1. **(1.5 점: 문항당 0.5 점)**

(m x n) 실수(real) 행렬 A, B, C, 그리고 (n x 1) 실수 벡터 x 와 b 를 가정하라.
A 의 열벡터(column vector)들을 순서대로 $a_1, ..., a_n$ 이라 하고, T 는 transpose 를 의미하며,
'은 역행렬을 나타내고, 행렬 및 벡터의 곱을 *으로 표시한다고 하자.
다음 각 설명의 참/거짓을 O/X 로 표시하시오.

(1) y = A*x 라 하면, y 는 $a_1, ..., a_n$ 들의 선형결합이 되므로, y 는 $a_1, ..., a_n$ 들과 선형종속(linearly dependent)이다.

(2) $x = [1\ 2\ \dots\ n]^T$, D = diag(x), B = D*A 라 하면, B 의 k 번째 열벡터는 A 의 k 번째 열벡터에 k 를 곱한 것과 같아진다.

(3) A, B, C 가 non-singular square matrix 라 할 때, 다음 세 등식이 항상 성립한다:
$(A^T)' = (A')^T$, $((A - B)C)^T = C^T(A^T - B^T)$, $((A - B)C)' = C'(A' - B')$.

(1) O

(2) X

(3) X

2. (2.5 점: 문항당 0.5 점)

(2 x 2) 행렬 A 의 열벡터가 순서대로 $a_1 = [1\ 2]^T$, $a_2 = [2\ 1]^T$ 일 때, 다음 물음에 답하시오.

(1) A 의 두 열벡터가 이루는 각을 θ라 할 때, cos(θ)의 값을 구하시오.

(2) A 의 Frobenius Norm 의 제곱 값을 구하시오.

(3) A 의 고유값(eigenvalue)들의 합을 구하시오.

(4) 다음 중 A 의 고유벡터(eigenvector)에 해당하는 벡터들의 기호를 모두 고르시오.

a. $[0, 0]^T$ b. $[1, 1]^T$ c. $[1, -1]^T$ d. $[-1, 1]^T$ e. $[-1, -1]^T$

(5) A 는 고유값분해(eigen-decomposition)가 가능한 행렬인가?
고유값분해가 가능하다면 O, 그렇지 않으면 X 라 답하시오.

(1) 0.8

(2) 10

(3) 2

(4) a,b,c,d,e

(5)

3. (1.5 점: 문항당 0.5 점)

(m x n) 실수(real) 행렬 A 그리고 (n x 1) 실수 벡터 x 와 b 에 대해,
다음 각 설명의 참/거짓을 O/X 로 표시하시오.

(1) $y = b^T*A*x$ 라 하면, x 에 대한 y 의 gradient 는 항상 A^T*b 가 된다.

(2) $v = ||x||_2^2$ 라 하면, x 에 대한 v 의 gradient 는 항상 2x 가 된다.

(3) $w = x^T*A*x$ 라 하면, x 에 대한 w 의 gradient 는 항상 $2A*x$ 가 된다.

(1) O

(2) X

(3) O

4. (2 점: 빈칸당 0.5 점)

대학교 1 학년 성적을 기반으로 대학교 2 학년 성적을 예측하는 기계학습 알고리즘을 만들고자 한다. 변수 x 를 어떤 학생의 1 학년 A 학점 과목의 개수라 하고, 변수 y 를 그 학생의 2 학년 A 학점 과목의 개수라 할 때, 우측 표와 같이 4 명의 학생에 대한 x 와 y 데이터를 확보하였다고 하자. (이 표에서 각 행은 학생 한 명에 대한 x 와 y 데이터를 의미한다.) 기계학습 모델로 아래 식(1)로 주어진 선형회귀(linear regression) 모형을 사용하고, 비용함수로 아래 식(2)로 주어진 Sum of Squared Error 를 사용하며, 학습 방법으로 Full-batch Gradient Descent 방법을 사용한다고 할 때, 다음 물음에 답하시오.

x	y
3	4
2	1
4	3
0	1

식(1) $h_\theta(x) = \theta_0 + \theta_1 x$

식(2) $J(\theta_0, \theta_1) = \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

(1) $J(0,1)$의 값을 구하시오 (정수로 나누어 떨어지지 않을 경우 분수로 답을 기재하시오).

(2) Gradient vector $\nabla_\theta J(0,1)$의 각 원소의 합을 구하시오 (정수로 나누어 떨어지지 않을 경우 분수로 답을 기재하시오).

(3) (1 점) 위의 결과들을 이용하여, 모델 파라미터 θ_0 와 θ_1를 현재 값 $(0,1)$에서 각각 어떻게 갱신(update) 하는 게 좋을지 아래 a, b, c 중에 골라 기호로 답하시오.

a. 갱신하지 않고 유지

b. 더 큰 값으로 갱신

c. 더 작은 값으로 갱신

(1) 4

(2) 22

(3) θ_0 : a , θ_1 : c

5. (1 점: 문항당 0.25 점)

식 $h_\theta(x) = \theta_0 + \theta_1 x$ 로 주어진 선형모형을 이용하여 회귀(regression) 문제를 풀던 중에, m 개의 훈련샘플로 이루어진 훈련데이터셋에 대한 MSE (Mean Squared Error) 비용함수 값이 0 이 되는 (즉, $J(\theta_0, \theta_1) = 0$) 파라메터 값들 θ_0, θ_1을 찾을 수 있었다고 한다.

이때, 다음 각 설명의 참/거짓을 O/X 로 표시하시오.

(1) For this to be true, we must have $\theta_0 = 0$ and $\theta_1 = 0$ so that $h_\theta(x) = 0$.

(2) For this to be true, we must have $y^{(i)} = 0$ for every value of i = 1, 2, …, m.

(3) Our training set can be fit perfectly by a straight line, i.e., all of our training examples lie perfectly on some straight line.

(4) Gradient descent is likely to get stuck at a local minimum and fail to find the global minimum.

(1) O

(2) O

(3) O

(4) O

6. (1 점: 문항당 0.25 점)

Feature Scaling 에 대한 다음 각 설명의 참/거짓을 O/X 로 표시하시오.

(1) It prevents the matrix $(X^T X)^{-1}$ from being non-invertible when we use the normal equation.

(2) It speeds up solving for θ when we use the normal equation.

(3) It prevents gradient descent from getting stuck in local optima.

(4) It speeds up gradient descent by making it require fewer iterations to get to a good solution.

(1) X

(2) O

(3) X

(4) O

7. **(1.25 점: 문항당 0.25 점)**

훈련데이터셋이 아래 표와 같이 주어졌을 때, $h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$인 logistic regression classifier 를 설계하고자 한다. 다음 문장들의 참/거짓 여부를 O/X 로 표시하시오.

x_1	x_2	y
1	0.5	0
1	1.5	0
2	1	1
3	1	0

(1) If we set the cost $J(\theta_0, \theta_1, \theta_2)$ be a convex function, then gradient descent should converge to the global minimum.

(2) If we set the cost $J(\theta_0, \theta_1, \theta_2)$ be a convex function and if we can find such $(\theta_0, \theta_1, \theta_2)$ that can make $J(\theta_0, \theta_1, \theta_2) = 0$, then this solution gives us the global minimum.

(3) Because positive examples (i.e., y=1) and negative examples (i.e., y=0) cannot be separated using a straight line, gradient descent will fail to converge.

(4) Adding polynomial features (e.g., using $h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2)$) could increase how well we can fit the training data.

(5) Adding polynomial features (e.g., using $h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2)$) could decrease the bias of our model but could increase the variance of our model.

8. (1 점: 문항당 0.25 점)

m 개의 훈련샘플로 이루어진 훈련데이터셋에 대해 비용함수를 아래 식(1)의 MSE (Mean Square Error)로 설정한 logistic regression 문제를 생각하자. σ는 sigmoid function 을 나타낸다고 하고, 학습률(learning rate)을 α 라 할 때, Stochastic Gradient Descent (즉, online mode)를 사용하는 파라메터 갱신식으로 올바른 것은 O, 그렇지 않은 것은 X 로 표시하시오.

식(1) $J(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2$ where $h_\theta(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T\mathbf{x})$

(1) $\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)}\right)\mathbf{x}^{(i)}$

(2) $\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{1+\exp\left(-\boldsymbol{\theta}^T x^{(i)}\right)} - y^{(i)}\right)\mathbf{x}^{(i)}$

(3) $\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)}\right)x_j^{(i)}$ (simultaneously update for all j)

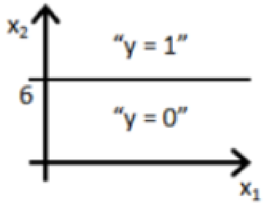
(4) $\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\sigma(\boldsymbol{\theta}^T\mathbf{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$ (simultaneously update for all j)

9. (0.75 점)

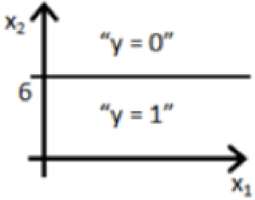
Logistic classifier $h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$를 학습시켜 최적 파라메터로

$\theta_0 = 6,\ \theta_1 = 0,\ \theta_2 = -1$를 찾았다고 한다.

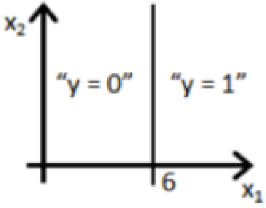
이 classifier 의 결정경계(decision boundary)를 나타낸 그림으로 올바른 것을 고르시오.



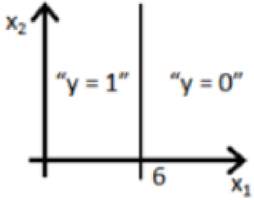
(a)



(b)



(c)



(d)

10. (2 점)

아래 코드는 Logistic Regression 모델을 Python 클래스로 구현한 것이다. 목적함수로는 Cross Entropy, 활성함수로는 Sigmoid 함수를 사용하였고, 최적화 방법으로는 Gradient Descent 를 사용하였다고 할 때 다음 물음에 답하시오.

```
class LogisticRegression():
#클래스 내에서 멤버 변수와 함수를 호출할 때는 self.variable 과 self.function()으로 사용합니다.
    def __init__(self, num_features):
        self.weights = [random.random() for i in range(num_features)]
        self.bias = 0
        self.lr = 0.01    # learning rate

    def sigmoid(self, z):
        return 1 / (1 + math.exp(-z))

    def forward(self, x):
        linear = sum([x[i] * self.weights[i] for i in range(len(x))])
        linear += self.bias
        y_hat = self.sigmoid(linear)
        return y_hat

    def backward(self, x, y):
        error = ([가] - [나])
        return error

    def train(self, x, y, epochs):
        for e in range(epochs): # epochs 만큼 학습
            for i in range(len(y)):
                x_, y_ = x[i], y[i] # Each training sample

                # To update the weights and bias with backward()
                for j in range(len(self.weights)):
                    error = self.backward(x_, y_)
                    gradient = error * [다]
                    self.weights[j] -=  gradient * self.lr
                self.bias -= error * self.lr
```

(1) (0.5 점) 이 코드에 대한 설명으로 옳은 것은 다음 중 어느 것인가?

a. Batch-size 가 1 인 Online mode 를 사용하고 있다.

b. Batch-size 가 훈련샘플의 개수와 같은 Full-batch mode 를 사용하고 있다.

c. Batch-size 가 1 보다 크고 훈련샘플의 개수보다 작은 Mini-batch mode 를 사용하고 있다.

(2) (빈칸당 0.5 점) 이 코드의 빈칸 [가]~[다]에 적합한 코드를 삽입하여, weights 와 bias 가 올바로 갱신되도록 Logistic Regression 모델을 완성하시오.

11. (2.5 점: 빈칸당 0.5 점)

다음 표는 표적 장기에 대한 종양의 개수와 종양의 최대 크기(mm)를 바탕으로 악성 종양 여부를 판단하는 문제에 대한 훈련데이터셋을 나타낸다.

이 표에서 x_1은 종양의 수, x_2는 종양 최대 크기, y는 악성 여부(1: 악성, 0: 양성)를 타나낸다.

이 문제를 퍼셉트론(perceptron) 모델로 풀고자 할 때, 다음 물음에 답하시오.

x_1	x_2	y
1	2	0
2	3	0
3	5	1
4	6	1

$$\tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases}$$

$$\sigma(s) = \frac{1}{1 + e^{-s}}$$

(1) 퍼셉트론의 활성함수로 계단함수 τ를 사용한다고 하자. 이 퍼셉트론의 가중치 벡터가 $w = [-1\ 1\ 1]^T$ 일 때, 첫 번째 훈련샘플에 대한 퍼셉트론의 출력은 얼마인가?

(2) 퍼셉트론의 활성함수로 시그모이드함수 σ를 사용한다고 하자. 이 퍼셉트론의 가중치 벡터가 $w = [-1\ 1\ 1]^T$ 일 때, 첫 번째 훈련샘플에 대한 Cross-Entropy Loss J(w)를 구하여 정리하였더니 아래 식과 같았다고 한다. 이 식의 a 와 b 에 해당하는 값을 구하시오.

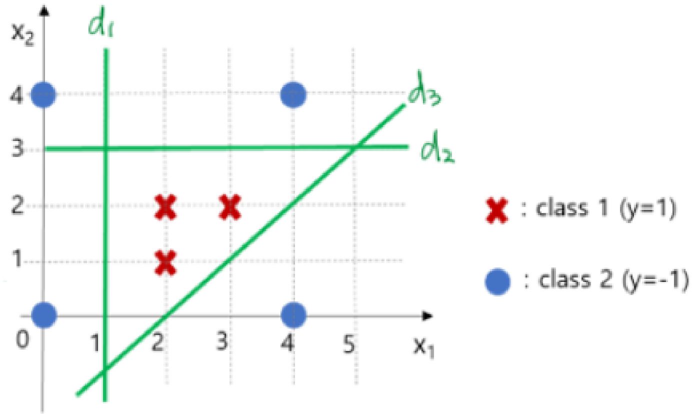
$J(w) = \log(a + e^{b})$

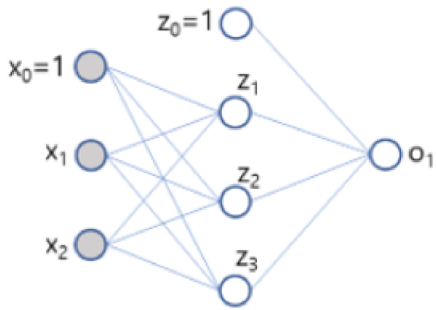
(3) 퍼셉트론의 활성함수로 시그모이드함수 σ를 사용한다고 하자. 이 퍼셉트론의 가중치 벡터가 $w = [-1\ 1\ 1]^T$ 일 때, 첫 번째 훈련샘플에 대한 Cross-Entropy Loss J(w)의 gradient 를 구하여 정리하였더니 아래 식과 같았다고 한다. 이 식의 c 와 d 에 해당하는 값을 구하시오.

$\nabla J(w) = \sigma(c) \cdot [1\ 1\ d]^T$

12. (7 점)

아래 그림 좌측에 나타낸 이진분류(binary classification) 문제를 아래 그림 우측에 나타낸 2 층 퍼셉트론 모델을 이용하여 풀고자 한다. 이때 모든 퍼셉트론의 활성화함수로는 계단함수 τ를 사용하고, 계단함수의 0 에서의 미분값은 0 으로 정의한다.





이 그림에서 x_i는 i 번째 입력 feature 값을 나타내고, z_j는 은닉층의 j 번째 뉴런의 출력값을 나타내며, o_1은 출력층 뉴런의 출력값을 나타낸다.

주어진 2 층 퍼셉트론의 은닉층 가중치 행렬 U 와 출력층 가중치 행렬 V 를 다음과 같이 정의하자:

- u_{ji}를 i 번째 입력뉴런과 j 번째 은닉뉴런을 연결하는 가중치라 하면, 은닉층 가중치 행렬 U 는 j 번째 행 i 번째 열의 원소로 u_{ji}를 갖는 3x3 행렬로 정의된다. (i=0,1,2; j=1,2,3).
- v_{kj}를 j 번째 은닉뉴런과 k 번째 출력뉴련을 연결하는 가중치라 하면, 출력층 가중치 행렬 V 는 k 번째 행 j 번째 열의 원소로 v_{kj}를 갖는 1x4 행렬로 정의된다. (j=1,2,3,4; k=1)

(1) (3 점) Decision boundary 를 이용한 가중치 결정:

그림에 나타낸 3 개의 decision boundary $d_1(x_1,x_2)$, $d_3(x_1,x_2)$, $d_3(x_1,x_2)$를 사용하면 두 클래스를 오류없이 분류할 수 있다. 그림에 나타낸 decision boundary d_1, d_2, d_3를 은닉층 뉴런 z_1, z_2, z_3이 각각 담당하도록 하여 주어진 이진문제를 오류없이 해결하는 가중치 행렬 U 와 V 를 구했더니, V = [-2.5 1 1 1]가 되었다고 한다. 이때 U 를 구하시오.

(1) U의 첫번째 행: [1, -1, 0] , U의 두번째 행: [3, 0, -1] , U의 세번째 행: [-2, 1, -1]

(2) (4 점) **Stochastic Gradient Descent 를 이용한 가중치 결정:**

이번에는 비용함수를 $J(\mathrm{U},\mathrm{V})=\frac{1}{2}\|o_1-y\|_2^2$ 로 정의한 후, 아래 Algorithm 의 Stochastic Gradient Descent 방법을 이용하여 가중치 행렬을 구하고자 한다. 이 Algorithm 을 컴퓨터로 수행하던 중 어떤 순간에 일시 정지하였더니 line 6 에서 멈추었다고 하자. 이 순간에서의 훈련샘플과 가중치 행렬들의 값이 아래와 같았다고 할 때 다음 물음에 답하시오.

- **X** = [1 2 2]^T
- U = [[-1 1 0], [1 0 -1], [0 1 0]].
- V = [-2 1 1 1]

Algorithm: Stochastic GD를 이용한 2층 퍼셉트론 학습

- 훈련집합: $\mathbb{X} = \{(0,0), (0,4), (4,0), (4,4), (2,1), (2,2), (3,2)\}$, $\mathbb{Y} = \{-1, -1, -1, -1, 1, 1, 1\}$
- 학습률=ρ
- 활성함수: 계단함수 τ
- 출력: 가중치 행렬 U와 V

1	가중치 행렬 U와 V를 초기화 한다.
2	Repeat
3	$\mathbb{X}$의 순서를 섞는다.
4	for ($\mathbb{X}$의 샘플 각각에 대해 반복)
5	현재 처리하는 샘플을 $\mathbf{x}=(x_0, x_1, x_2)$와 y라 표기. 이때 bias x_0와 z_0는 각각 1로 설정.
6	# 전방 계산 (forward)
7	$\tilde{\mathbf{z}}_{sum} = \mathrm{U}\mathbf{x}$. $\tilde{\mathbf{z}} = \boldsymbol{\tau}(\tilde{\mathbf{z}}_{sum})$. # 은닉층 출력값 계산
8	$o_{sum} = \mathrm{V}\mathbf{z}$. $o = \tau(o_{sum})$. # 출력층 출력값 계산
9	# 위에서 $\tilde{\mathbf{z}}$는 $\mathbf{z}$의 bias(첫번째 원소)를 제외한 벡터. (예: $\mathbf{z}=[a\ b\ c\ d]^T \rightarrow \tilde{\mathbf{z}}=[b\ c\ d]^T$)
10	# 오류 역전파 (backward)
11	$\delta = (y - o) \odot \tau'(o_{sum})$. # 출력층 오차에 대한 미분 계산
12	$\boldsymbol{\eta} = (\delta^T \tilde{\mathrm{V}})^T \odot \boldsymbol{\tau}'(\tilde{\mathbf{z}}_{sum})$. # 은닉층 오차에 대한 미분 계산
13	# 위에서 $\tilde{\mathrm{V}}$는 V의 bias(첫번째 원소)를 제외한 행렬. (예: $\mathrm{V}=[a\ b\ c\ d] \rightarrow \tilde{\mathrm{V}}=[b\ c\ d]$)
14	# 가중치 갱신 (parameter update)
15	$\Delta\mathrm{V} = -\delta\mathbf{z}^T$. $\mathrm{V} = \mathrm{V} - \rho\Delta\mathrm{V}$. # 출력층 가중치 갱신
16	$\Delta\mathrm{U} = -\boldsymbol{\eta}\mathbf{x}^T$. $\mathrm{U} = \mathrm{U} - \rho\Delta\mathrm{U}$. # 은닉층 가중치 갱신
17	Until (ΔV==0 && ΔU==0) # 멈춤 조건: 가중치가 더 이상 갱신되지 않으면 중지

(2a) Line 8의 o_{sum} 값을 계산하시오. 4

(2b) Line 11의 δ 값을 계산하시오.

(2c) Line 12의 벡터 $\boldsymbol{\eta}$의 모든 원소의 절대값의 합을 계산하시오.

(2d) 이 알고리즘을 멈춤조건에 도달할 때까지 수행한 후, 학습된 가중치들을 이용하여 훈련 샘플들에 대해 분류하면 **위 (1)에서와 달리 오분류 샘플이 많이 발생**하게 된다. 이런 결과가 발생하는 가장 중요한 문제가 무엇인지, 또 해결방법은 무엇인지 간략히 기술하시오.

- 문제점 가중치가 과도하게
- 해결책 가중치 갱신에 규제

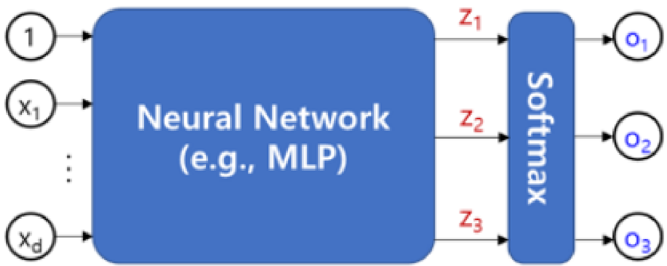
13. (1.5 점: 문항당 0.5 점)

아래 그림은 3 개의 class {1, 2, 3}를 분류하기 위한 신경망(neural network) 모델을 나타낸다.
어떤 training example 의 입력 $\mathbf{x} = [x_1\ x_2\ \dots\ x_d]^T$ 에 대해 softmax 를 통과하기 직전의 출력벡터를 $\mathbf{z} = [z_1\ z_2\ z_3]^T = [10\ {-15}\ 5]^T$ 라 하고, one-hot encoding 된 target vector 를 $\mathbf{y} = [0\ 1\ 0]^T$ 라 하자.
비용함수로 Cross-Entropy 를 사용한다고 할 때, 다음 설명의 참/거짓을 O/X 로 표시하시오.

(1) 이 training example **x** 는 class 1, 2, 3 중 class 1 으로 분류될 것이다.

(2) 이 training example **x** 에 대한 Cross-Entropy 는 o_1 과 o_3 값과는 무관하고 o_2 값에 의해서만 결정된다.

(3) Softmax 를 포함한 출력층의 가중치에 대한 비용함수의 gradient 는 o_1 과 o_3 값과는 무관하고 o_2 값에 의해서만 결정된다.



14. (1 점: 문항당 0.25 점)

Overfitting 을 방지하기 위해 weight penalty(가중치 벌칙)로 L1-norm 을 사용하는 경우와 L2-norm 을 사용하는 경우의 공통점과 차이점에 대해 설명한 것이다.
각 설명에 대해 맞으면 O, 틀리면 X 로 표시 하시오.

(1) L1 과 L2 모두 가중치의 크기를 0 에 가깝게 하는 효과가 있다.

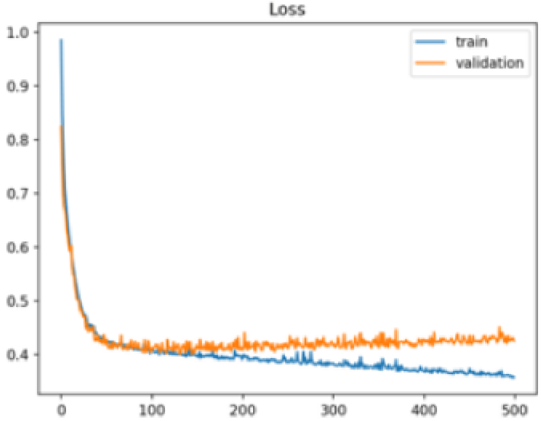
(2) L1 과 L2 모두 bias 에 해당하는 가중치에는 weight penalty 를 적용하지 말아야 한다.

(3) 가중치 업데이트시 weight penalty 에 의해 변화되는 가중치의 크기는, L1 의 경우 현재 가중치 크기와 무관하게 일정하지만 L2 의 경우는 현재 가중치 크기에 비례한다.

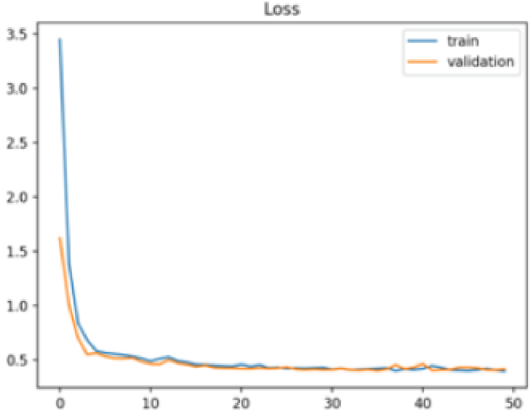
(4) L1 은 L2 보다 non-zero 가중치가 더 많이 발생하는 경향이 있다.

15. (2 점: 빈칸당 0.5 점)

다음 그래프들은 training set 과 validation set 에 대한 기계학습 모델의 학습곡선(learning curve)들을 나타낸다. (학습곡선에서 가로축은 epoch 수를, 세로축은 매 epoch 수행수의 loss 값을 나타낸다.) 각 그래프는 서로 다른 데이터셋과 모델을 사용한 결과이며, 각 그래프에 나타낸 최종 epoch 만큼 해당 모델을 학습 시켰다고 한다. 그래프 1~2 각각의 **최종학습상태**와 **개선방안**을 각각 **<보기 1>과 <보기 2>에서** 고르되, **해당되는 기호를 모두** 고르시오.



그래프 1



그래프 2

<보기 1. 학습된 모델의 상태>

A. 이 모델은 과소적합(underfit)일 가능성이 높다.

B. 이 모델은 과대적합(overfit)일 가능성이 높다.

C. 이 모델은 적합(good fit) 상태일 가능성이 높다.

D. 이 모델은 high-bias 일 가능성이 높다.

E. 이 모델은 high-variance 일 가능성이 높다.

<보기 2. 개선 방안>

가. Epoch 수를 늘려 학습을 더 시키는 게 도움이 된다.

나. Training data 를 늘려 다시 학습 시키는 게 도움이 된다.

다. Weight penalty 와 같은 regularization(규제) 방법을 사용하여 다시 학습 시키는 게 도움이 된다.

라. 모델의 용량(capacity)를 늘려 다시 학습 시키는 게 도움이 된다.

마. 현재 상태로 학습을 완료하는 게 좋다.

바. Validation loss 가 가장 작은 epoch 에서 학습을 조기종료(Early Stop)하는 게 도움이 된다.

(1) 그래프1: 상태 = B, D , 개선 방안 = 나, 다, 바

(2) 그래프2: 상태 = C , 개선 방안 = 마

문제 16

(1) 3-way hold-out 방법을 사용한 **모델 선택 및 평가** 과정에 대해 설명하시오.

문제 17

(2) 5-fold cross-validation 방법을 사용한 **모델 선택** 과정에 대해 설명하시오.

문제 18

0.25 / 0.5점

(3) 3-way hold-out 방법과 k-fold cross-validation 방법은 각각 **어떤 상황**에서 적용하면 좋을 지 설명하시오.