/y1) * (-y1*y2) + (-t2/y2) * (y2*(1-y2))

        dL_w7 = dL_o1 * a1
        dL_w8 = dL_o1 * a2
        dL_w9 = dL_o1 * a3
        dL_b4 = dL_o1 * 1

        dL_w10 = dL_o2 * a1
        dL_w11 = dL_o2 * a2
        dL_w12 = dL_o2 * a3
        dL_b5  = dL_o2 * 1

        dL_w1 = (dL_o1 * w7 + dL_o2 * w10) * dRelu(h1) * x1
        dL_w2 = (dL_o1 * w7 + dL_o2 * w10) * dRelu(h1) * x2
        dL_b1 = (dL_o1 * w7 + dL_o2 * w10) * dRelu(h1) * 1

        dL_w3 = (dL_o1 * w8 + dL_o2 * w11) * dRelu(h2) * x1
        dL_w4 = (dL_o1 * w8 + dL_o2 * w11) * dRelu(h2) * x2
        dL_b2 = (dL_o1 * w8 + dL_o2 * w11) * dRelu(h2) * 1

        dL_w5 = (dL_o1 * w9 + dL_o2 * w12) * dRelu(h3) * x1
        dL_w6 = (dL_o1 * w9 + dL_o2 * w12) * dRelu(h3) * x2
        dL_b3 = (dL_o1 * w9 + dL_o2 * w12) * dRelu(h3) * 1

        w1  = GD(w1, dL_w1)
        w2  = GD(w2, dL_w2)
        w3  = GD(w3, dL_w3)
        w4  = GD(w4, dL_w4)
        w5  = GD(w5, dL_w5)
        w6  = GD(w6, dL_w6)
        w7  = GD(w7, dL_w7)
        w8  = GD(w8, dL_w8)
        w9  = GD(w9, dL_w9)
        w10 = GD(w10, dL_w10)
        w11 = GD(w11, dL_w11)
        w12 = GD(w12, dL_w12)
        b1  = GD(b1, dL_b1)
        b2  = GD(b2, dL_b2)
        b3  = GD(b3, dL_b3)
        b4  = GD(b4, dL_b4)
        b5  = GD(b5, dL_b5)
```