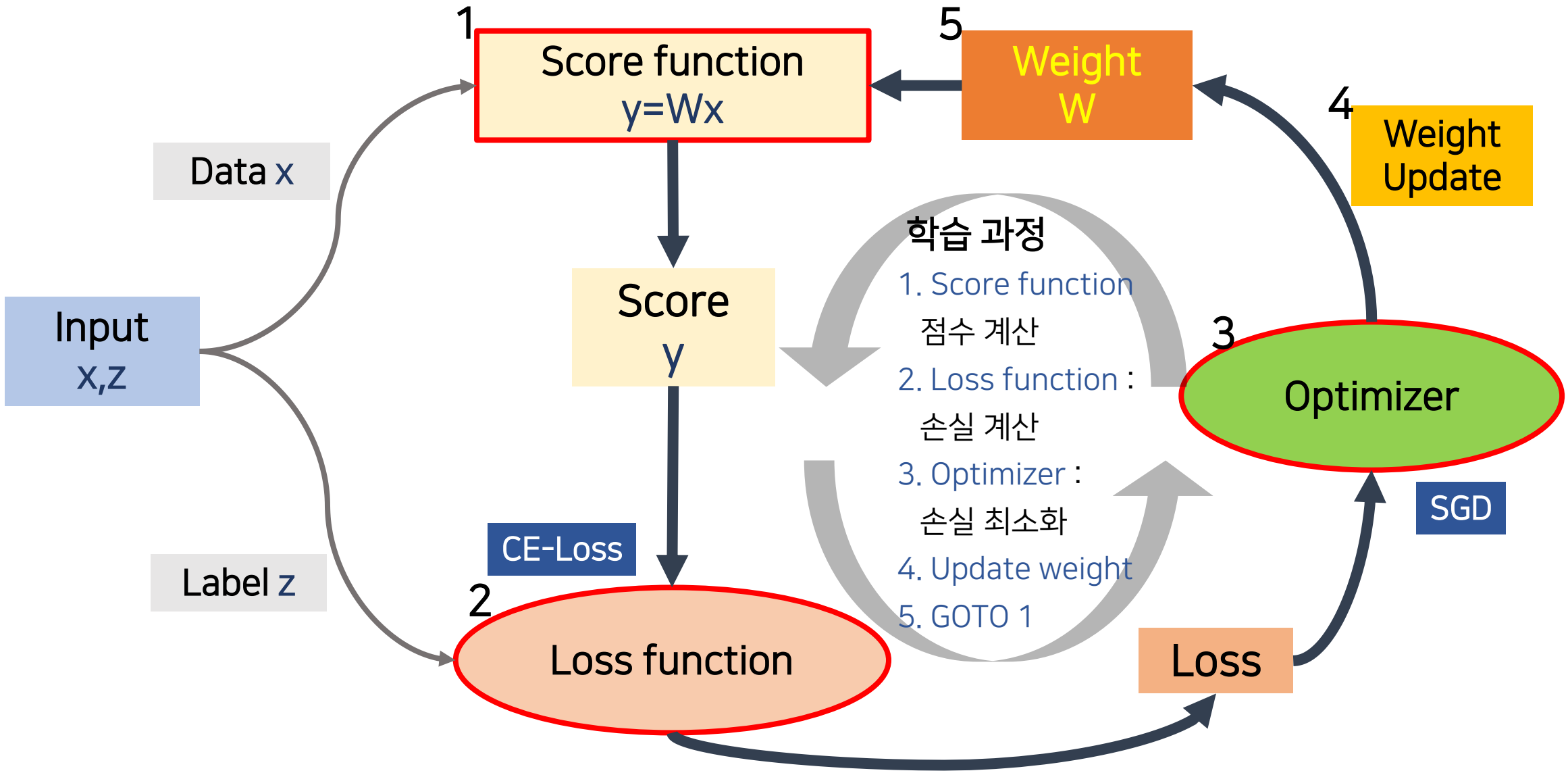
Contents

4장 선형모델

선형 모델 과정


1
Score function
y=Wx
5
Weight
W
4
Weight
Update
Data x
Input
x,z
Score
y
학습 과정
1. Score function
점수 계산
2. Loss function :
손실 계산
3. Optimizer :
손실 최소화
4. Update weight
5. GOTO 1
3
Optimizer
SGD
CE-Loss
Label z
2
Loss function
Loss

Contents

4장 선형모델

이미지 데이터

Gray image

1 channel

0~255

Color image

RGB 3-channel

0~255

0~255

0~255

벡터화

입력 이미지

45	66	76	3
9	56	252	54
5	12	145	83
45	59	134	123

4x4 pixel

45	66	76	3	9	56	252	54	5	12	145	83	45	59	134	123

입력 테이터 : 벡터(16)

Contents

4장 선형모델

Score 함수

$$f(x, W) = Wx + b = y$$

flatten

score function

x → f → $Wx + b$

score

36.8

Input image

W: weight

b: bias

Class: cat

z: Label

선형 분류 모델

0.2	-0.4	0.1	0.7
0.7	0.4	-0.4	0.5
1.2	1.3	2.1	-1.1

W

43
204
35
150

x

\+

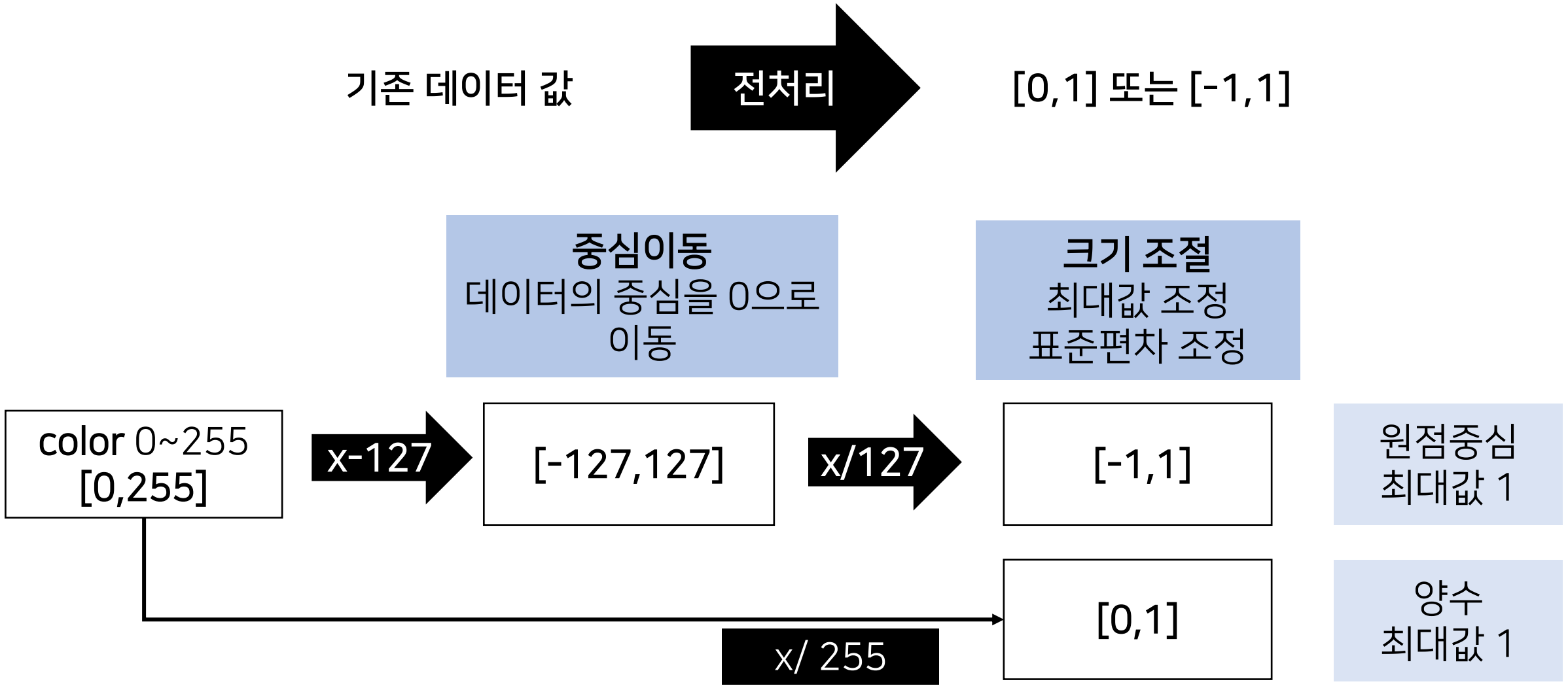
1.3
2.1
-1.3

b

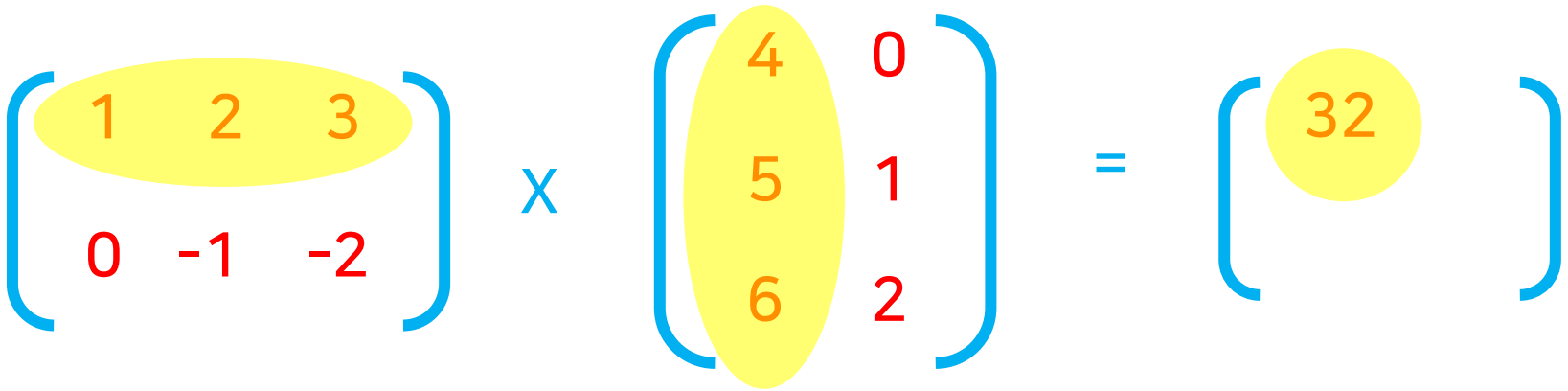
=

36.8	Cat Score
174.8	Dog Score
224	Ship Score

데이터 전처리


기존 데이터 값
전처리
[0,1] 또는 [-1,1]
중심이동
데이터의 중심을 0으로 이동
크기 조절
최대값 조정
표준편차 조정
color 0~255
[0,255]
x-127
[-127,127]
x/127
[-1,1]
원점중심
최대값 1
x/ 255
[0,1]
양수
최대값 1

행렬 곱셈



1*4 + 2*5 + 3*6 = 32

행렬 곱셈과 내적

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} X \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = 1$$

$$1*0 + 2*(-1) + 3*1 = 1$$

선형 모델 가중치 학습 결과

Contents

4장 선형모델

Softmax 분류기

Score

$$z_i = f(x_i, W) = Wx_i$$

softmax

$$s_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

C-E Loss

$$L_i = -\log s_i$$

전체 손실

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i$$

Softmax 과정

Softmax function

$$f_j(z) \rightarrow [e^{z_j}] \rightarrow \frac{e^{z_j}}{\sum_k e^{z_k}} := s_j$$

모든 수를 양수로 변환

전체 합이 1이 되도록 변환

z → **지수화 $\exp(z)$** → **확률분포**

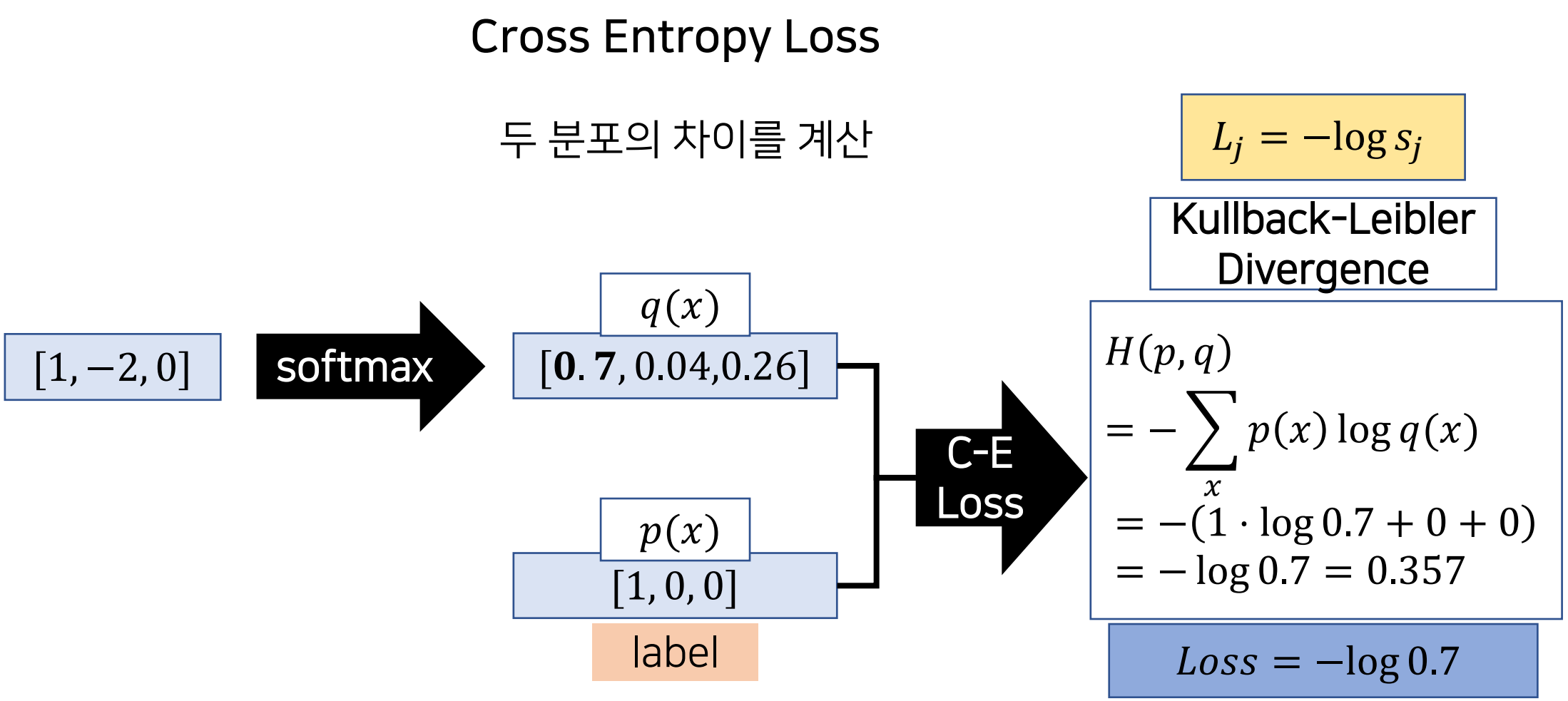
$[1, -2, 0] \rightarrow [e^1, e^{-2}, e^0]$

$2.71 + 0.14 + 1 = 3.85$

$[2.71, 0.14, 1]/3.85$

$[\mathbf{1}, \mathbf{-2}, \mathbf{0}]$ $[\mathbf{2.71}, \mathbf{0.14}, \mathbf{1}]$ $[\mathbf{0.7}, \mathbf{0.04}, \mathbf{0.26}]$

Cross Entropy Loss



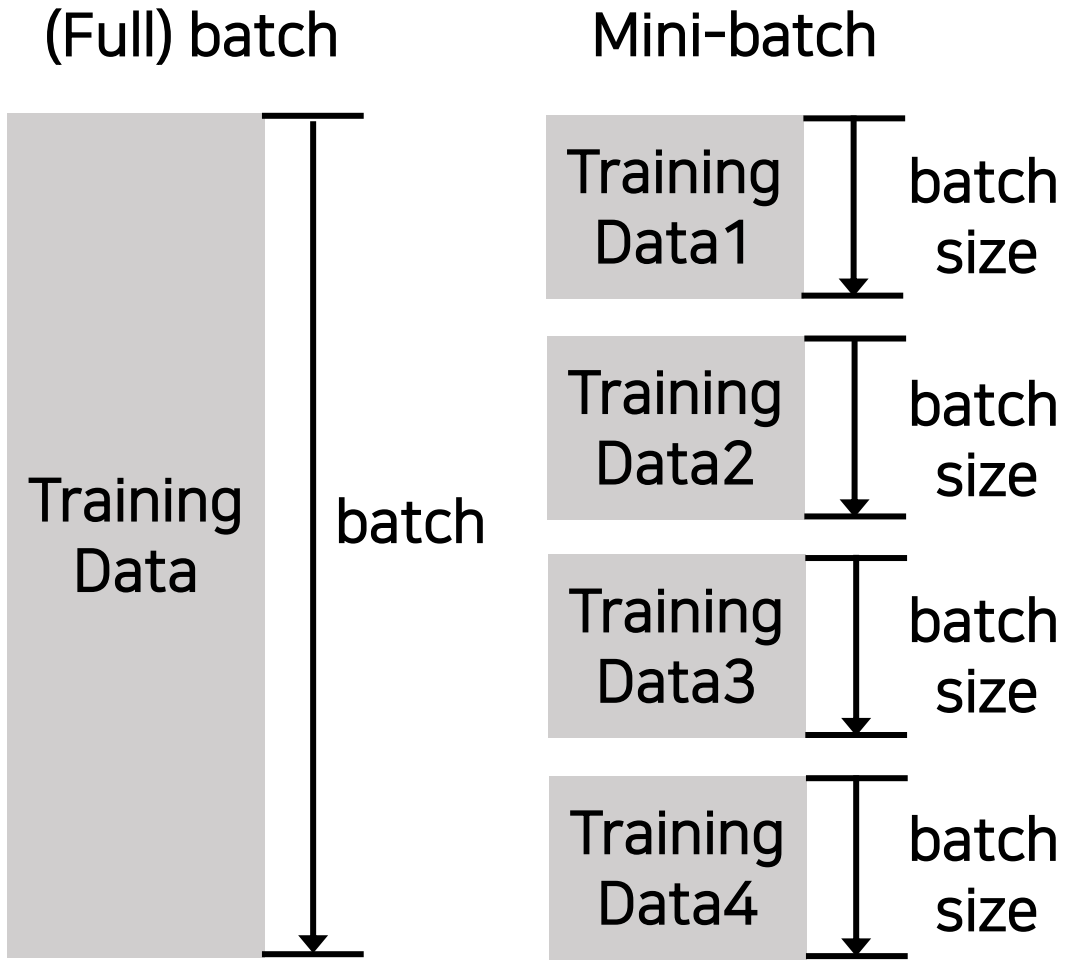
Contents

4장 선형모델

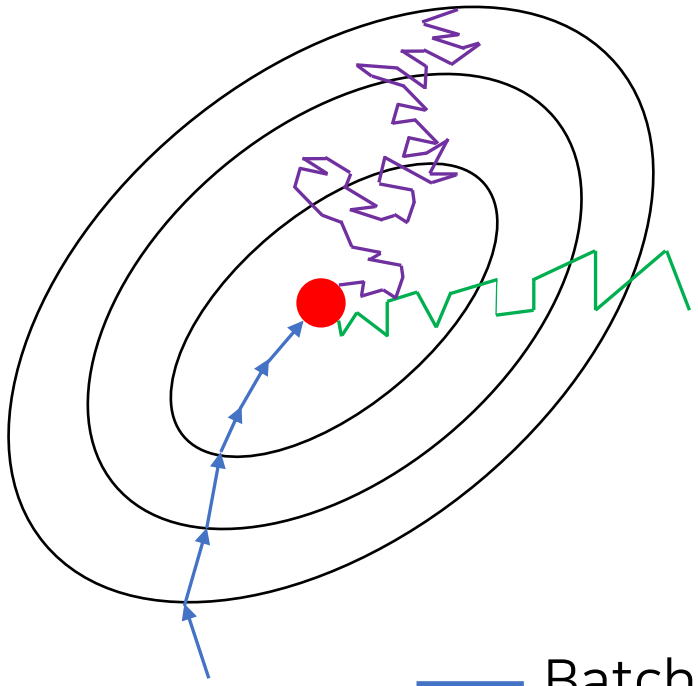
SGD

: 8, 16, 32, 64 배치 사이즈로 계산

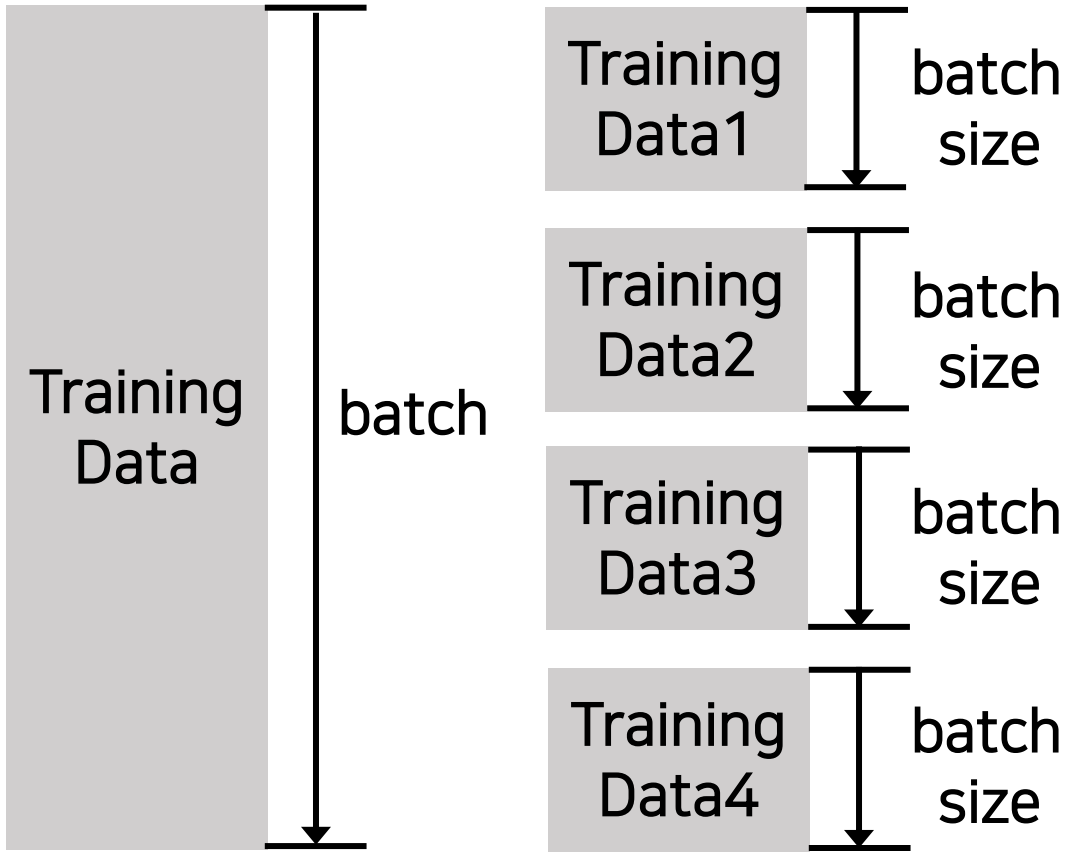
싼 비용
계산 속도 빠름
노이즈 있음
덜 안정적



SGD



Batch GD
Mini-batch GD
Stochastic GD



Contents

4장 선형 모델

실습예제

선형 분류 Mnist

주요 명령어 1/2

PIL

```
from PIL import Image #이미지 처리용 라이브러리
Image.fromarray() #array(배열)을 이미지로 변환
Image.open(path) #이미지(path) 불러오기
```

주요 명령어 2/2

pyplot

```
from matplotlib import pyplot as plt #그래프 도구 라이브러리
plt.imshow(img) # 이미지(img) 그리기
plt.xlim() # x축의 범위 설정
plt.plot(x, y, 'b') #그래프 그리기(x 값, y값, 'b' : 색상 설정)
```

선형분류 coding / data loading

numpy /pandas 호출

```
import numpy as np
import pandas as pd
```

data load

```
from tensorflow.keras.datasets.mnist import load_data
```

```
(train_x, train_y), (test_x, test_y) = load_data()
```

```
train_x.shape, train_y.shape
```

선형분류 coding / data 확인

data 확인

```
from PIL import Image
img=train_x[0]

import matplotlib.pyplot as plt
img1=Image.fromarray(img, mode='L')
plt.imshow(img1)

train_y[0]    #첫번째 데이터 label 확인
```

선형분류 coding / data 전처리

data 전처리

벡터화

```
train_x1=train_x.reshape(60000,-1)

test_x1=test_x.reshape(10000,-1)
```

크기 조절

```
train_x2=train_x1/255

test_x2=test_x1/255
```

선형분류 coding / 모델 설정

모델 설정

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
md=Sequential()  # 모델명을 md로 정의
md.add(Dense(10, activation='softmax', input_shape=(28*28,)))
md.summary()
```

선형분류 coding / 학습 설정

compile/fit

```
md.compile(loss='sparse_categorical_crossentropy',  optimizer='sgd’,
            metrics='acc')
hist=md.fit(train_x, train_y, epochs=30, batch_size=64,
            validation_split=0.2)
```

선형분류 coding / 학습 분석

학습 분석 그래프

```
acc=hist.history['acc']
val_acc=hist.history['val_acc']
epoch=np.arange(1,len(acc)+1) #x축 값 설정(1~epochs 값)


plt.figure(figsize=(10,8))
plt.xlim(250,len(acc)+1) #x축 범위 설정
plt.plot(epoch,acc, 'b',label='acc')
plt.plot(epoch, val_acc, 'g', label='val_acc')
plt.legend()
```

선형분류 coding / 평가

평가

```
md.evaluate(test_x2, test_y)
```

#0.9211999