

8강 연습문제

1. (i) 활성화 함수가 sigmoid인 이층 신경망이 dictionary

$$\{W_1 : \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, b_1 : [0, 0, 0, 0, 0], W_2 : \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, b_2 : [0, 0, 0, 0, 0]\}$$

로 주어져 있다. 데이터 $\mathbf{x} = [\log 2, 0, 0, 0, 0]$ 의 라벨이 $[0, 0, 0, 1, 0]$ 이라 하자. 손실함수는 교차 엔트로피(cross entropy)에 L^2 -규제 (L^2 -regularization)가 가해져 있다. 가중치 감소의 세기가 $\lambda = 1$ 일 때, 데이터 \mathbf{x} 의 손실함수 값을 구하시오.

- (ii) 코드로 계산하시오.

2. (i) (1번 문제 계속) 교차 엔트로피(cross entropy)값을 L_c 라 하고 L^2 -규제 (L^2 -regularization)까지 고려한 손실함수 값을 L 이라 하자. 첫번째 Affine층에 흘러들어온 교차 엔트로피의 미분이

$$\frac{\partial L_c}{\partial \mathbf{y}} = [0 \quad 1 \quad 1 \quad 1 \quad 0]$$

으로 주어져 있다. 여기서, \mathbf{y} 는 첫번째 Affine층의 출력값이다. L^2 -규제 (L^2 -regularization)까지 고려한 손실함수 L 의 첫번째 가중치 W_1 에 대한 미분

$$\frac{\partial L}{\partial W_1}$$

을 구하시오.

- (ii) 코드로 계산하시오.

3. (i) 활성화 함수가 Relu인 이층 신경망이 dictionary

$$\{W_1 : \begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}, b_1 : [0, 0, 0, 0, 0], W_2 : \begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \end{bmatrix}, b_2 : [0, 0, 0, 0, 0]\}$$

로 주어져 있다. 데이터 $\mathbf{x} = [1, 2, 3, 4, 5]$ 의 라벨이 $[1, 0, 0, 0, 0]$ 이라 하자. 손실함수는 교차 엔트로피(cross entropy)에 L^2 -규제 (L^2 -regularization)가 가해져 있다. 가중치 감소의 세기가 $\lambda = \frac{1}{5}$ 일 때, 데이터 \mathbf{x} 의 손실함수 값을 구하시오.

- (ii) 코드로 계산하시오.

4. 교차 엔트로피(cross entropy)값을 L_c 라 하고 L^2 -규제 (L^2 -regularization)까지 고려한 손실함수 값을 L 이라 하자. k 번째 Affine층의 가중치 행렬 W_k , 편향벡터 b_k 그리고 가중치 감소 계수 λ 에 대하여 등식

$$\frac{\partial L}{\partial W_k} = \frac{\partial L_c}{\partial W_k} + \lambda W_k, \quad \frac{\partial L}{\partial b_k} = \frac{\partial L_c}{\partial b_k}$$

을 설명하시오. (편의를 위해 W_k 를 2×2 행렬, b_k 를 2차원 벡터라 가정해도 좋다.)

5. (i) 다음은 Drop Out 층의 순전파와 역전파를 구현하는 코드이다.

```

class Dropout:
    def __init__(self, dropout_ratio=0.5):
        self.dropout_ratio = dropout_ratio
        self.mask = None
    def forward(self, x, train_flg=True):
        if train_flg:
            self.mask = np.random.rand(*x.shape) > self.dropout_ratio
            return x * self.mask
        else:
            return x * (1.0 - self.dropout_ratio)
    def backward(self, dout):
        return dout * self.mask

```

입력 데이터와 생성된 random 행렬이 각각

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad \begin{pmatrix} 0.4 & 0.1 & 0.5 \\ 0.5 & 0.5 & 0.6 \\ 0.2 & 0.1 & 0.9 \end{pmatrix}$$

이고 dropout ratio는 0.3이라 하자. 훈련할 때와 테스트할 때 출력값이 각각 어떻게 되겠는가?

(ii) 코드로 계산하시오.

6. (i) Drop Out 층으로 흘러들어온 미분과 생성된 random 행렬이 각각

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix}, \quad \begin{pmatrix} 0.2 & 0.1 & 0.5 \\ 0.5 & 0.3 & 0.9 \\ 0.2 & 0.5 & 0.9 \end{pmatrix}$$

이고 dropout ratio는 0.4라 하자. Drop Out층이 아래로 흘러 보내는 미분을 구하시오.

(ii) 코드로 계산하시오.

7. 다음 코드를 실행했을 때 출력될 값들을 대략적으로 쓰시오.

```

dropout = Dropout(0.3)
x = np.random.rand(100,100)
y = dropout.forward(x, train_flg=True)
print(np.sum(np.abs(y)<1e-7) / np.size(y))
print(np.mean(y))
z = dropout.forward(x, train_flg=False)
print(np.sum(np.abs(z)<1e-7) / np.size(z))
print(np.mean(z))

```