



클라우드 컴퓨팅과 AI서비스 (13주차)

융합학과 권오영

oykwon@koreatech.ac.kr

학습내용

- ❖ 프로젝트 안내
- ❖ Matrix 연산

AI 프로젝트에 활용되는 패키지

AI는 어떻게 시작해야 할까?

- ❖ 데이터 수집: 데이터 획득과 관리
- ❖ 연산 및 패턴인식: 계산 시스템
- ❖ 사고 및 학습 (문제해결방법) : 고급 AI 알고리즘



```
import tensorflow as tf  
import numpy as np  
import matplotlib.pyplot as plt
```

```
turtle      import turtle as t
```

```
minecraft    from mcpi.minecraft import minecraft as mf
```

Tensorflow/Keras 활용

```
# 텐서플로우 수입
import tensorflow as tf
```

```
# 텐서플로우로 hello world 출력
msg = tf.constant('hello world!')
tf.print(msg)
```

```
# MNIST 손글씨 데이터 package 수입
mnist = tf.keras.datasets.mnist
```

```
# MNIST 4분할 데이터 불러오기
(X_train, Y_train), (X_test, Y_test) =
mnist.load_data()
print('학습용 입력 데이터 모양:', X_train.shape)
print('학습용 출력 데이터 모양:', Y_train.shape)
print('평가용 입력 데이터 모양:', X_test.shape)
print('평가용 출력 데이터 모양:', Y_test.shape)
```

```
# 이미지 데이터 원본 출력
import matplotlib.pyplot as plt
plt.imshow(X_train[0], cmap='gray')
plt.show()
```

```
print('첫번째 학습용 데이터 입력값:', X_train[0])
print('첫번째 학습용 데이터 출력값:', Y_train[0])
```

```
# 이미지 데이터 [0, 1] 스케일링
X_train = X_train / 255.0
X_test = X_test / 255.0
```

```
# 스케일링 후 데이터 확인
plt.imshow(X_train[0], cmap='gray')
plt.show()
print('첫번째 학습용 데이터 입력값:', X_train[0])
```

인공신경망 구현

```
model = tf.keras.models.Sequential()
layers = tf.keras.layers
```

```
model.add(layers.Flatten(input_shape=(28, 28)))
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(10,
activation='softmax'))
```

```
# 인공신경망 요약
model.summary()
```

```
# 인공신경망 학습 환경 설정
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
# 인공신경망 학습
model.fit(X_train, Y_train, epochs=5)
```

```
# 인공신경망 평가
model.evaluate(X_test, Y_test)
```

```
# 인공신경망 예측
pick = X_test[0].reshape(1, 28, 28)
pred = model.predict(pick)
answer = tf.argmax(pred, axis=1)
```

```
print('인공신경망 추측 결과 (원본):', pred)
print('인공신경망 추측 결과 (해석):', answer)
print('정답:', Y_test[0])
```

STREAMLIT

Streamlit

- ❖ <https://streamlit.io/>
- ❖ 데이터 앱을 빠르게 만들고 배포할 수 있는 오픈소스 파이썬 라이브러리
- ❖ 일반적인 컴퓨터

\$ pip install streamlit

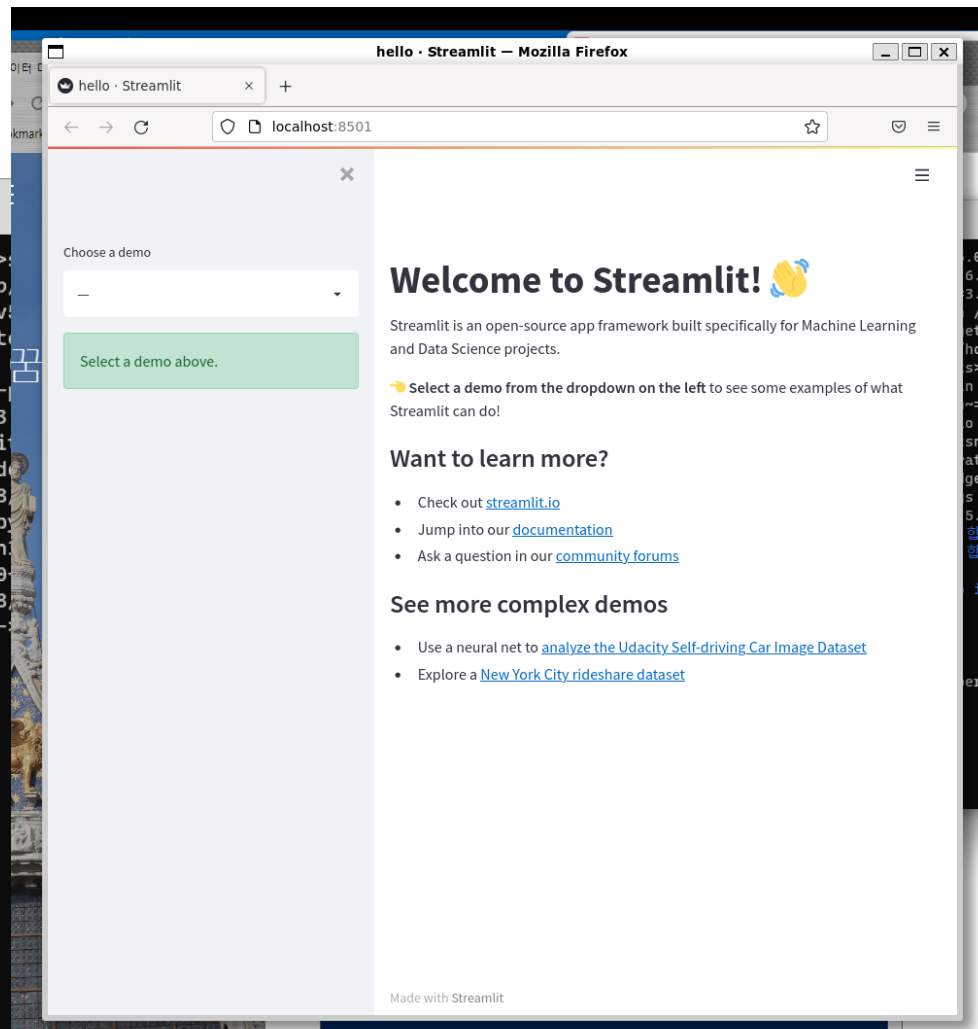
```
oykwon@Ohyoung-K506: /mr x + v
->notebook>=4.4.1->widgetsnextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in /home/oykwon/anaconda3/lib/python3.8/site-packages/nbclient-0.5.1-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: testpath in /home/oykwon/anaconda3/lib/python3.8/site-packages/testpath-0.4.4-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: bleach in /home/oykwon/anaconda3/lib/python3.8/site-packages/bleach-3.1.0-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: pycparser in /home/oykwon/anaconda3/lib/python3.8/site-packages/pycparser-2.19-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: nest-asyncio in /home/oykwon/anaconda3/lib/python3.8/site-packages/nest-asyncio-1.5.1-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: nbconvert in /home/oykwon/anaconda3/lib/python3.8/site-packages/nbconvert-6.0.0-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: async-generator in /home/oykwon/anaconda3/lib/python3.8/site-packages/async-generator-1.10-py3.8.egg (from notebook==4.4.1)
Requirement already satisfied: webencodings in /home/oykwon/anaconda3/lib/python3.8/site-packages/webencodings-0.5.1-py3.8.egg (from notebook==4.4.1)
(base) oykwon@Ohyoung-K506: /mnt/c/Users/용 할$ pip install streamlit^C
(base) oykwon@Ohyoung-K506: /mnt/c/Users/용 할$ streamlit hello

Welcome to Streamlit. Check out our demo in your browser.

Local URL: http://localhost:8501
Network URL: http://172.24.133.115:8501

Ready to create your own Python apps super quickly?
Head over to https://docs.streamlit.io

May you create awesome apps!
```



Streamlit Demo

❖ Interactive Web App with Streamlit and Scikit-learn (<https://github.com/python-engineer/streamlit-demo>)

Installation

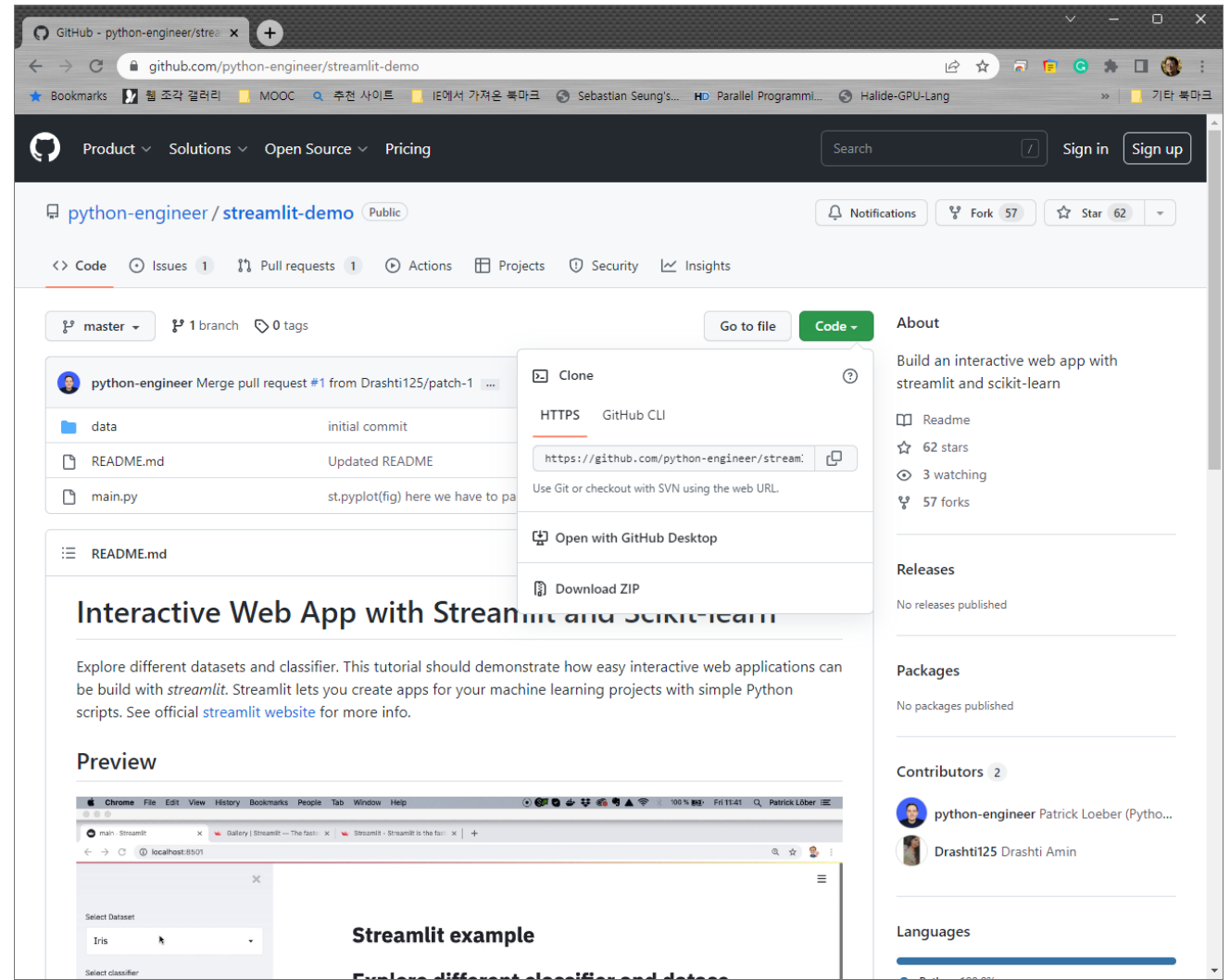
You need these dependencies:

```
pip install streamlit
pip install scikit-learn
pip install matplotlib
```

Usage

Run

```
streamlit run main.py
```

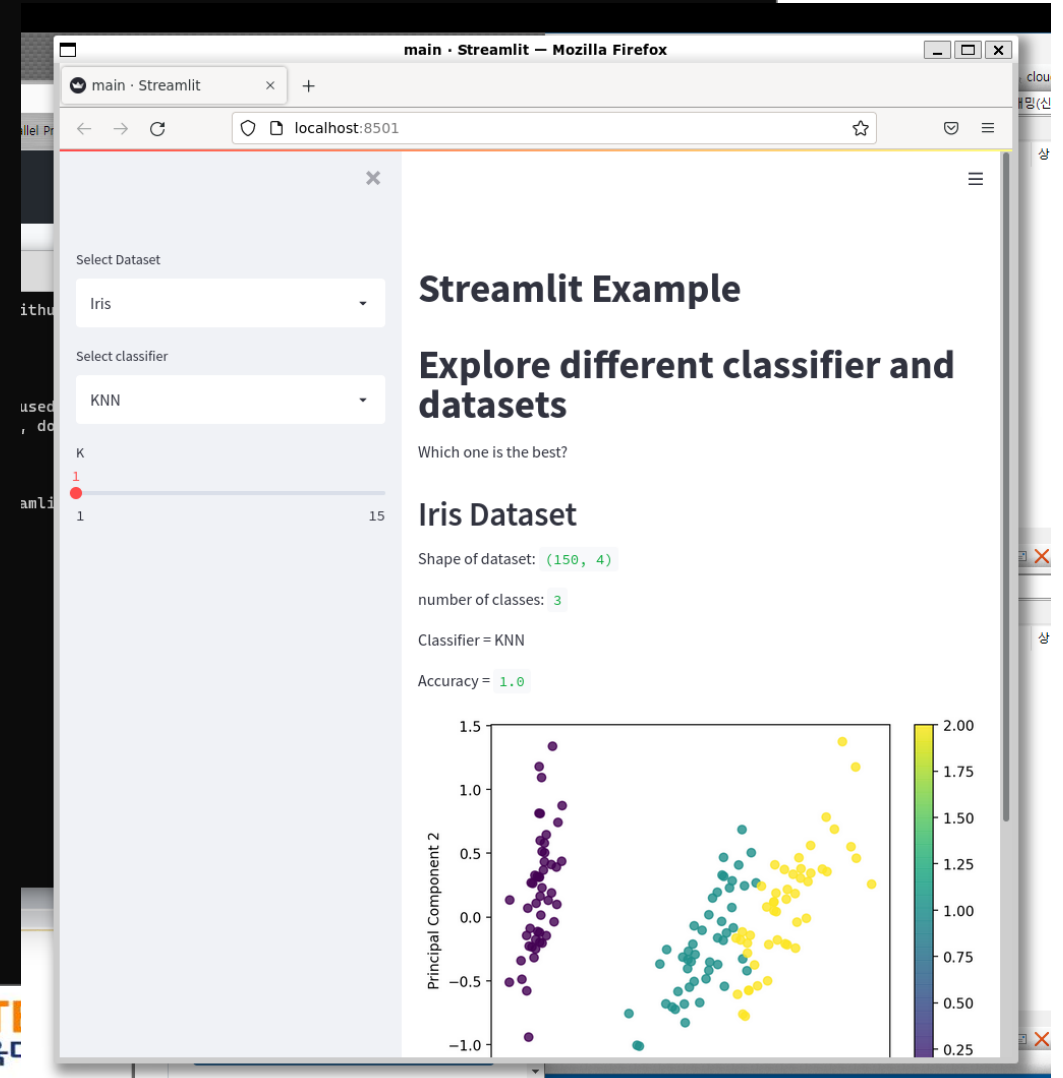


Streamlit Demo

```
oykwon@Ohyoung-K506: ~/c X + v
(base) oykwon@Ohyoung-K506:~/codes$ git clone https://github.com/python-engineer/streamlit-demo.git
Cloning into 'streamlit-demo'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 16 (delta 4), reused 9 (delta 1), pack-reused 0
Unpacking objects: 100% (16/16), 10.62 MiB | 6.92 MiB/s, done.
(base) oykwon@Ohyoung-K506:~/codes$ cd streamlit-demo/
(base) oykwon@Ohyoung-K506:~/codes/streamlit-demo$ ls
README.md data main.py
(base) oykwon@Ohyoung-K506:~/codes/streamlit-demo$ streamlit run main.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.24.133.115:8501
```



streamlit-demo/main.py at mas

github.com/python-engineer/streamlit-demo/blob/master/main.py

streamlit-demo / main.py / <> Jump to ▾

Drashti125 st.pyplot(fig) here we have to pass a parameter

2 contributors

109 lines (86 sloc) | 2.67 KB

```
1 import streamlit as st
2 import numpy as np
3
4 import matplotlib.pyplot as plt
5 from sklearn import datasets
6 from sklearn.model_selection import train_test_split
7
8 from sklearn.decomposition import PCA
9 from sklearn.svm import SVC
10 from sklearn.neighbors import KNeighborsClassifier
11 from sklearn.ensemble import RandomForestClassifier
12
13 from sklearn.metrics import accuracy_score
14
15 st.title('Streamlit Example')
16
17 st.write("""
18 # Explore different classifier and datasets
19 Which one is the best?
20 """)
21
22 dataset_name = st.sidebar.selectbox(
23     'Select Dataset',
24     ('Iris', 'Breast Cancer', 'Wine')
25 )
26
```

streamlit-demo/main.py at mas

github.com/python-engineer/streamlit-demo/blob/master/main.py

```
27 st.write(f"## {dataset_name} Dataset")
28
29 classifier_name = st.sidebar.selectbox(
30     'Select classifier',
31     ('KNN', 'SVM', 'Random Forest')
32 )
33
34 def get_dataset(name):
35     data = None
36     if name == 'Iris':
37         data = datasets.load_iris()
38     elif name == 'Wine':
39         data = datasets.load_wine()
40     else:
41         data = datasets.load_breast_cancer()
42     X = data.data
43     y = data.target
44     return X, y
45
46 X, y = get_dataset(dataset_name)
47 st.write('Shape of dataset:', X.shape)
48 st.write('number of classes:', len(np.unique(y)))
49
50 def add_parameter_ui(clf_name):
51     params = dict()
52     if clf_name == 'SVM':
53         C = st.sidebar.slider('C', 0.01, 10.0)
54         params['C'] = C
55     elif clf_name == 'KNN':
56         K = st.sidebar.slider('K', 1, 15)
57         params['K'] = K
58     else:
59         max_depth = st.sidebar.slider('max_depth', 2, 15)
60         params['max_depth'] = max_depth
61         n_estimators = st.sidebar.slider('n_estimators', 1, 100)
62         params['n_estimators'] = n_estimators
63
```

```
streamlit-demo/main.py at mas... x +
github.com/python-engineer/streamlit-demo/blob/master/main.py
★ Bookmarks  📁 웹 조각 갤러리  📁 MOOC  🔍 추천 사이트  📁 IE에서 가져온 북마크  Sebastian Seung's...

62     params['n_estimators'] = n_estimators
63     return params
64
65     params = add_parameter_ui(classifier_name)
66
67     def get_classifier(clf_name, params):
68         clf = None
69         if clf_name == 'SVM':
70             clf = SVC(C=params['C'])
71         elif clf_name == 'KNN':
72             clf = KNeighborsClassifier(n_neighbors=params['K'])
73         else:
74             clf = RandomForestClassifier(n_estimators=params['n_estimators'],
75                                         max_depth=params['max_depth'], random_state=1234)
76         return clf
77
78     clf = get_classifier(classifier_name, params)
79     ##### CLASSIFICATION #####
80
81     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234)
82
83     clf.fit(X_train, y_train)
84     y_pred = clf.predict(X_test)
85
86     acc = accuracy_score(y_test, y_pred)
87
88     st.write(f'Classifier = {classifier_name}')
89     st.write(f'Accuracy =', acc)
90
91     ##### PLOT DATASET #####
92     # Project the data onto the 2 primary principal components
93     pca = PCA(2)
94     X_projected = pca.fit_transform(X)
95
96     x1 = X_projected[:, 0]
97     x2 = X_projected[:, 1]
```

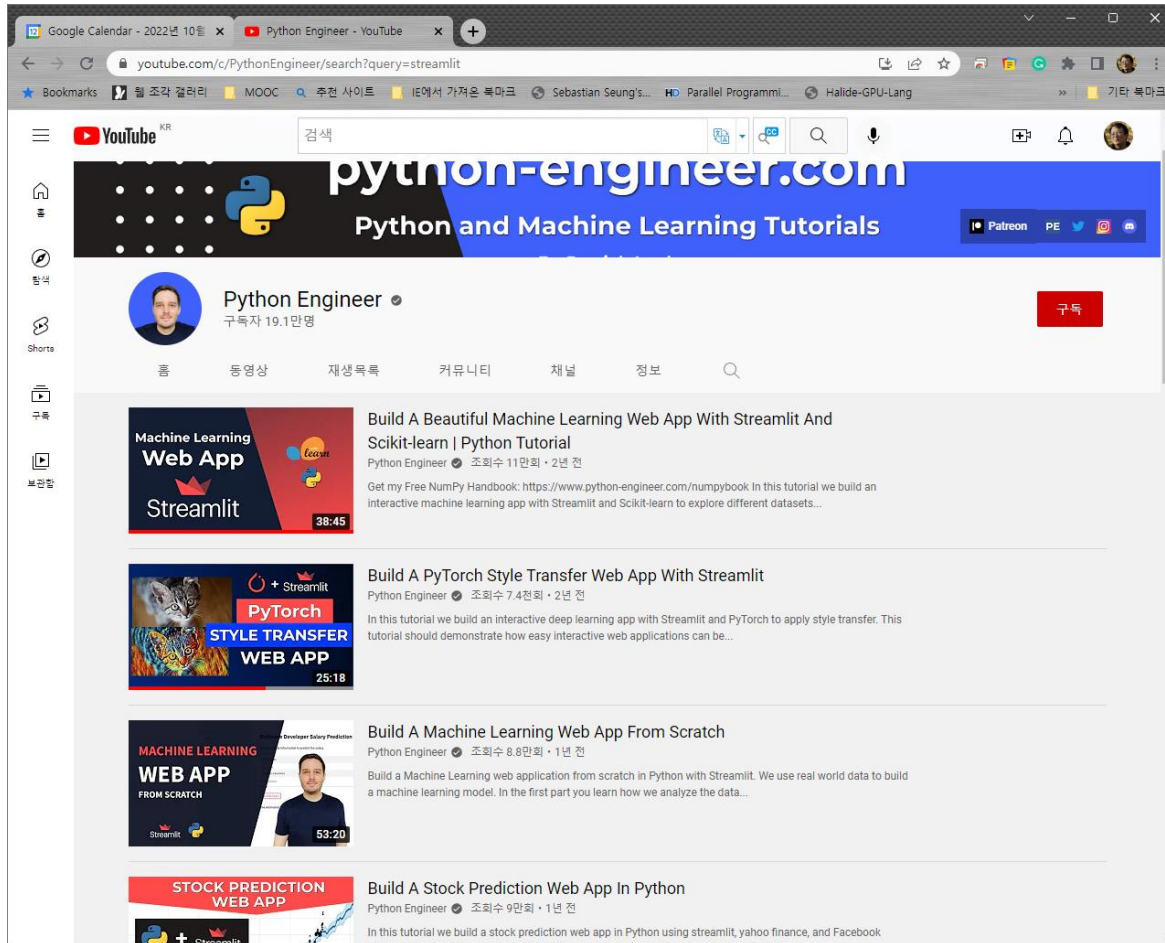
```
streamlit-demo/main.py at mas... x +
github.com/python-engineer/streamlit-demo/blob/master/main.py
★ Bookmarks  📁 웹 조각 갤러리  📁 MOOC  🔍 추천 사이트  📁 IE에서 가져온 북마크  Sebastian Seung's...  기타 북마크

83     clf.fit(X_train, y_train)
84     y_pred = clf.predict(X_test)
85
86     acc = accuracy_score(y_test, y_pred)
87
88     st.write(f'Classifier = {classifier_name}')
89     st.write(f'Accuracy =', acc)
90
91     ##### PLOT DATASET #####
92     # Project the data onto the 2 primary principal components
93     pca = PCA(2)
94     X_projected = pca.fit_transform(X)
95
96     x1 = X_projected[:, 0]
97     x2 = X_projected[:, 1]
98
99     fig = plt.figure()
100     plt.scatter(x1, x2,
101                 c=y, alpha=0.8,
102                 cmap='viridis')
103
104     plt.xlabel('Principal Component 1')
105     plt.ylabel('Principal Component 2')
106     plt.colorbar()
107
108     #plt.show()
109     st.pyplot(fig)

Terms  Privacy  Security  Status  Docs  Contact GitHub  Pricing  API  Training  Blog  About
© 2022 GitHub, Inc.
```

Streamlit 참고사이트

- ❖ 유튜브 Python Engineer
 - streamlit 검색 상위 4개



프로젝트(기말과제)

❖ 인공지능을 활용한 서비스 제안 및 구현

1. 제안서

- 2페이지내외 자유형식 (14주차 개인별 발표; 컨설팅)

(필기숫자인식, 공구의 모양인식 등; 여러분 전공과 연관된 서비스를 자유롭게 정의해보세요.)

2. 서비스 구현보고서

- 구현에 사용한 알고리즘과 제안한 서비스와 최종 구현물 사이에 발생한 차이점 위주로 5페이지이내로 작성

* 데이터는 가능하면 공개된 데이터 활용하고, 아니면 어떻게 데이터를 만들었는지 기술

* 클라우드시스템이 카메라등 외부 입력이 원할하지 않으므로 입력데이터는 외부에서 생성해서 클라우드로 업로드해도 됨

* 컴퓨터공학부학생은 반드시 streamlit을 이용

3. 소스코드

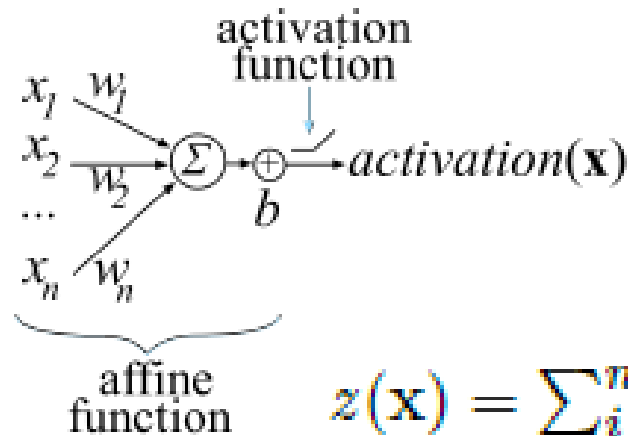
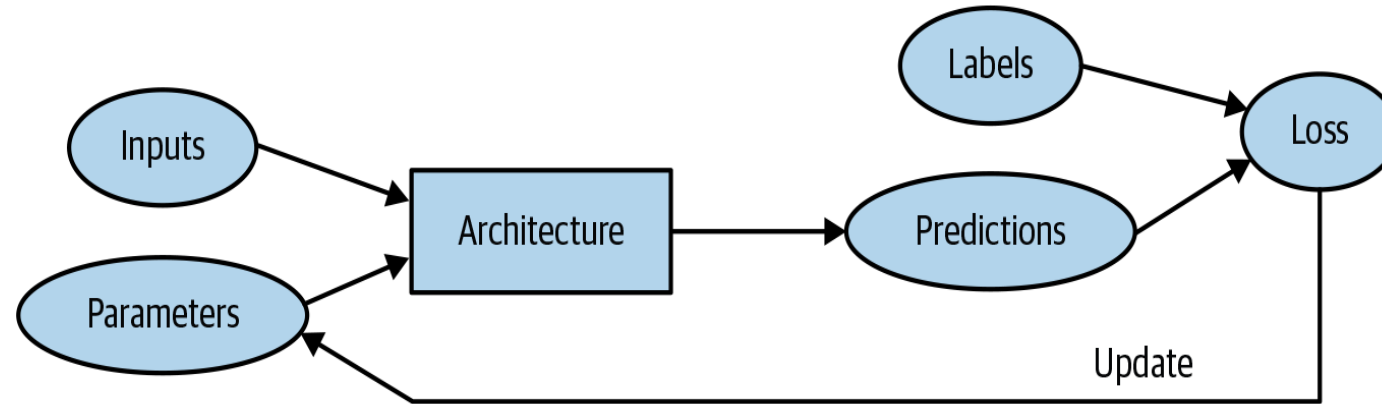
❖ 개인 제출 3분 동영상 형태로 제출 (3분 초과시 감점)

❖ 제출마감일: 12월 19일 15:00

MATRIX CALCULATION

Matrix Calculus

❖ <https://arxiv.org/pdf/1802.01528.pdf>



(mean squared error):

$$\frac{1}{N} \sum_{\mathbf{x}} (\text{target}(\mathbf{x}) - \text{activation}(\mathbf{x}))^2 = \frac{1}{N} \sum_{\mathbf{x}} (\text{target}(\mathbf{x}) - \max(0, \sum_i^{|x|} w_i x_i + b))^2$$

$$z(\mathbf{x}) = \sum_i^n w_i x_i + b = \mathbf{w} \cdot \mathbf{x} + b.$$

미분 정리

Rule	$f(x)$	Scalar derivative notation with respect to x	Example
Constant	c	0	$\frac{d}{dx}99 = 0$
Multiplication by constant	cf	$c\frac{df}{dx}$	$\frac{d}{dx}3x = 3$
Power Rule	x^n	nx^{n-1}	$\frac{d}{dx}x^3 = 3x^2$
Sum Rule	$f + g$	$\frac{df}{dx} + \frac{dg}{dx}$	$\frac{d}{dx}(x^2 + 3x) = 2x + 3$
Difference Rule	$f - g$	$\frac{df}{dx} - \frac{dg}{dx}$	$\frac{d}{dx}(x^2 - 3x) = 2x - 3$
Product Rule	fg	$f\frac{dg}{dx} + \frac{df}{dx}g$	$\frac{d}{dx}x^2x = x^2 + x2x = 3x^2$
Chain Rule	$f(g(x))$	$\frac{df(u)}{du}\frac{du}{dx}$, let $u = g(x)$	$\frac{d}{dx}\ln(x^2) = \frac{1}{x^2}2x = \frac{2}{x}$

$$\frac{d}{dx}9(x + x^2) = 9\frac{d}{dx}(x + x^2) = 9\left(\frac{d}{dx}x + \frac{d}{dx}x^2\right) = 9(1 + 2x) = 9 + 18x$$

미분 정리

❖ $f(x_0 + dx) = f(x_0) + f'(x_0)dx$

```
1 def f(x):
2     return 9*(x+x*x)
3
4 def df(x):
5     return 9+18*x
6
7 print(f(1))
8
9 print('f(1+0.01) = ', f(1+0.01), 'dy = ', df(1)*0.01)
10 print('f(1+0.001) = ', f(1+0.001), 'dy = ', df(1)*0.001)
11 print('f(1+0.0001) = ', f(1+0.0001), 'dy = ', df(1)*0.0001)
```

Shell ×

```
18
f(1+0.01) =  18.2709 dy =  0.27
f(1+0.001) =  18.027008999999993 dy =  0.027
f(1+0.0001) =  18.00270009 dy =  0.0027
```

편미분

❖ 변수가 여럿일 때 각 변수별로 미분

$$f(x, y) = 3x^2y.$$

$$\frac{\partial}{\partial x} 3yx^2 = 3y \frac{\partial}{\partial x} x^2 = 3y 2x = 6yx.$$

$$\frac{\partial}{\partial y} 3x^2y = 3x^2 \frac{\partial}{\partial y} y = 3x^2 \frac{\partial y}{\partial y} = 3x^2 \times 1 = 3x^2.$$

❖ gradient of $f(x, y)$

$$\nabla f(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right] = [6yx, 3x^2]$$

단위 벡터를 곱하면 scalar값이 됨

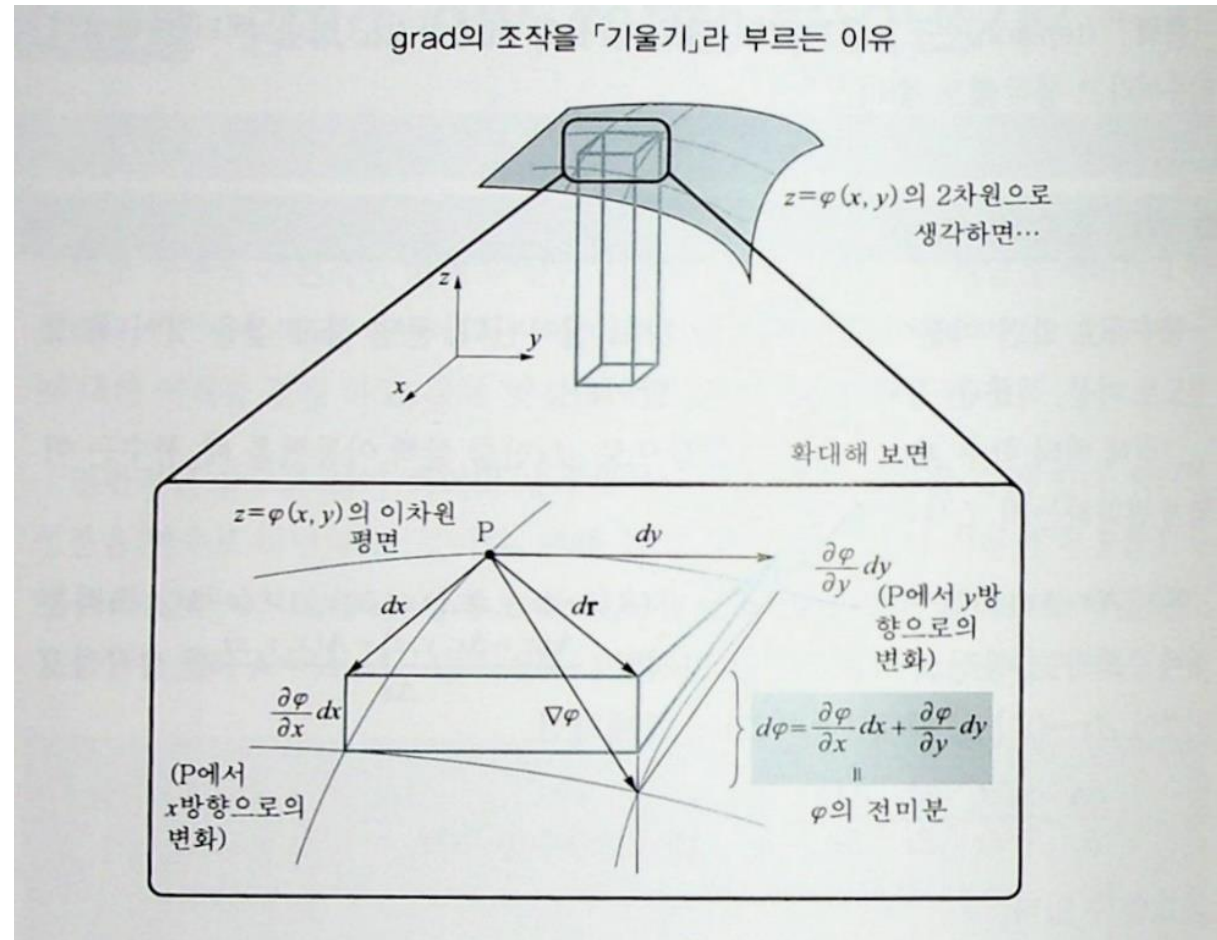
```
1 def f(x,y):
2     return 3*x*x*y
3
4 def dfx(x, y):
5     return 6*y*x
6
7 def dfy(x, y):
8     return 3*x*x
9
10 print(f(1,1))
11
12 print('f(1+0.01,1) = ', f(1+0.01, 1), 'xdf = ', dfx(1,1)*0.01)
13 print('f(1+0.001, 1) = ', f(1+0.001, 1), 'xdf = ', dfx(1,1)*0.001)
14 print('f(1+0.0001, 1) = ', f(1+0.0001, 1), 'xdf = ', dfx(1,1)*0.0001)
15
16 print('f(1, 1+0.01) = ', f(1, 1+0.01), 'ydf = ', dfy(1,1)*0.01)
17 print('f(1, 1+0.001) = ', f(1, 1+0.001), 'ydf = ', dfy(1,1)*0.001)
18 print('f(1, 1+0.0001) = ', f(1, 1+0.0001), 'ydf = ', dfy(1,1)*0.0001)
```

Shell ×

```
3
f(1+0.01,1) = 3.0603000000000002 xdf = 0.06
f(1+0.001, 1) = 3.0060029999999993 xdf = 0.006
f(1+0.0001, 1) = 3.00060003 xdf = 0.00060000000000000001
f(1, 1+0.01) = 3.0300000000000002 ydf = 0.03
f(1, 1+0.001) = 3.0029999999999997 ydf = 0.003
f(1, 1+0.0001) = 3.0003 ydf = 0.00030000000000000003
```

Gradient

❖ 변화율과 그 방향을 보여줌



matrix calculus

- ❖ Jacobian matrix: 다변수 벡터 함수의 편미분함수 행렬
(미시 영역에서 비선형적 변화를 선형변환으로 근사)
- ❖ m 개의 함수, n 개의 변수를 갖는 벡터 (m 개의 gradient 벡터)

$$\begin{array}{rcl} y_1 & = & f_1(\mathbf{x}) \\ y_2 & = & f_2(\mathbf{x}) \\ & \vdots & \\ y_m & = & f_m(\mathbf{x}) \end{array} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \nabla f_1(\mathbf{x}) \\ \nabla f_2(\mathbf{x}) \\ \dots \\ \nabla f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}} f_1(\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{x}} f_2(\mathbf{x}) \\ \dots \\ \frac{\partial}{\partial \mathbf{x}} f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\mathbf{x}) & \frac{\partial}{\partial x_2} f_1(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_1(\mathbf{x}) \\ \frac{\partial}{\partial x_1} f_2(\mathbf{x}) & \frac{\partial}{\partial x_2} f_2(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_2(\mathbf{x}) \\ \dots & \dots & \dots & \dots \\ \frac{\partial}{\partial x_1} f_m(\mathbf{x}) & \frac{\partial}{\partial x_2} f_m(\mathbf{x}) & \dots & \frac{\partial}{\partial x_n} f_m(\mathbf{x}) \end{bmatrix}$$

matrix calculus

❖ Jacobian의 모습

	vector	
	scalar	
	x	\mathbf{x}
scalar	$\frac{\partial f}{\partial x}$	$\frac{\partial f}{\partial \mathbf{x}}$
f		
vector	$\frac{\partial \mathbf{f}}{\partial x}$	$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$
\mathbf{f}		

chain rule

❖ single-variable chain rule -> divide and conquer

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$$

$$y = f(g(x)) = \sin(x^2):$$

$$u = x^2 \quad (\text{relative to definition } f(g(x)), g(x) = x^2)$$

$$y = \sin(u) \quad (y = f(u) = \sin(u))$$

$$\frac{du}{dx} = 2x$$

$$\frac{dy}{du} = \cos(u)$$

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx} = \cos(u)2x$$

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx} = \cos(x^2)2x = 2x\cos(x^2)$$

$$\left. \begin{array}{c} y = \sin \\ \uparrow \\ u = \text{square} \\ \uparrow \\ x \end{array} \right\} \frac{dy}{du} \frac{du}{dx}$$

single-variable chain rule

❖ Forward and backward differentiation

Forward differentiation from x to y Backward differentiation from y to x

$$\frac{dy}{dx} = \frac{du}{dx} \frac{dy}{du}$$

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$$

1. Introduce intermediate variables.

$$y = f(x) = \ln(\sin(x^3)^2):$$

$$\begin{aligned} u_1 &= f_1(x) = x^3 \\ u_2 &= f_2(u_1) = \sin(u_1) \\ u_3 &= f_3(u_2) = u_2^2 \\ u_4 &= f_4(u_3) = \ln(u_3) (y = u_4) \end{aligned}$$

$$\begin{array}{c} y = r_4 \\ r_3 \\ r_2 \\ r_1 \end{array} \begin{array}{c} \ln \\ \uparrow \\ \text{square} \\ \uparrow \\ \sin \\ \uparrow \\ \text{cube} \\ \uparrow \\ x \end{array} \left\{ \begin{array}{l} \frac{du_4}{du_3} \\ \frac{du_3}{du_2} \\ \frac{du_2}{du_1} \\ \frac{du_1}{du_1} \end{array} \right\} \frac{du_4}{dx}$$

2. Compute derivatives.

$$\begin{aligned} \frac{d}{u_1} u_1 &= \frac{d}{x} x^3 = 3x^2 \\ \frac{d}{u_1} u_2 &= \frac{d}{u_1} \sin(u_1) = \cos(u_1) \\ \frac{d}{u_2} u_3 &= \frac{d}{u_2} u_2^2 = 2u_2 \\ \frac{d}{u_3} u_4 &= \frac{d}{u_3} \ln(u_3) = \frac{1}{u_3} \end{aligned}$$

3. Combine four intermediate values.

$$\frac{dy}{dx} = \frac{du_4}{dx} = \frac{du_4}{du_3} \frac{du_3}{du_2} \frac{du_2}{du_1} \frac{du_1}{dx} = \frac{1}{u_3} 2u_2 \cos(u_1) 3x^2 = \frac{6u_2 x^2 \cos(u_1)}{u_3}$$

4. Substitute.

$$\frac{dy}{dx} = \frac{6\sin(u_1)x^2\cos(x^3)}{u_2^2} = \frac{6\sin(x^3)x^2\cos(x^3)}{\sin(u_1)^2} = \frac{6\sin(x^3)x^2\cos(x^3)}{\sin(x^3)^2} = \frac{6x^2\cos(x^3)}{\sin(x^3)}$$

single-variable total-derivative chain rule

❖ 일반적인 수식으로 확장 예) $y = f(x) = x + x^2$

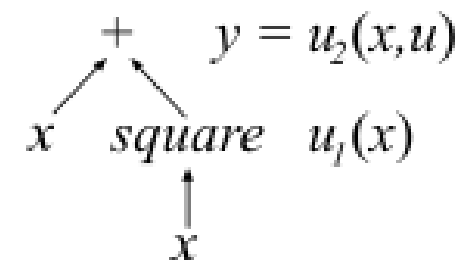
$$\frac{dy}{dx} = \frac{d}{dx}x + \frac{d}{dx}x^2 = 1 + 2x$$

$$\begin{aligned} u_1(x) &= x^2 \\ u_2(x, u_1) &= x + u_1 \quad (y = f(x) = u_2(x, u_1)) \end{aligned}$$

Let's try it anyway to see what happens. If we pretend that $\frac{du_2}{du_1} = 0 + 1 = 1$ and $\frac{du_1}{dx} = 2x$, then $\frac{dy}{dx} = \frac{du_2}{dx} = \frac{du_2}{du_1} \frac{du_1}{dx} = 2x$ instead of the right answer $1 + 2x$.

$$\begin{aligned} \frac{\partial u_1(x)}{\partial x} &= 2x && \text{(same as } \frac{du_1(x)}{dx} \text{)} \\ \frac{\partial u_2(x, u_1)}{\partial u_1} &= \frac{\partial}{\partial u_1}(x + u_1) = 0 + 1 = 1 \\ \frac{\partial u_2(x, u_1)}{\partial x} &\neq \frac{\partial}{\partial x}(x + u_1) = 1 + 0 = 1 && \text{(something's not quite right here!)} \end{aligned}$$

❖ x에 대한 편미분을 할 때 라는 변수는 x에 영향을 받지 않아야 한다.
(u1이 x에 영향을 받음 -> 위 편미분의 가정 위배)



single-variable total-derivative chain rule

❖ total derivative

$$y = f(x) = x + x^2$$

$$\frac{dy}{dx} = \frac{\partial f(x)}{\partial x} = \frac{\partial u_2(x, u_1)}{\partial x} = \frac{\partial u_2}{\partial x} \frac{\partial x}{\partial x} + \frac{\partial u_2}{\partial u_1} \frac{\partial u_1}{\partial x} = \frac{\partial u_2}{\partial x} + \frac{\partial u_2}{\partial u_1} \frac{\partial u_1}{\partial x}$$

$$\frac{dy}{dx} = \frac{\partial f(x)}{\partial x} = \frac{\partial u_2}{\partial x} + \frac{\partial u_2}{\partial u_1} \frac{\partial u_1}{\partial x} = 1 + 1 \times 2x = 1 + 2x$$

$$\frac{\partial f(x, u_1, \dots, u_n)}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u_1} \frac{\partial u_1}{\partial x} + \frac{\partial f}{\partial u_2} \frac{\partial u_2}{\partial x} + \dots + \frac{\partial f}{\partial u_n} \frac{\partial u_n}{\partial x} = \frac{\partial f}{\partial x} + \sum_{i=1}^n \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x}$$

single-variable total-derivative chain rule

❖ total derivative $f(x) = \sin(x + x^2)$

$$\begin{aligned} u_1(x) &= x^2 \\ u_2(x, u_1) &= x + u_1 \\ u_3(u_2) &= \sin(u_2) \quad (y = f(x) = u_3(u_2)) \end{aligned}$$

and partials:

$$\begin{aligned} \frac{\partial u_1}{\partial x} &= 2x \\ \frac{\partial u_2}{\partial x} &= \frac{\partial x}{\partial x} + \frac{\partial u_1}{\partial x} = 1 + 1 \times 2x = 1 + 2x \\ \frac{\partial f(x)}{\partial x} &= \frac{\partial u_3}{\partial x} + \frac{\partial u_3}{\partial u_2} \frac{\partial u_2}{\partial x} = 0 + \cos(u_2) \frac{\partial u_2}{\partial x} = \cos(x + x^2)(1 + 2x) \end{aligned}$$

❖ 정리

$$\frac{\partial f(x, u_1, \dots, u_n)}{\partial x} = \frac{\partial f}{\partial x} + \sum_{i=1}^n \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x} \rightarrow \frac{\partial f(u_1, \dots, u_{n+1})}{\partial x} = \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x} \quad u_{n+1} = x.$$

vector chain rule

❖ vector function

$$\mathbf{y} = \mathbf{f}(\mathbf{x}): \quad \begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} \ln(x^2) \\ \sin(3x) \end{bmatrix}$$

$$\mathbf{y} = \mathbf{f}(\mathbf{g}(x)):$$

$$\begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

$$\begin{bmatrix} f_1(\mathbf{g}) \\ f_2(\mathbf{g}) \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial f_1(\mathbf{g})}{\partial x} \\ \frac{\partial f_2(\mathbf{g})}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} \frac{\partial g_1}{\partial x} + \frac{\partial f_1}{\partial g_2} \frac{\partial g_2}{\partial x} \\ \frac{\partial f_2}{\partial g_1} \frac{\partial g_1}{\partial x} + \frac{\partial f_2}{\partial g_2} \frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + \cos(g_2) 3 \end{bmatrix} = \begin{bmatrix} \frac{2x}{x^2} \\ 3\cos(3x) \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

vector chain rule

❖ vector function

$$\begin{bmatrix} \frac{\partial f_1}{\partial g_1} \frac{\partial g_1}{\partial x} + \frac{\partial f_1}{\partial g_2} \frac{\partial g_2}{\partial x} \\ \frac{\partial f_2}{\partial g_1} \frac{\partial g_1}{\partial x} + \frac{\partial f_2}{\partial g_2} \frac{\partial g_2}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} \\ \frac{\partial f_2}{\partial g_1} & \frac{\partial f_2}{\partial g_2} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x} \\ \frac{\partial g_2}{\partial x} \end{bmatrix} = \frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial x}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial x} = \begin{bmatrix} \frac{1}{g_1} & 0 \\ 0 & \cos(g_2) \end{bmatrix} \begin{bmatrix} 2x \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + \cos(g_2) 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

$$\frac{\partial}{\partial x} \mathbf{f}(\mathbf{g}(x)) = \frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial x} \qquad \frac{d}{dx} f(g(x)) = \frac{df}{dg} \frac{dg}{dx}$$

vector chain rule

❖ vector function

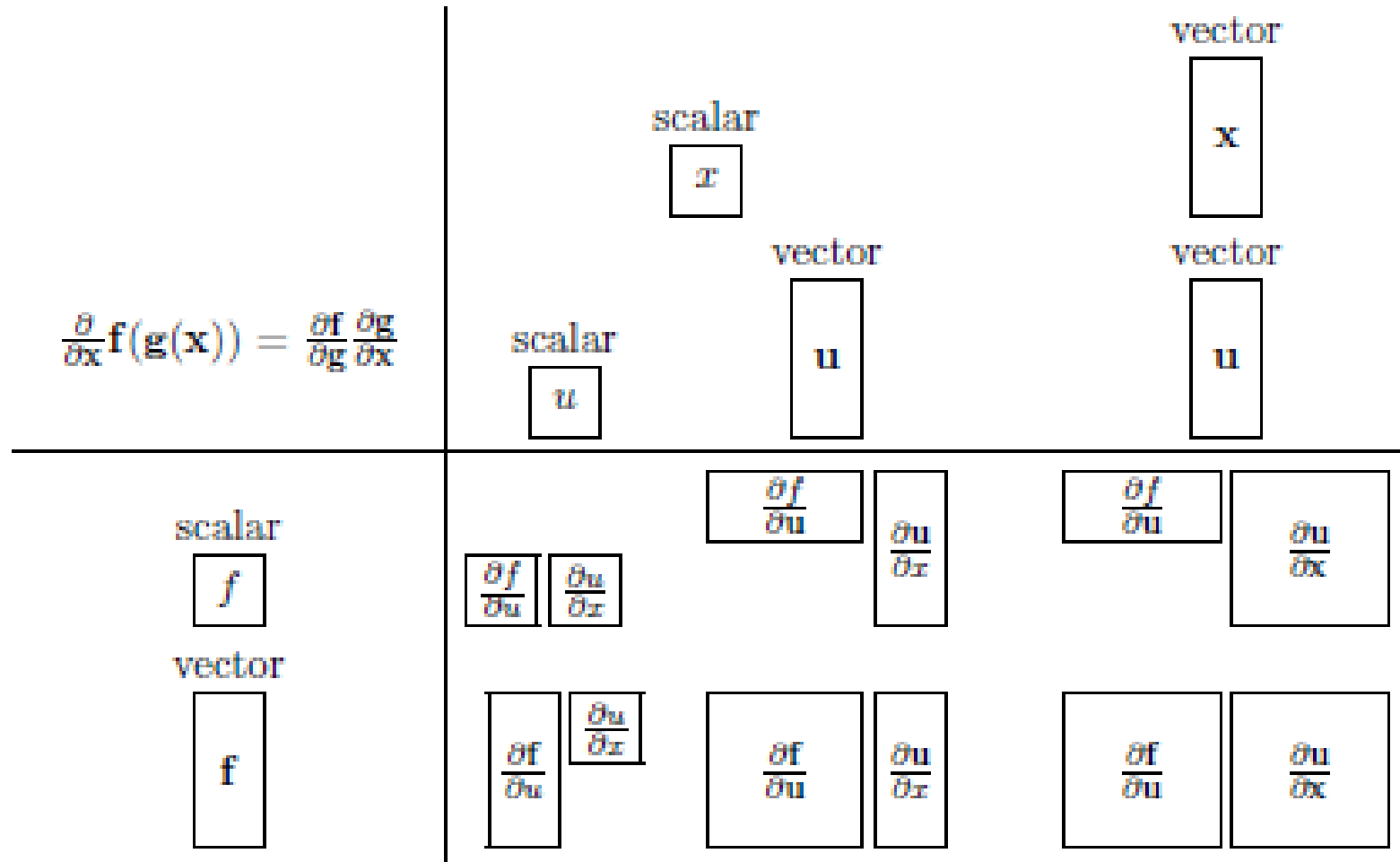
$$\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{g}(\mathbf{x})) = \frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{g}(\mathbf{x})) = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} & \cdots & \frac{\partial f_1}{\partial g_k} \\ \frac{\partial f_2}{\partial g_1} & \frac{\partial f_2}{\partial g_2} & \cdots & \frac{\partial f_2}{\partial g_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial g_1} & \frac{\partial f_m}{\partial g_2} & \cdots & \frac{\partial f_m}{\partial g_k} \end{bmatrix} \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_k}{\partial x_1} & \frac{\partial g_k}{\partial x_2} & \cdots & \frac{\partial g_k}{\partial x_n} \end{bmatrix}$$

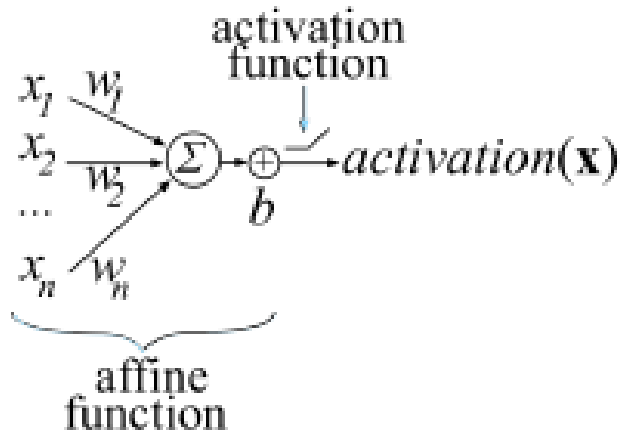
$$\frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{g}(\mathbf{x})) = \text{diag}\left(\frac{\partial f_i}{\partial g_i}\right) \text{diag}\left(\frac{\partial g_i}{\partial x_i}\right) = \text{diag}\left(\frac{\partial f_i}{\partial g_i} \frac{\partial g_i}{\partial x_i}\right)$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial x} = \begin{bmatrix} \frac{1}{g_1} & 0 \\ 0 & \cos(g_2) \end{bmatrix} \begin{bmatrix} 2x \\ 3 \end{bmatrix}$$

vector chain rule



The gradient of neuron activation



$$activation(\mathbf{x}) = \max(0, \mathbf{w} \cdot \mathbf{x} + b)$$

$$z(\mathbf{x}) = \sum_i^n w_i x_i + b = \mathbf{w} \cdot \mathbf{x} + b.$$

w와 b 값을 조정

$$\frac{\partial}{\partial \mathbf{w}}(\mathbf{w} \cdot \mathbf{x} + b) \text{ and } \frac{\partial}{\partial b}(\mathbf{w} \cdot \mathbf{x} + b).$$

$$d(u^T v) = du^T v + u^T dv = v^T du + u^T dv$$

$$\begin{aligned} d(x^T x) &= dx^T x + x^T dx = x^T dx + x^T dx \\ &= (2x)^T dx \end{aligned}$$

$$\mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T \mathbf{x}.$$

The gradient of neuron activation

$$\frac{\partial}{\partial \mathbf{w}}(\mathbf{w} \cdot \mathbf{x} + b) \text{ and } \frac{\partial}{\partial b}(\mathbf{w} \cdot \mathbf{x} + b), \quad \mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T \mathbf{x}.$$

$$\begin{aligned} y &= \mathbf{w} \cdot \mathbf{x} = \sum_i^n (w_i x_i) \\ \frac{\partial y}{\partial w_j} &= \frac{\partial}{\partial w_j} \sum_i (w_i x_i) = \sum_i \frac{\partial}{\partial w_j} (w_i x_i) = \frac{\partial}{\partial w_j} (w_j x_j) = x_j \end{aligned}$$

$$\begin{aligned} \frac{\partial y}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}} \mathbf{w} \cdot \mathbf{x} + \frac{\partial}{\partial \mathbf{w}} b = \mathbf{x}^T + \vec{0}^T = \mathbf{x}^T \\ \frac{\partial y}{\partial b} &= \frac{\partial}{\partial b} \mathbf{w} \cdot \mathbf{x} + \frac{\partial}{\partial b} b = 0 + 1 = 1 \end{aligned}$$

The gradient of neuron activation

$$z(\mathbf{w}, b, \mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

$$\text{activation}(z) = \max(0, z)$$

$$\frac{\partial \text{activation}}{\partial \mathbf{w}} = \frac{\partial \text{activation}}{\partial z} \frac{\partial z}{\partial \mathbf{w}}$$

$$\frac{\partial \text{activation}}{\partial \mathbf{w}} = \begin{cases} 0 \frac{\partial z}{\partial \mathbf{w}} = \vec{0}^T & z \leq 0 \\ 1 \frac{\partial z}{\partial \mathbf{w}} = \frac{\partial z}{\partial \mathbf{w}} = \mathbf{x}^T & z > 0 \end{cases} \quad (\text{we computed } \frac{\partial z}{\partial \mathbf{w}} = \mathbf{x}^T \text{ previously})$$

and then substitute $z = \mathbf{w} \cdot \mathbf{x} + b$ back in:

$$\frac{\partial \text{activation}}{\partial \mathbf{w}} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$\frac{\partial \text{activation}}{\partial b} = \begin{cases} 0 \frac{\partial z}{\partial b} = 0 & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1 \frac{\partial z}{\partial b} = 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

The gradient of neural network loss function

multiple vector inputs (multiple images) $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$

scalar targets (분류결과)

$$\mathbf{y} = [\text{target}(\mathbf{x}_1), \text{target}(\mathbf{x}_2), \dots, \text{target}(\mathbf{x}_N)]^T$$

비용함수 -> 원하는 결과(target)과 신경망의 결과(activation)의 차이

$$C(\mathbf{w}, b, X, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \text{activation}(\mathbf{x}_i))^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \max(0, \mathbf{w} \cdot \mathbf{x}_i + b))^2$$

비용함수가 최소가 되게 \mathbf{w} 와 b 를 조정해야됨 -> 미분을 수행 -> gradient를 구해서 \mathbf{w} 를 조정하는 과정

$$u(\mathbf{w}, b, \mathbf{x}) = \max(0, \mathbf{w} \cdot \mathbf{x} + b)$$

$$v(y, u) = y - u$$

$$C(v) = \frac{1}{N} \sum_{i=1}^N v^2$$

The gradient with respect to the weights

neuron의 결과

$$\frac{\partial}{\partial \mathbf{w}} u(\mathbf{w}, b, \mathbf{x}) = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$u(\mathbf{w}, b, \mathbf{x}) = \max(0, \mathbf{w} \cdot \mathbf{x} + b)$$

$$v(y, u) = y - u$$

$$C(v) = \frac{1}{N} \sum_{i=1}^N v^2$$

$$\frac{\partial v(y, u)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} (y - u) = \vec{0}^T - \frac{\partial u}{\partial \mathbf{w}} = -\frac{\partial u}{\partial \mathbf{w}} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ -\mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$\frac{\partial C(v)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \frac{1}{N} \sum_{i=1}^N v^2$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \mathbf{w}} v^2 = \frac{1}{N} \sum_{i=1}^N 2v \frac{\partial v}{\partial \mathbf{w}}$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{\partial v^2}{\partial v} \frac{\partial v}{\partial \mathbf{w}} = \frac{1}{N} \sum_{i=1}^N \begin{cases} 2v \vec{0}^T = \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ -2v \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases}$$

The gradient with respect to the weights

neuron의 결과

$$\frac{\partial}{\partial \mathbf{w}} u(\mathbf{w}, b, \mathbf{x}) = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$u(\mathbf{w}, b, \mathbf{x}) = \max(0, \mathbf{w} \cdot \mathbf{x} + b)$$

$$v(y, u) = y - u$$

$$C(v) = \frac{1}{N} \sum_{i=1}^N v^2$$

$$\frac{\partial v(y, u)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} (y - u) = \vec{0}^T - \frac{\partial u}{\partial \mathbf{w}} = -\frac{\partial u}{\partial \mathbf{w}} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ -\mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$\begin{aligned} \frac{\partial C(v)}{\partial \mathbf{w}} &= \frac{1}{N} \sum_{i=1}^N \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ -2(y_i - u)\mathbf{x}_i^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases} \\ &= \frac{1}{N} \sum_{i=1}^N \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ -2(y_i - \max(0, \mathbf{w} \cdot \mathbf{x}_i + b))\mathbf{x}_i^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ \frac{-2}{N} \sum_{i=1}^N (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))\mathbf{x}_i^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases} \\ &= \frac{1}{N} \sum_{i=1}^N \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ -2(y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))\mathbf{x}_i^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ \frac{2}{N} \sum_{i=1}^N (\mathbf{w} \cdot \mathbf{x}_i + b - y_i)\mathbf{x}_i^T & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases} \end{aligned}$$

The gradient with respect to the weights

$$\frac{\partial C}{\partial \mathbf{w}} = \frac{2}{N} \sum_{i=1}^N e_i \mathbf{x}_i^T \quad (\text{for the nonzero activation case})$$

If the error is 0, then the gradient is zero and we have arrived at the minimum loss.

If error is some small positive difference, the gradient is a small step in the direction of \mathbf{x} .

If error is large, the gradient is a large step in that direction.

If error is negative, the gradient is reversed, meaning the highest cost is in the negative direction.

결과값과 올바른값의 차이를 최소화하도록 w 값을 재조정
학습율(이타, eta) \rightarrow 경험적으로 조정

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial C}{\partial \mathbf{w}}$$

The gradient with respect to the bias

$$\frac{\partial u}{\partial b} = \begin{cases} 0 & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases} \quad \begin{aligned} u(\mathbf{w}, b, \mathbf{x}) &= \max(0, \mathbf{w} \cdot \mathbf{x} + b) \\ v(y, u) &= y - u \\ C(v) &= \frac{1}{N} \sum_{i=1}^N v^2 \end{aligned}$$

$$\frac{\partial v(y, u)}{\partial b} = \frac{\partial}{\partial b}(y - u) = 0 - \frac{\partial u}{\partial b} = -\frac{\partial u}{\partial b} = \begin{cases} 0 & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ -1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

$$\frac{\partial C(v)}{\partial b} = \frac{\partial}{\partial b} \frac{1}{N} \sum_{i=1}^N v^2 = \begin{cases} 0 & \mathbf{w} \cdot \mathbf{x}_i + b \leq 0 \\ \frac{2}{N} \sum_{i=1}^N (\mathbf{w} \cdot \mathbf{x}_i + b - y_i) & \mathbf{w} \cdot \mathbf{x}_i + b > 0 \end{cases}$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial b} v^2$$

$$= \frac{1}{N} \sum_{i=1}^N \frac{\partial v^2}{\partial v} \frac{\partial v}{\partial b}$$

$$b_{t+1} = b_t - \eta \frac{\partial C}{\partial b}$$