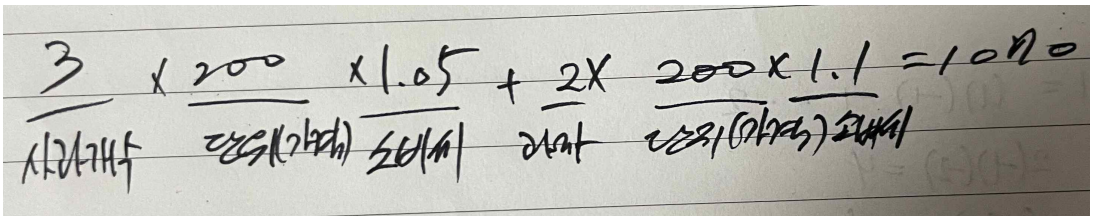
실습문제: 5장. 오차역전파법(2)

신경망

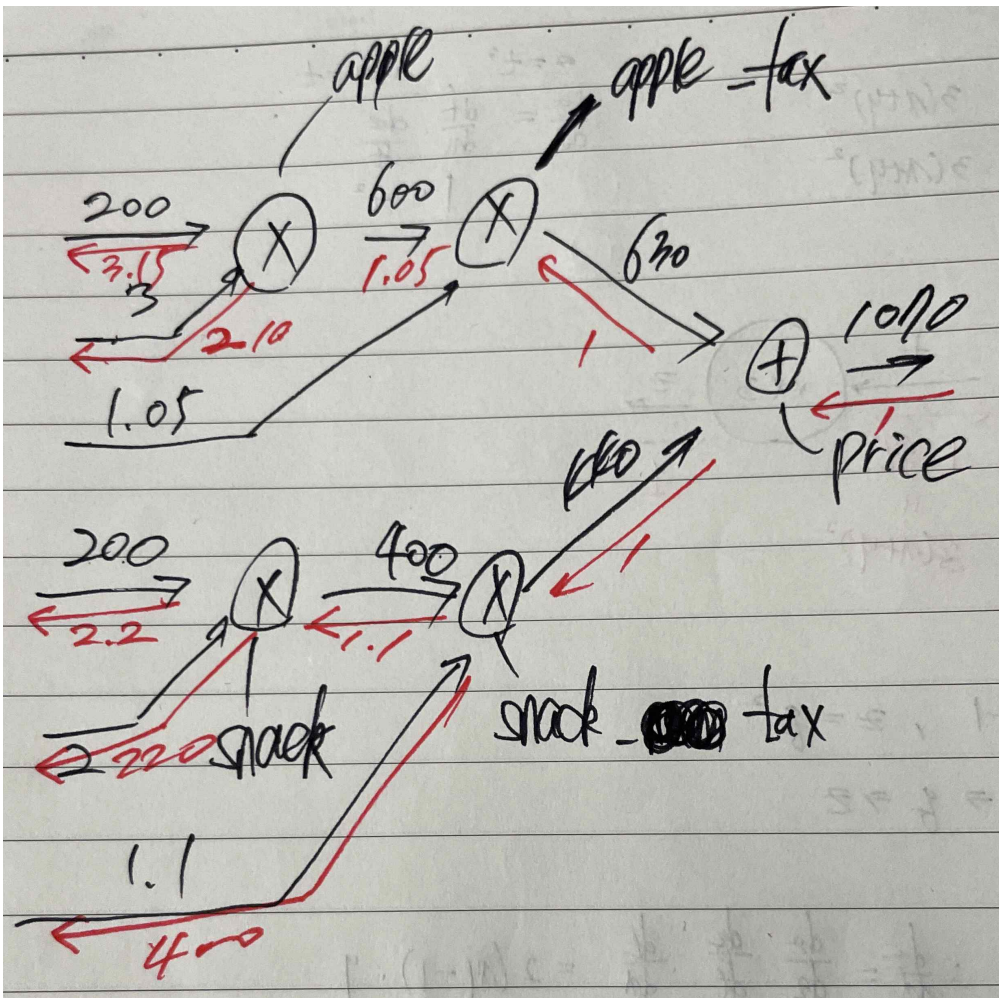
※ 3번, 8번은 .py 파일에 코드를 적고 실행하여 제출하시오. 나머지 문제는 이 hwp 파일에 답을 적어 제출하시오.

[1-3] 사과(apple) 3개, 과자(snack) 2개를 샀다. 사과는 1개에 200원이고 과자는 한 개에 200원이다. 사과의 소비세는 5%, 과자의 소비세는 10%이다. 지불금액을 구하려 한다.

1. 단순 손계산으로 지불금액을 계산하라.



2. p.163 [그림 5-17]을 참고하여 계산 그래프를 그리시오.



3. p.164 코드를 참고하여, dapple, dsnack, dapple_num, dsnack_num, dapple_tax, d_snack_tax를 구하는 프로그램을 작성하시오. 여기서 dsnack은 과자 1개 값의 변화율이다. 위 1번과 답이 같은가?

일치한다.

C: > Users > mskang > Downloads > W9 > W9_work3.py > ...

```python
apple = 200
apple_num = 3
apple_tax = 1.05

snack = 200
snack_num = 2
snack_tax = 1.1

# 순전파
apple_price = apple * apple_num
apple_tax_price = apple_price * apple_tax
snack_price = snack * snack_num
snack_tax_price = snack_price * snack_tax
price = apple_tax_price + snack_tax_price

# 역전파
dprice = 1
dapple_tax_price, dsnack_tax_price = 1 * dprice, 1 * dprice
dsnack_price, dsnack_tax = dsnack_tax_price * snack_tax, dsnack_tax_price * snack_price
dapple_price, dapple_tax = dapple_tax_price * apple_tax, dapple_tax_price * apple_price
dapple, dapple_num = dapple_price * apple_num, dapple_price * apple
dsnack, dsnack_num = dsnack_price * snack_num, dsnack_price * snack

print(dsnack)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\mskang> & C:/python/python.exe c:/Users/mskang/Downloads/W9/W9_work3.py
PS C:\Users\mskang> & C:/python/python.exe c:/Users/mskang/Downloads/W9/W9_work3.py
PS C:\Users\mskang> & C:/python/python.exe c:/Users/mskang/Downloads/W9/W9_work3.py
2.2
PS C:\Users\mskang>
```

-

4. ReLU 계층은 역전파(backward)에서 두 종류의 값을 출력한다(왼쪽으로 전달한다). 무슨 값인가?

0보다 큰 입력 : dout 그대로 전달
0 이하 입력 : 0 전달

5. Sigmoid 계층에 순전파 입력이 1 이었다.

(1) 순전파 출력은? y=1/1+e−1≈0.731

(2) 그 후, 역전파 입력이 $e+1$이었다고 할 때 역전파 출력은?

dy=(e+1)×y×(1−y) ≈(2.718 +1)×0.731×0.269≈0.731×0.269×3.718≈0.731×1.001≈0.732≈ (2.718+1)×0.731×0.269≈0.731×0.269×3.718≈0.731×1.001≈0.732

6, $W=\begin{pmatrix}1 & 0 & -1\\ 1 & 2 & 1\end{pmatrix}$, $B=(1,-1,0.5)$인 Affine 계층에, (1, 2)가 입력되었다.

(1) 이 Affine 계층의 입력되는 노드의 개수와 출력되는 노드의 개수는?

(2) 출력 값 Y는? 입력 2, 출력 3

(3) $\frac{\partial L}{\partial Y}=(1,0,-2)$일 때, (경사하강법을 위한) $\frac{\partial L}{\partial W}$와 $\frac{\partial L}{\partial B}$의 값은?

∂L/∂W = [[1,0,−2],[2,0,−4]]

∂L/∂B = [1,0,−2]∂B∂L

=[1,0,−2]

<참고> Affine 계층에서 dx 변수를 멤버 변수에 포함시키지 않았다.

7. Softmax-with-Loss 계층의 순전파 입력 값이 $(1,1,1)$이었고 입력 정답 레이블이 $(1,0,0)$이었다.

(1) Softmax-with-Loss 계층의 순전파 출력값은? y=[1/3,1/3,1/3]

(2) Softmax-with-Loss 계층의 역전파 출력값은? L=−log(1/3)=1.0986

8. 다음 네 파이썬 클래스를 정확히 타이핑하시오.

class Relu, class Sigmoid, class Affine, class SoftmaxWithLoss

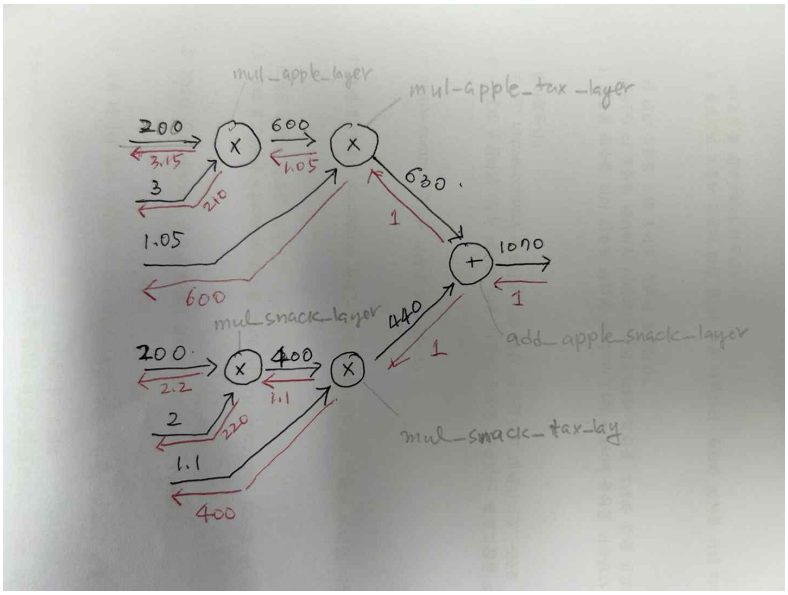
```python
import numpy as np

class Relu:
    def __init__(self):
        self.mask = None
    def forward(self, x):
        self.mask = (x <= 0)
        out = x.copy()
        out[self.mask] = 0
        return out
    def backward(self, dout):
        dout[self.mask] = 0
        dx = dout
        return dx

class Sigmoid:
    def __init__(self):
        self.out = None
    def forward(self, x):
        out = 1 / (1 + np.exp(-x))
        self.out = out
        return out
    def backward(self, dout):
        dx = dout * (1.0 - self.out) * self.out
        return dx

class Affine:
    def __init__(self, W, b):
        self.W = W
        self.b = b
        self.x = None
    def forward(self, x):
        self.x = x
        out = np.dot(x, self.W) + self.b
        return out
    def backward(self, dout):
        dx = np.dot(dout, self.W.T)
        dW = np.dot(self.x.T, dout)
        db = np.sum(dout, axis=0)
        return dx, dW, db
```

```python
class SoftmaxWithLoss:
    def __init__(self):
        self.loss = None
        self.y = None
        self.t = None
    def forward(self, x, t):
        self.t = t
        x = x - np.max(x)
        exp_x = np.exp(x)
        self.y = exp_x / np.sum(exp_x)
        self.loss = -np.sum(t * np.log(self.y + 1e-7))
        return self.loss
    def backward(self, dout=1):
        batch_size = self.t.shape[0] if self.t.ndim != 1 else 1
        dx = (self.y - self.t) / batch_size
        return dx
```

힌트

1. 3×200×1.05 + 2×200×1.1 = 1070

2.



3.

```
apple=200
apple_num = 3
apple_tax=1.05

snack=200
snack_num=2
snack_tax=1.1

#계층들
mul_apple_layer = MulLayer()
mul_apple_tax_layer = MulLayer()

mul_snack_layer = MulLayer()
mul_snack_tax_layer = MulLayer()

add_apple_snack_layer = AddLayer()



#순전파
apple_price = mul_apple_layer.forward(apple, apple_num)
apple_tax_price = mul_apple_tax_layer.forward(apple_price, apple_tax)
```

```
snack_price = mul_snack_layer.forward(snack, snack_num)
snack_tax_price = mul_snack_tax_layer.forward(snack_price, snack_tax)

price = add_apple_snack_layer.forward(apple_tax_price, snack_tax_price)

# 역전파
# 생략
```