

“**클라우드를 있는 무한 *IT* 시대!**



클라우드 컴퓨팅과 AI서비스 (12주차)

융합학과 권오영

oykwon@koreatech.ac.kr

학습내용

- ❖ 인공신경망
 - CNN
 - LSTM
- ❖ AI에 사용되는 패키지
- ❖ streamlit 활용

CNN응용

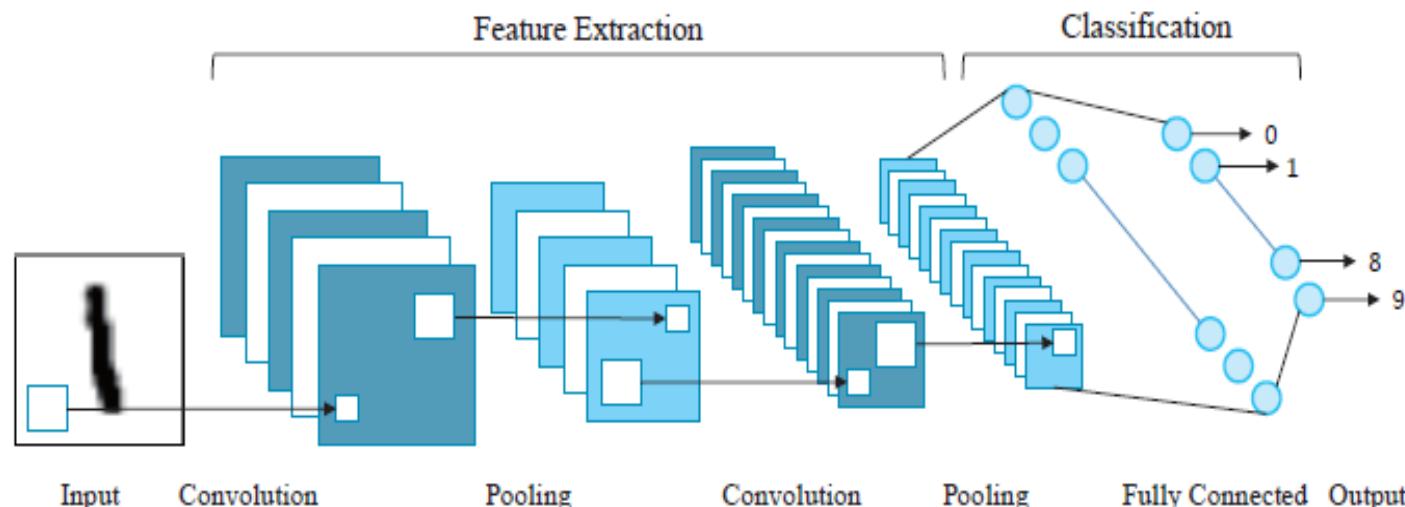
합성곱 신경망(CNN) 구현

❖ 합성곱 신경망 이란?

- 1998년 Yann Lecun이 처음 제안한 알고리즘
- 페이스북의 자동 사진 태그, Google과 네이버의 이미지 검색, 아마존의 제품 추천, 카카오의 형태소 분석기 등

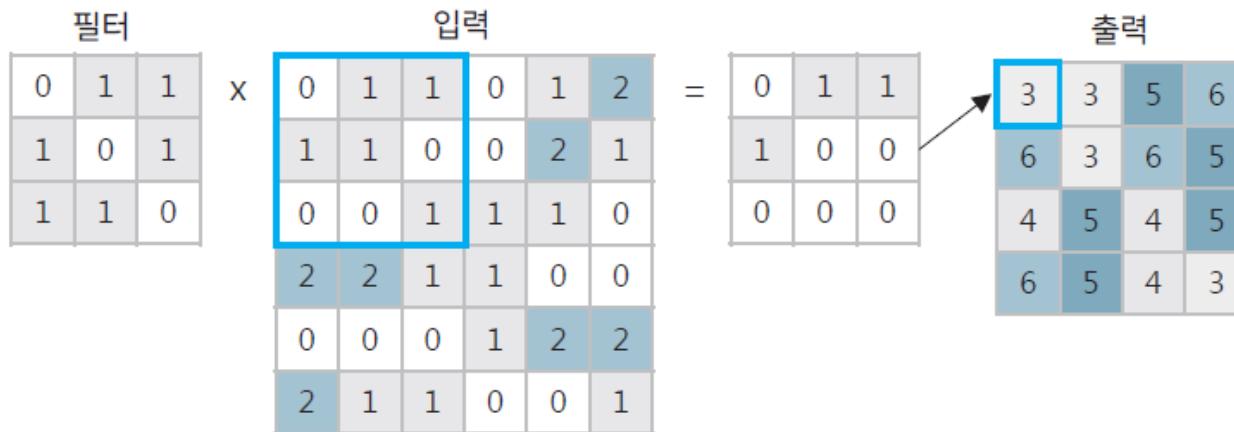
❖ 합성곱 신경망 구조

- 합성곱 계층, 풀링 계층, 완전 결합 계층으로 구성



합성곱 신경망(CNN) 구현

- 합성곱 계층
 - 특징을 추출하기 위한 필터(filter) / 커널(kernel)
 - 이미지의 행렬을 합성곱 하여 특성 맵(feature map) 구성



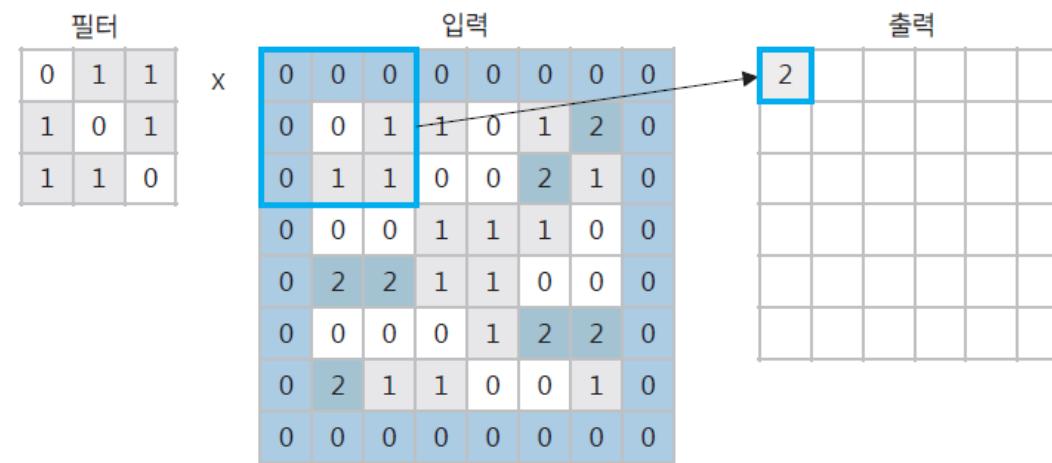
- ✓ 필터와 이미지의 각 위치에 있는 값을 곱하고 모든 행렬의 값을 더하여 구성
- 스트라이드(stride) : 옆으로 이동하며 동일한 연산을 계속 진행
 - ✓ 스트라이드의 크기에 따라 출력값의 크기 변경
 - 이미지의 크기가 6×6 , 필터가 3×3 으로 구성
 - 스트라이드 값이 1이면 출력 이미지 크기는 4×4 로 구성
- 활성화 함수로는 ReLU를 주로 사용

합성곱 신경망(CNN) 구현

- 합성곱 계층

- 제로패딩(Zero Padding)

- ✓ 필터의 크기와 스트라이드 값에 따라 출력 이미지 크기가 줄어드는 것 방지
 - ✓ 입력 이미지의 행렬의 상, 하, 좌, 우에 0을 채움

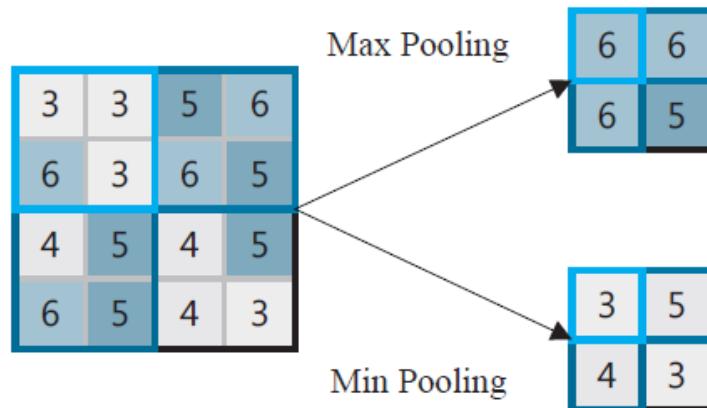


- 입력 이미지 6×6 크기에 제로 패딩을 사용하여 8×8 로 구성
- 3×3 크기의 필터를 한 칸씩 스트라이드
- 출력 이미지의 크기는 입력 이미지의 크기와 동일하게 구성됨

합성곱 신경망(CNN) 구현

- 풀링 계층

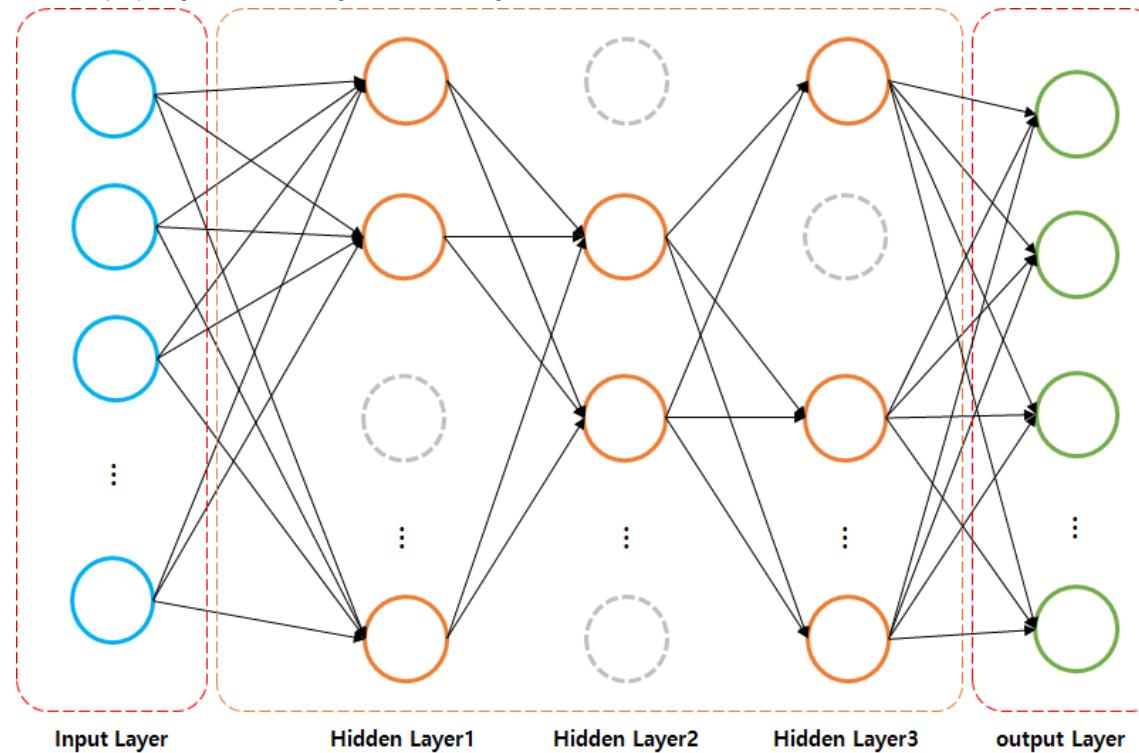
- 선택된 영역에서의 최솟값(Min Pooling), 최댓값(Max Pooling), 평균값(Average Pooling)을 풀링하여 이미지를 축소 처리
 - ✓ 차원을 축소함에 따라 연산량 감소
 - ✓ 과적합(Overfitting)을 방지
 - ✓ 영역 내에서의 특징을 가진 부분을 추출
- 4×4 크기의 입력 이미지를 2×2 크기의 필터와 스트라이드 값을 2로 설정



- 합성곱 신경망에서는 주로 최댓값 풀링을 사용

합성곱 신경망(CNN) 구현

- 드롭 아웃(DropOut)
 - 완전 결합 계층에서의 과적합(Overfitting)을 방지
 - 신경망에서의 뉴런들을 임의적으로 선택하여 버린 후 나머지 뉴런들에 대해서만 학습
 - 학습 시에는 드롭아웃을 사용하고, 학습 이후 검증 시에는 모든 뉴런들을 사용하도록 드롭아웃을 사용하지 않는 것이 일반적인 방식



응용: 딥러닝을 활용한 말라리아 판별

Thonny - D:\mycodes\malaria\malaria.py @ 140 : 1

File Edit View Run Device Tools Help

malaria.py *

```
123 inp = tf.keras.layers.Input(shape=INPUT_SHAPE)
124
125 conv1 = tf.keras.layers.Conv2D(32, kernel_size=(3, 3), activation='relu', padding='same')(inp)
126 pool1 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv1)
127 conv2 = tf.keras.layers.Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same')(pool1)
128 pool2 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv2)
129 conv3 = tf.keras.layers.Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same')(pool2)
130 pool3 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(conv3)
131
132 flat = tf.keras.layers.Flatten()(pool3)
133
134 hidden1 = tf.keras.layers.Dense(512, activation='relu')(flat)
135 drop1 = tf.keras.layers.Dropout(rate=0.3)(hidden1)
136 hidden2 = tf.keras.layers.Dense(512, activation='relu')(drop1)
137 drop2 = tf.keras.layers.Dropout(rate=0.3)(hidden2)
138
139 out = tf.keras.layers.Dense(1, activation='sigmoid')(drop2)
140
141 model = tf.keras.Model(inputs=inp, outputs=out)
142 model.compile(optimizer='adam',
143                 loss='binary_crossentropy',
144                 metrics=['accuracy'])
145
146 model.summary()
```

Shell

```
Epoch 14/25
21/40 [=====>.....] - ETA: 18s - loss: 0.6931 - accuracy: 0.5112
```

학습전이

❖ 잘 개발된 모델의 활용

<https://opensource.com/article/19/4/detecting-malaria-deep-learning>

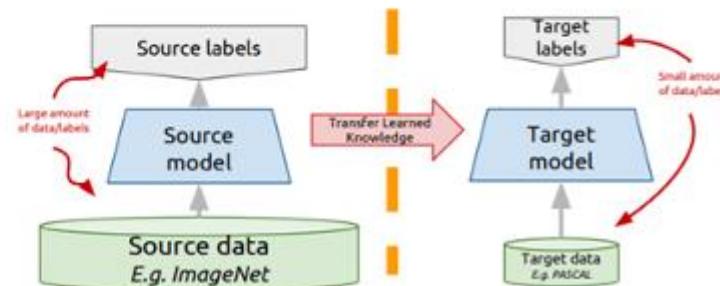
Transfer learning: idea

Instead of training a deep network from scratch for your task:

- Take a network trained on a different domain for a different **source task**
- Adapt it for your domain and your **target task**

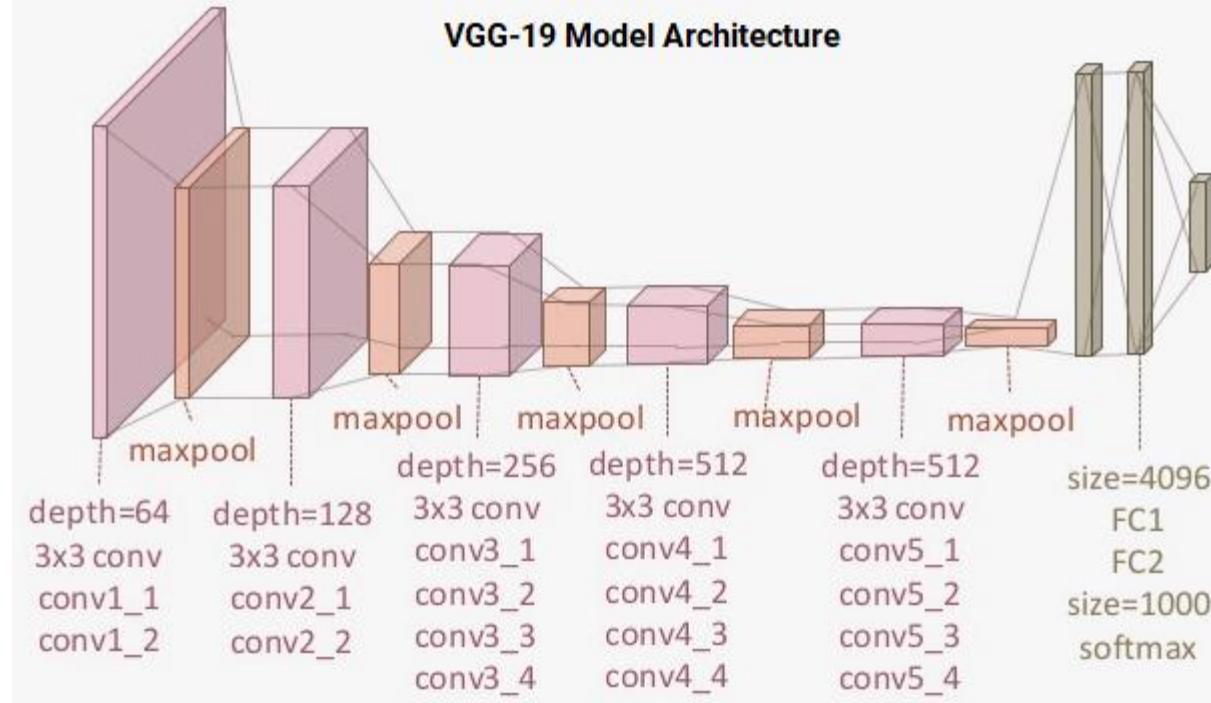
Variations:

- Same domain, different task
- Different domain, same task



학습전이

- ❖ keras내에 유명한 모델들이 많이 구현되어 있음
- ❖ VGG-19 모델



학습전이

❖ 변경없이 사용

```
vgg = tf.keras.applications.vgg19.VGG19(include_top=False, weights='imagenet',
                                         input_shape=INPUT_SHAPE)

vgg.trainable = False
# Freeze the layers
for layer in vgg.layers:
    layer.trainable = False

base_vgg = vgg
base_out = base_vgg.output
pool_out = tf.keras.layers.Flatten()(base_out)
hidden1 = tf.keras.layers.Dense(512, activation='relu')(pool_out)
drop1 = tf.keras.layers.Dropout(rate=0.3)(hidden1)
hidden2 = tf.keras.layers.Dense(512, activation='relu')(drop1)
drop2 = tf.keras.layers.Dropout(rate=0.3)(hidden2)

out = tf.keras.layers.Dense(1, activation='sigmoid')(drop2)

model = tf.keras.Model(inputs=base_vgg.input, outputs=out)
model.compile(optimizer=tf.keras.optimizers.RMSprop(lr=1e-4),
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.summary()
```

학습전이

❖ 모델조정

```
vgg = tf.keras.applications.VGG19(include_top=False, weights='imagenet',
                                   input_shape=INPUT_SHAPE)

# Freeze the layers
vgg.trainable = True

set_trainable = False
for layer in vgg.layers:
    if layer.name in ['block5_conv1', 'block4_conv1']:
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False

base_vgg = vgg
base_out = base_vgg.output
pool_out = tf.keras.layers.Flatten()(base_out)
hidden1 = tf.keras.layers.Dense(512, activation='relu')(pool_out)
drop1 = tf.keras.layers.Dropout(rate=0.3)(hidden1)
hidden2 = tf.keras.layers.Dense(512, activation='relu')(drop1)
drop2 = tf.keras.layers.Dropout(rate=0.3)(hidden2)

out = tf.keras.layers.Dense(1, activation='sigmoid')(drop2)

model = tf.keras.Model(inputs=base_vgg.input, outputs=out)
model.compile(optimizer=tf.keras.optimizers.RMSprop(lr=1e-5),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

모델의 활용

❖ 모델 저장

```
model.save('vgg_finetuned.h5')
```

❖ 저장된 모델의 이용

```
# Load Saved Deep Learning Models
basic_cnn = tf.keras.models.load_model('./basic_cnn.h5')
vgg_frz = tf.keras.models.load_model('./vgg_frozen.h5')
vgg_ft = tf.keras.models.load_model('./vgg_finetuned.h5')

# Make Predictions on Test Data
basic_cnn_preds = basic_cnn.predict(test_imgs_scaled, batch_size=512)
vgg_frz_preds = vgg_frz.predict(test_imgs_scaled, batch_size=512)
vgg_ft_preds = vgg_ft.predict(test_imgs_scaled, batch_size=512)
```

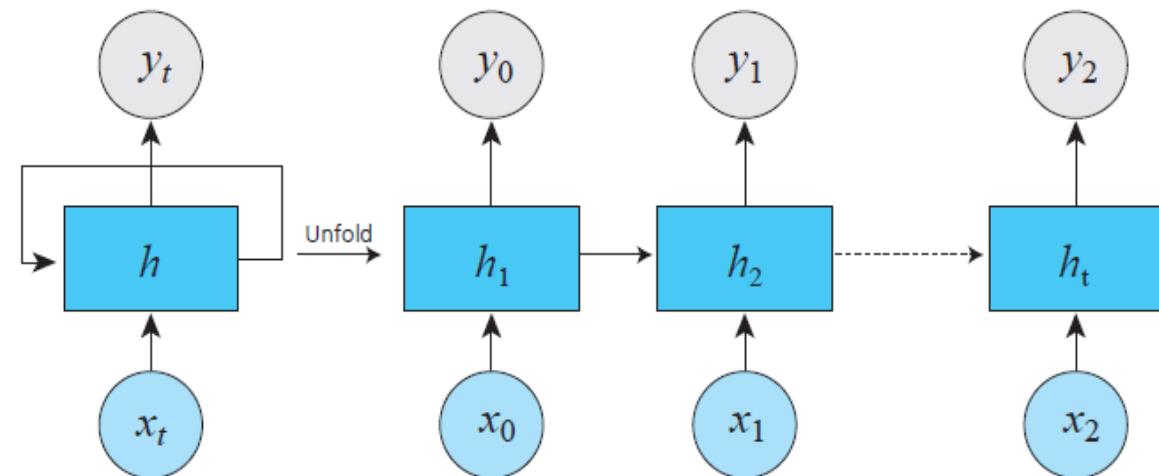
❖ cell_images의 이미지들과 malaria.py 를 확장해보세요.

장단기 기억 네트워크 (LONG-SHORT TERM MEMORY NETWORK)

장단기 기억 네트워크(LSTM) 구현

❖ 순환신경망(Recurrent Neural Network, RNN) 이란?

- 문서 감정 분류, 필기체 인식, 음성 인식과 같은 자연어 처리
- 주가 등 시간을 중심으로 앞, 뒤의 내용이 연관 관계가 있는 시계열 데이터를 처리에 좋은 성능
- 시간 스텝 t 에서의 입력값 x_t , 출력값 y_t 와 h 인 은닉층이 존재
 - ✓ 은닉층의 출력이 다음 시간 스텝에서의 은닉층으로 입력되는 구조가 반복되는 형태
 - ✓ 하나의 네트워크 구조가 여러 개가 연결되어 다음 단계로의 정보를 전달
 - ✓ 메모리 셀(memory cell) : 이전 정보를 은닉층에서 일시적으로 메모리(memory) 형태로 기억
 - ✓ 은닉 상태(hidden state) : 메모리 셀의 상태
 - ✓ 은닉 상태값은 현재 입력값과 이전의 은닉 상태의 값을 가중치를 곱하고 편향을 더함
 - ✓ 활성화 함수로 하이퍼볼릭 탄젠트(tanh) 함수



장단기 기억 네트워크(LSTM) 구현

❖ 순환신경망(Recurrent Neural Network, RNN) 이란?

- 순환 신경망의 학습에는 경사 하강법을 이용하며 출력에서의 경사가 현재 시간에만 의존하는 게 아니라 이전 시간 스텝에도 의존
- 시간 기반 역전파(BackPropagation Through Time, BPTT)라는 변형된 알고리즘으로 가중치를 업데이트
- 경사도 사라짐 문제(Gradient Vanishing Problem)
 - ✓ 시간을 많이 거슬러 올라가게 되면 신경망이 곱하기 연산으로 되어 있기 때문에 역전파에서의 경사가 점점 줄어들어 학습 능력이 저하하는 단점

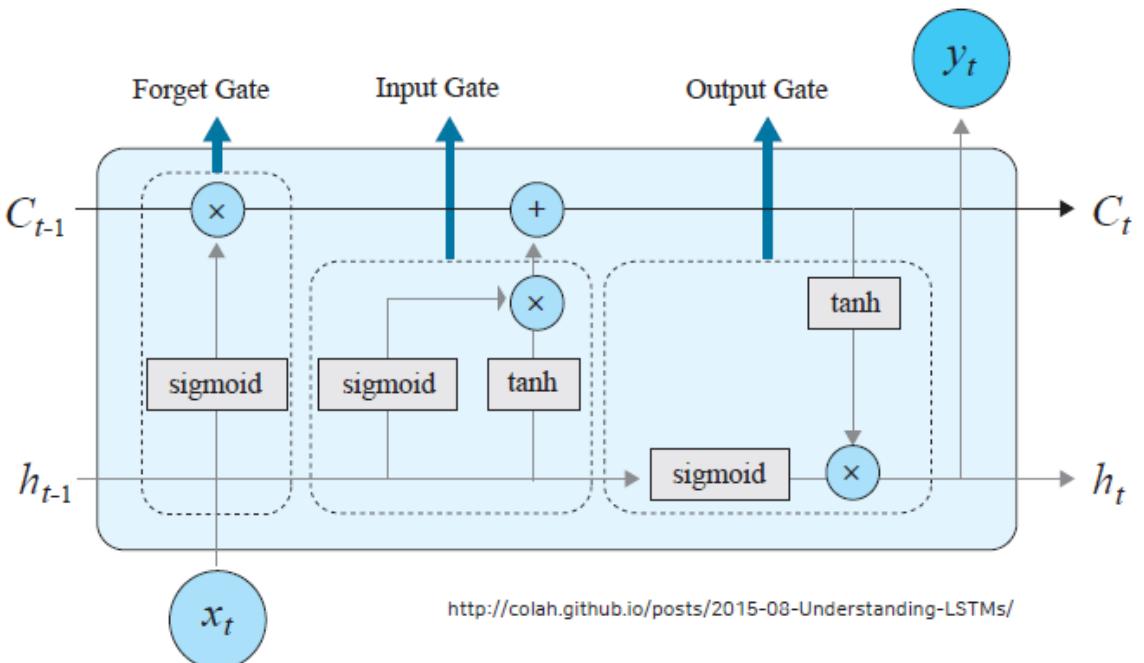
❖ 장단기 기억 네트워크(Long-Short Term Memory Network, LSTM) 이란?

- 1997년 Hochreiter & Schmidhuber 이 제안한 알고리즘
- 순환 신경망에서의 장기 의존성(Long-Term Dependencies) 문제를 해결
- 순차적으로 입력되는 데이터의 시간 흐름이 길더라도 잊어야 할 정보들은 잊고 유지해야 될 정보들은 유지하면서 성능을 최적화
- 순환 신경망에서 존재하지 않던 ct인 셀 스테이트(Cell State)가 추가

장단기 기억 네트워크(LSTM) 구현

❖ 장단기 기억 네트워크구조

- 망각, 입력, 출력의 정도를 조절하는 3개의 게이트(Gate)가 추가
 - ✓ 셀 스테이트 : 각 게이트의 정보들이 다음 단계로 진행될 수 있도록 역할
 - ✓ 망각 게이트 : 셀 스테이트에서 버릴 정보를 정하는 단계
 - 입력값과 이전 은닉층에서 입력된 값과 함께 시그모이드 출력값 생성
 - 시그모이드 출력값이 1인 경우 과거의 값을 그대로 유지하고, 0인 경우에는 완전히 값을 버림
 - ✓ 입력 게이트 : 새로운 정보에 대해 셀 스테이트에 저장할지를 결정하는 단계
 - 시그모이드를 통해 업데이트할 정보 결정
 - tanh 레이어를 통해 셀 스테이트에 더할 새로운 후보 값을 만들고 두 값을 합쳐 새로운 셀 스테이트로 정보를 업데이트
 - ✓ 출력 게이트
 - 어떤 값을 출력할지 시그모이드 레이어를 통해 결정
 - 셀 스테이트를 tanh 레이어를 통한 결괏값을 곱하여 원하는 결괏값만 반영

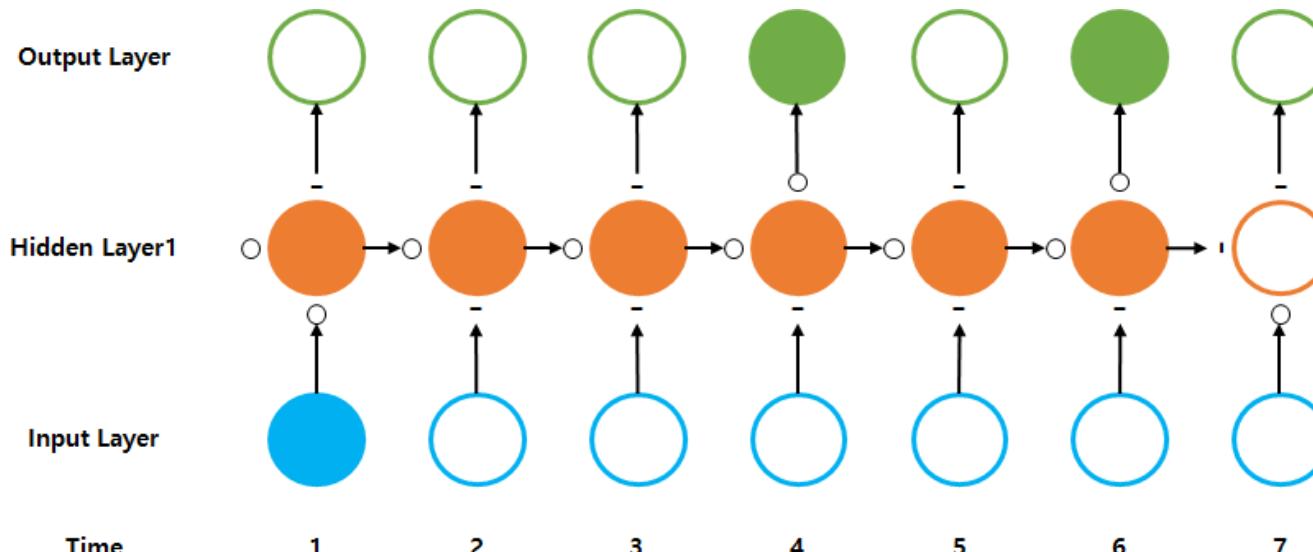


장단기 기억 네트워크(LSTM) 구현

❖ 장단기 기억 네트워크구조

▪ 시간에 따라 각 게이트 동작

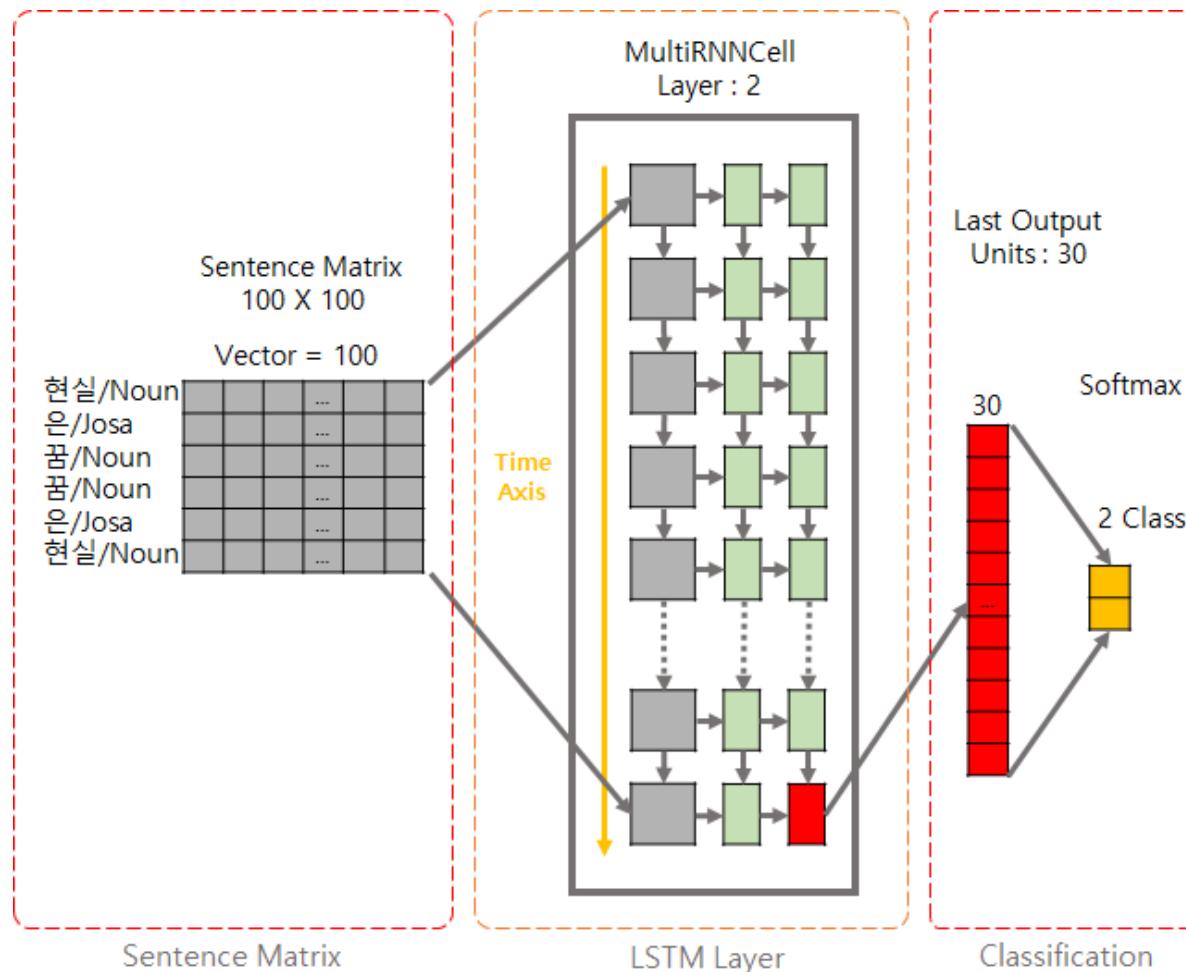
- ✓ 망각, 입력, 출력 게이트를 열고 닫으면서 오랜 시간이 지나더라도 기억을 오랫동안 보존
- ✓ 직선은 닫힌 게이트, 동그라미는 열린 게이트
- ✓ 은닉층의 위, 왼쪽, 아래는 게이트가 출력, 망각, 입력 게이트를 표현
- ✓ 입력층에서는 2~6번째의 시간에서 입력 게이트를 닫음
- ✓ 출력에서는 4, 6번째 시간에서만 출력 게이트를 열어 경사도 사라짐을 방지



<https://skymind.ai/wiki/lstm/>

장단기 기억 네트워크(LSTM) 구현

❖ 장단기 기억 네트워크 구현 전반적 구조



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
```

데이터 수집

❖ Scrapy

- Python으로 작성된 오픈소스 웹 크롤링 프레임워크
- 웹 데이터를 수집하는 것을 목표로 설계 되어 있음
- 웹에 있는 데이터를 검색 및 추출 할 수 있음

```
1 import scrapy
2 import json
3
4 class PostsSpider(scrapy.Spider):
5     name = "posts"
6
7     start_urls = [
8         "https://www.jobstreet.vn/j?sp=search&q=C%C3%B4ng+ngh%E1%BB%87+th%C3%B4ng+tin&l"
9     ]
10
11     def parse(self, response):
12         page_number = 1
13         for post in response.css('a.job-item'):
14             data = {
15                 "url" : post.css(".job-item ::attr(href)").get()
16             }
17             link = "https://www.jobstreet.vn/" + data.get("url")
18             if link is not None:
19                 yield scrapy.Request(url = link, callback = self.parse)
```

Webpage 의 URL

게시판 Page 넘기기

데이터 수집

❖ Pandas

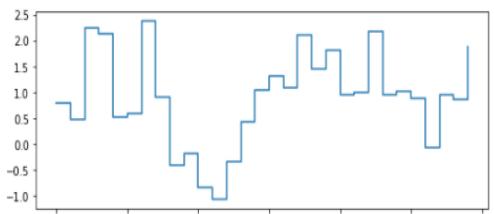
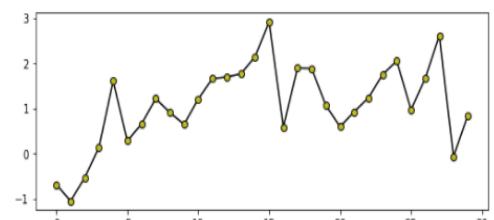
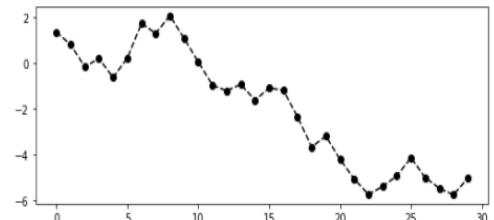
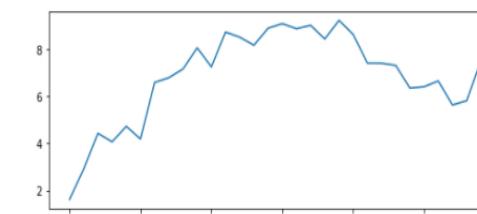
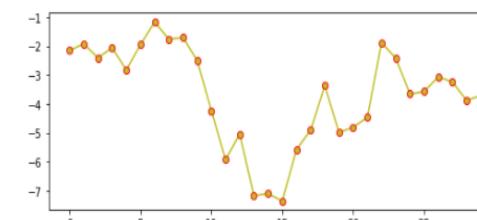
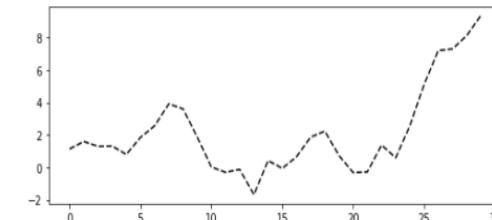
- Pandas는 데이터 처리를 위한 라이브러리 (excel)
- 데이터 분석을 위한 필수 패키지 삼대장 중 1 (Pandas, Numpy, Matplotlib)

```
In [3]: import pandas as pd
# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]
       }
data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
# in the Jupyter notebook
```

```
In [4]: data_pandas
```

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

```
[<matplotlib.lines.Line2D at 0x11cba44a8>]
```

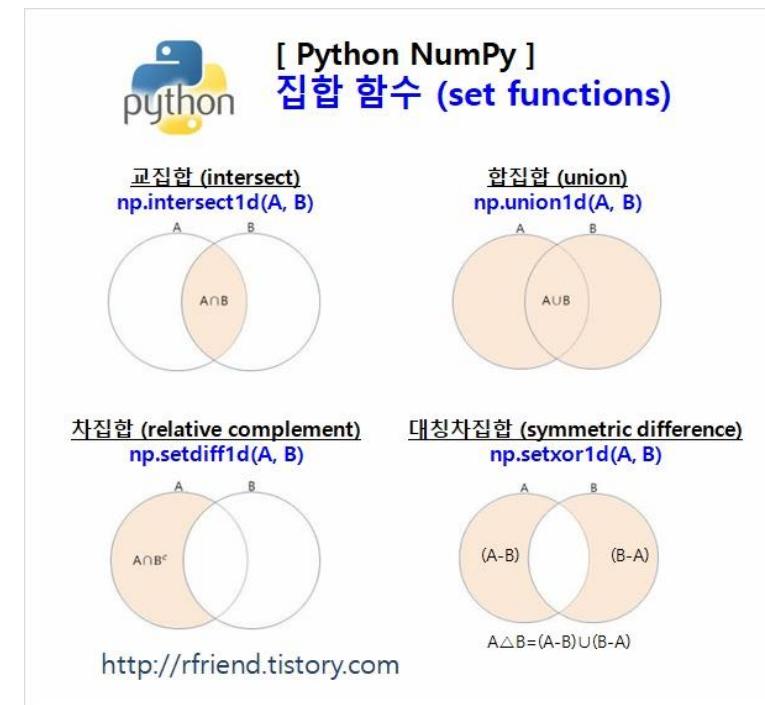
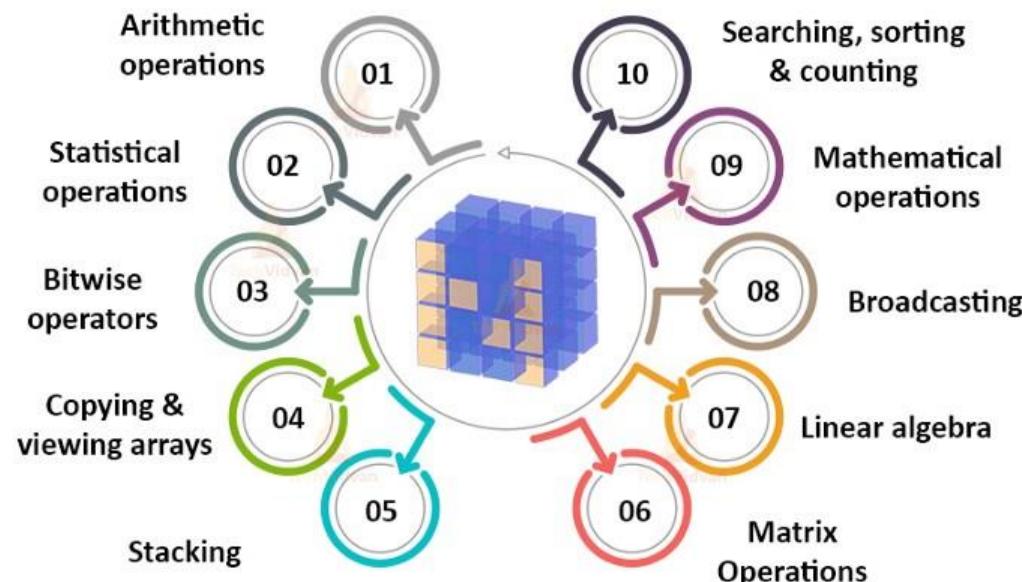


연산 및 패턴인식

❖ NumPy

- 수학적인 계산과 연산, 특히 행렬이 필요
- 선형 대수, 난수, 행렬 등 n차원 배열과 복잡한 수학연산을 처리하는 기능
- 음파, 이진 함수 등 과학적 계산까지 가능

Uses of NumPy



연산 및 패턴인식

❖ SciPy

- 행렬에 기반한 수학, 과학 알고리즘을 지원
- 지수 및 삼각함수, 이중 적분, 보간, 다변량 보간, 신호처리 (사운드, 이미지)

```
from scipy import signal
x = np.arange(35).reshape(7, 5)
domain = np.identity(3)
print(x, end='nn')
print(signal.order_filter(x, domain, 1))
```

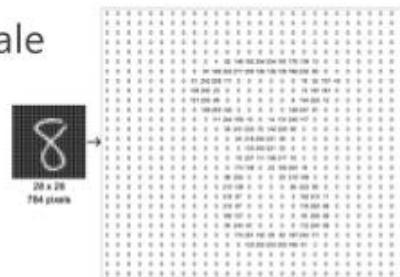
order_filter () : 입력된 신호들을 정렬 하는 기능

사고 및 학습 (문제해결방법)

❖ Tensorflow

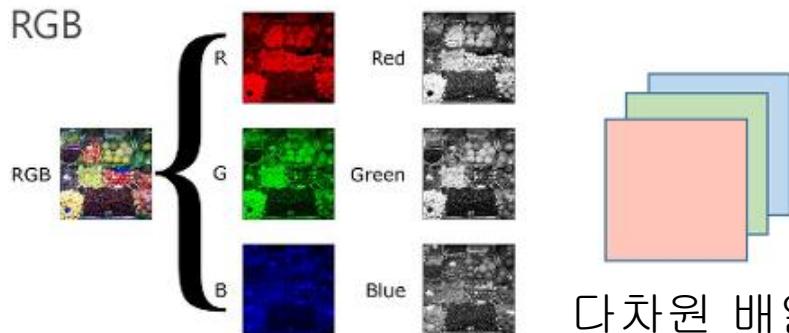
- 구글에서 만든 오픈 AI 프레임워크 (고급 AI 알고리즘을 포함)
- Tensor는 2차원 형태의 배열을 높은 차원으로 확장한 다차원 배열
- Flow는 흐름 (데이터의 흐름 그래프)

Grayscale



2차원 배열

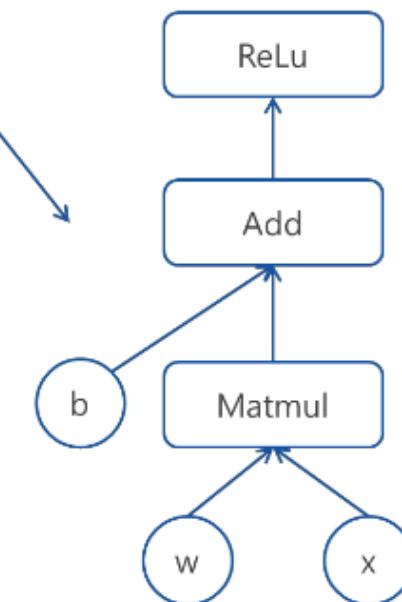
RGB



다차원 배열

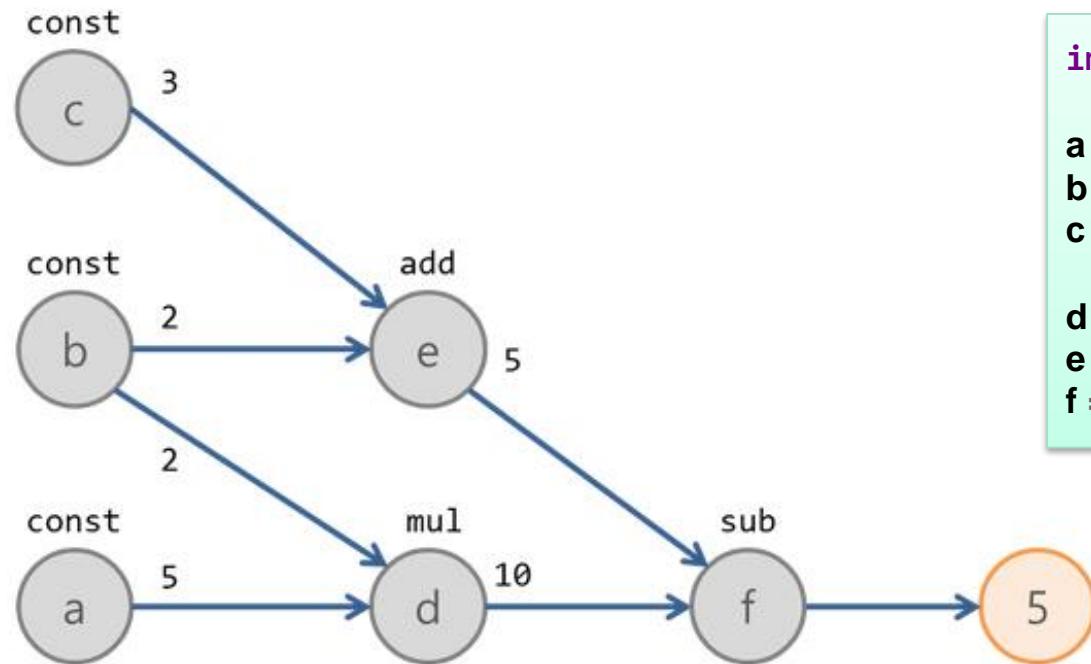
TensorFlow

3.2	-1.4	5.1	...
-1.0	-2	2.4	...
...
...



사고 및 학습 (문제해결방법)

❖ Tensorflow



```
import tensorflow as tf
```

```
a = tf.constant(5)  
b = tf.constant(2)  
c = tf.constant(3)
```

```
d = tf.multiply(a, b) # a * b  
e = tf.add(c, b) # c + b  
f = tf.subtract(d, e) # d - e
```

사고 및 학습 (문제해결방법)

❖ PyTorch

- Tensorflow와 같이 딥러닝 프레임워크 중 하나
- Facebook이 구글의 Tensorflow를 맞서기 위해 개발
- 자연어 처리에 대한 강점을 가지고 있음

PyTorch vs. Tensorflow

- 1.설치가 간편하다.
- 2.이해와 디버깅이 쉬운 직관적이고 간결한 코드로 구성되었다.
- 3.Define by Run 방식을 기반으로 한 실시간 결과값을 시각화 한다.
- 4.파이썬 라이브러리(Numpy, Scipy, Cython)와 높은 호환성을 가진다.

구분	Tensorflow	PyTorch
패러다임	Define and Run	Define by Run
그래프 형태	Static graph(정적)	Dynamic graph(동적)
현재 사용자	많음	적음
자체 운영 포럼	없음	있음
한국 사용자 모임	Tensorflow Korea(TF-KR)	Pytorch Korea(Pytorch-KR)

사고 및 학습 (문제해결방법)

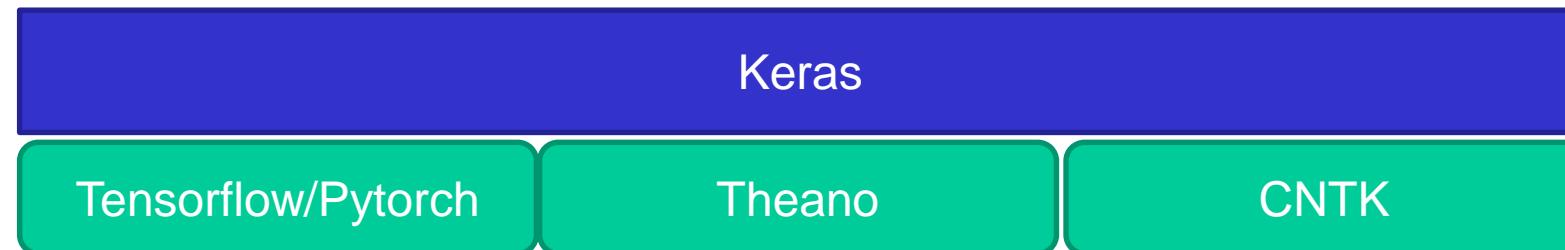
❖ Theano

- 최초의 딥러닝 라이브러리
- 다차원 배열과 관계가 있는 수학적 표현을 정의, 최적화, 평가 가능
- Tensorflow 와 같이 저수준 라이브러리로 딥러닝 모델을 직접 만들 수 있음
- 캐나다 몬트리올 대학의 머신러닝 연구그룹이 개발

사고 및 학습 (문제해결방법)

❖ Keras (일종의 인터페이스)

- Tensorflow와 동일하게 딥러닝 프레임워크 중 하나
- Tensorflow를 보다 쉽게 사용할 수 있도록 상위 레벨의 인터페이스를 제공
- 사용자는 Tensorflow, Theano, CNTK(MS) 등 딥러닝 전용 엔진의 내부구조를 전혀 알 필요가 없음
- 딥러닝 비전문가들이 주로 사용, 다층퍼셉트론 모델, 신경망 모델, 순환 신경망 모델 등 다양한 구성이 가능
- (Tensorflow 2.0에 내포)



AI framework 오픈소스 라이브러리 인기 순위

딥러닝 오픈소스 라이브러리 인기 순위

순위	딥러닝 라이브러리	점수
1	텐서플로(tensorflow)	172
2	카페(caffe)	89
3	케라스(keras)	69
4	MX넷(mxnet)	53
5	테아노(Theano)	38
6	딥러닝4j(Deeplearning4j)	29
7	CNTK	27
8	토치7(torch7)	17
9	패들(paddle)	14
10	체이너(chainer)	13

출처:깃허브

Pytorch로 컬러 이미지 구분

❖ CIFAR-10 데이터

- **Canadian Institute For Advanced Research**
- 32x32픽셀의 60000개 컬러이미지가 포함되어있으며, 각 이미지는 10개의 클래스로 라벨링
- train데이터 50,000개와 test 데이터 10,000개로 구성

```
# 파일 패키지 수입
import numpy as np
import matplotlib.pyplot as plt
from time import time
import torch
from torch import nn

import torchvision
import torchvision.transforms as transforms
from torchvision.datasets import CIFAR10
from torch.utils.data import DataLoader
from torchsummary import summary
```

비행기



자동차



새



고양이



사슴



개



개구리



말

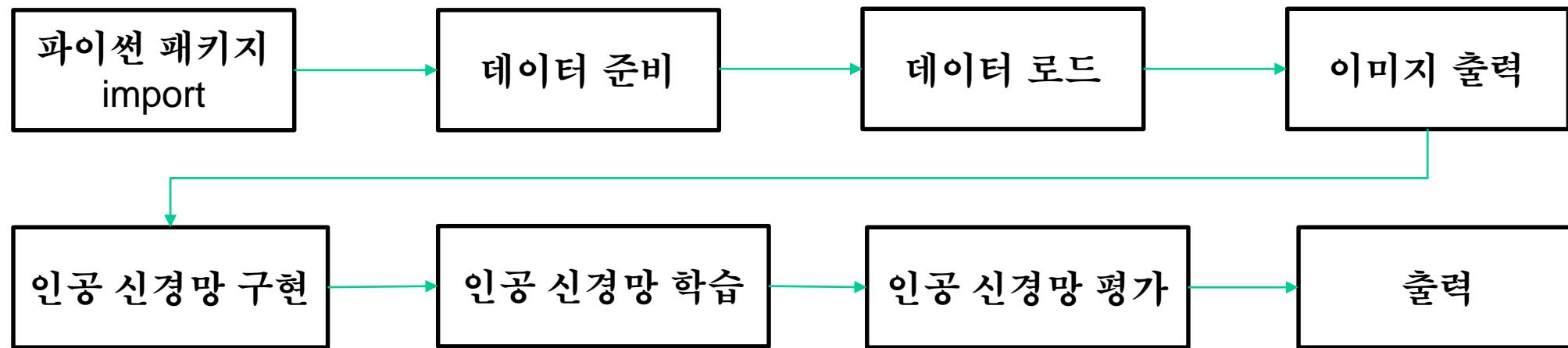


배



트럭

Pytorch로 컬러 이미지 구분



Pytorch로 컬러 이미지 구분

```
# 파이토치 CNN을 Sequential 모델로 구현
model = nn.Sequential()

# 첫번째 합성곱 블럭
nn.Conv2d(in_channels=3,
          out_channels=6,
          kernel_size=4),
nn.ReLU(),
nn.MaxPool2d(kernel_size=2,
             stride=2),

# 두번째 합성곱 블럭
nn.Conv2d(in_channels=6,
          out_channels=16,
          kernel_size=4),
nn.ReLU(),

# DNN 블럭
nn.Flatten(),
nn.Linear(in_features=16 * 11 * 11,
          out_features=120),
nn.Linear(120, 84),
nn.Linear(84, 10),
nn.Softmax(dim=1))

)
```

```
# CNN 학습
begin = time()
print('\nCNN 학습 시작')

for epoch in range(MY_EPOCH):
    batch = 0
    for data in train_loader:
        inputs, labels = data
        outputs = model(inputs)

        # 손실값 계산
        loss = criterion(outputs, labels)
        print(' 배치:', batch,
              ' 손실: {:.3f}'.format(loss.item()))

        # 역전파 알고리즘으로 가중치 보정
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        batch += 1

    # 손실값 출력
    print('에포크: {},'.format(epoch),
          '손실: {:.3f}'.format(loss.item()))
```

Pytorch로 컬러 이미지 구분

```
# 이미지 라벨  
classes = ['airplane', 'automobile', 'bird', 'cat', 'deer',  
          'dog', 'frog', 'horse', 'ship', 'truck']
```

```
# 혼동 행렬 초기화  
correct = 0  
confusion = np.zeros([10, 10], int)
```

```
# 평가용 데이터로 CNN 평가  
with torch.no_grad():  
    for data in test_loader:  
        images, labels = data  
        outputs = model(images)  
  
        # 1차원 축으로 최대치 찾아 내기  
        _, pred = torch.max(outputs, dim=1)
```

```
# 혼동행렬 업데이트  
for i, truth in enumerate(labels):  
    if (truth.item() == pred[i]):  
        correct += 1  
    confusion[pred[i].item(), truth.item()] += 1
```

```
# 최종 정확도 출력  
print("\n최종 정확도: {:.2f}%"  
     .format(correct / 10000 * 100))
```

```
# 카테고리별 정확도 출력  
print('카테고리별 Precision:')  
for i, row in enumerate(confusion):  
    print('{0:10s} : {1:.1f}%'  
          .format(classes[i], row[i]/np.sum(row)*100))
```

```
# 혼동 행렬 출력  
print("\n혼동 행렬:")  
print(confusion)
```

```
# 평가용 마지막 batch (16개 데이터) 결과  
print("\n마지막 batch 데이터 모양:", data[0].shape)  
print('예상:', pred)  
print('정답:', labels)
```

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])
```

데이터 전처리

❖ 데이터 전처리(Data Preprocessing)

■ 데이터 전처리

대부분의 데이터 분석 업무는 반드시 데이터 전처리 과정을 거쳐야 합니다. 데이터의 품질과 데이터에 담긴 정보량에 따라 머신러닝에서는 학습 모델의 정확도에 영향을 주게 됩니다. 실제 데이터는 학습 데이터로 사용할 수 없을 정도로 지저분한 상태인 데이터가 많습니다. 이러한 데이터를 분석이 가능한 상태로 만들기 위해 데이터 전처리 과정을 통하여 데이터를 다시 가공합니다.

- 데이터 전처리 과정
 - 1. 결측값(Missing Value)을 제거하거나 다른 값으로 대치
 - 2. 범주형 자료는 정수형 데이터로 매팅
 - 3. 변수 표준화 : 측정 단위에 따라 머신러닝 알고리즘이 왜곡되는 현상을 방지하기 위하여 변수를 정규화, 표준화 척도를 조정

데이터 전처리

❖ 학습 데이터 준비 : 데이터 전처리 (정규화)

■ Feature Scaling

데이터의 값이 너무 크거나 작아서 변수의 영향이 제대로 반영 안 될 수 있기 때문에 변수를 같은 크기의 척도로 조정하여 학습에 동일한 영향을 주도록 합니다.

1. Min, Max Normalization

- 데이터를 일반적으로 0~1 사이의 값으로 변환
- 식 : $(X - X\text{의 최솟값}) / (X\text{의 최댓값} - X\text{의 최솟값})$
- 데이터의 최솟값, 최댓값을 알 경우 사용

2. Standardization

- 기존 변수에 범위를 정규 분포로 변환
- 식 : $(X - X\text{의 평균값}) / (X\text{의 표준편차})$
- 데이터의 최솟값, 최댓값을 모를 경우 사용

과적합, 과소적합

❖ 데이터 섞기

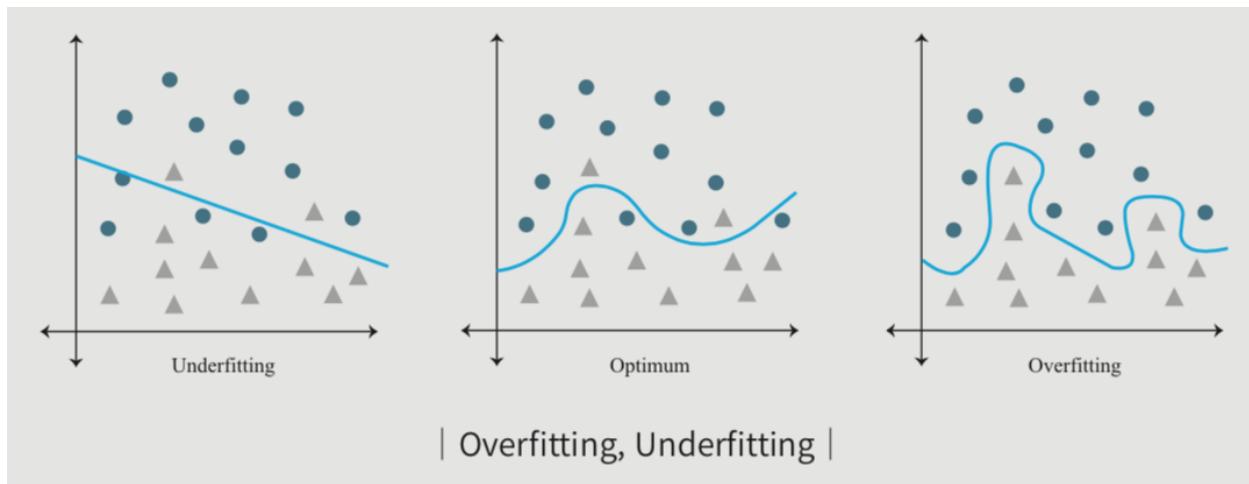
- 학습 된 모델이 훈련데이터에 Overfitting 되는 것을 방지

- ✓ Overfitting(과적합)

- : 한정된 학습 데이터를 과하게 잘 학습하면 특화된 모델이 생성되지만 테스트 데이터나 다른 데이터가 입력 되면 오차가 증가 하는 현상

- ✓ Underfitting(과소적합)

- : 학습데이터에 대해서 충분히 학습하지 못하여 테스트 데이터뿐만 아니라 학습 데이터에서도 성능이 낮게 나타나는 현상



▪ Overfitting 해결법

다음과 세가지 방법을 이용하여 Overfitting 문제를 해결할 수 있습니다.

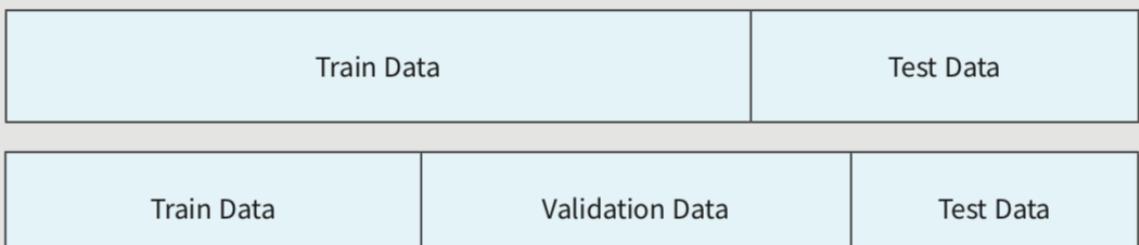
1. 학습 데이터의 수를 늘려 학습을 진행
2. 학습 데이터의 중복된 feature를 제거
3. Regularization(정규화)

학습데이터 나누기

❖ 학습 데이터 나누기

■ 학습 데이터 나누기

학습 데이터는 훈련 데이터(Train Data), 검증 데이터(Validation Data), 테스트 데이터(Test Data) 3가지 혹은 2가지로 나눠서 사용합니다. 훈련 데이터는 우리가 생성한 학습 모델의 W , b 의 변수를 최적화하여 모델을 학습할 때 사용하며 검증 데이터는 생성한 학습 모델의 성능을 평가하기 위해서 사용합니다. 테스트 데이터는 완성한 모델에 입력하여 결과를 확인하기 위하여 사용됩니다.



| 학습 데이터 나누기 |

학습 데이터가 많을 경우 위 그림처럼 2가지 혹은 3가지로 나눠서 학습 모델을 평가합니다. 그러나 학습 데이터가 적을 경우 검증 데이터를 생략하고 모델을 생성, 평가하기도 합니다.

■ 검증 데이터와 테스트 데이터 차이점

검증 데이터와 테스트 데이터는 모델의 성능을 평가하는데 사용되는 데이터로 비슷한 역할을 하는데 나눠서 사용하는지에 대한 의문이 들게 됩니다. 두 데이터가 사용되는 방법 차이는 다음과 같습니다.

1. 검증 데이터는 학습(Optimization)에 사용되지만 테스트 데이터는 학습에 쓰이지 않음
2. 검증 데이터는 다양한 학습 모델 중 성능이 좋은 모델 하나를 결정하는 데 사용
3. 테스트 데이터는 최종적으로 선택된 모델의 성능을 평가하기 위하여 사용

즉 W , b 의 최적값을 찾는 학습 단계에 검증 데이터를 사용하고 학습이 모두 끝난 후 최종적으로 만들어진 모델에 테스트 데이터를 사용하여 모델의 성능을 평가합니다.

Softmax Function

- ❖ Softmax Function 수식 정리

(https://en.wikipedia.org/wiki/Multinomial_logistic_regression)

The resulting equations for the probabilities are

- ❖ 전체값을 합치면 1이 된다.

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot \mathbf{X}_i}}{\sum_{k=1}^K e^{\beta_k \cdot \mathbf{X}_i}}$$

$$\Pr(Y_i = 2) = \frac{e^{\beta_2 \cdot \mathbf{X}_i}}{\sum_{k=1}^K e^{\beta_k \cdot \mathbf{X}_i}}$$

.....

$$\Pr(Y_i = K) = \frac{e^{\beta_K \cdot \mathbf{X}_i}}{\sum_{k=1}^K e^{\beta_k \cdot \mathbf{X}_i}}$$

Or generally:

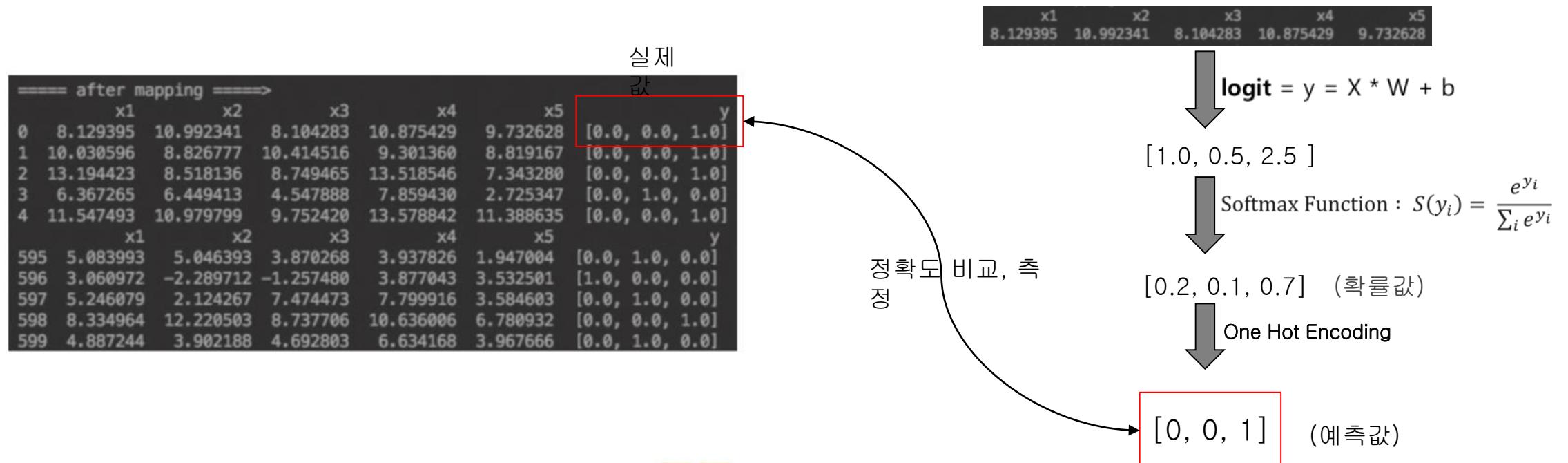
$$\Pr(Y_i = c) = \frac{e^{\beta_c \cdot \mathbf{X}_i}}{\sum_{k=1}^K e^{\beta_k \cdot \mathbf{X}_i}}$$

The following function:

$$\text{softmax}(k, x_1, \dots, x_n) = \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}}$$

One Hot Encoding

- ❖ One Hot Encoding
 - 가장 큰 확률을 1로 표현하고 나머지는 0으로 표현
 - 예를 들어 [0.2, 0.1, 0.7] 의 값을 가진 결과를 One Hot Encoding을 적용하면 [0, 0 ,1] 이 됨
- ❖ Softmax Function을 통하여 나온 확률 예측값을 One Hot Encoding 적용하여 값을 변경하고 학습데이터의 실제값(종속변수)과 비교하여 예측 정확도를 측정



배치트레이닝

❖ 배치 트레이닝

- 학습데이터를 일정 사이즈로 나누고 미니 배치(mini batch)를 통해 Gradient descent를 수행하는 방법
- 미니 배치를 이용하여 기울기를 계산하여 W , b 를 업데이트 함
- 전체데이터를 이용하여 W , b 의 값을 한번에 업데이트 하는 방법보다 빠르게 W , b 를 업데이트 할 수 있음
- 배치 사이즈는 메모리가 허용하는 범위 내에서 최대한 크게 잡는것이 좋음
- epoch은 전체 데이터 학습 하는 횟수

❖ 학습 조건

- 훈련 데이터수 : 55,000개
- 최적화 함수 : Adams Optimizer
- 학습률 : 0.001
- totalEpochs : 20회
- totalBatch : 275회
- batch_size : 200개 데이터

배치트레이닝(예제)

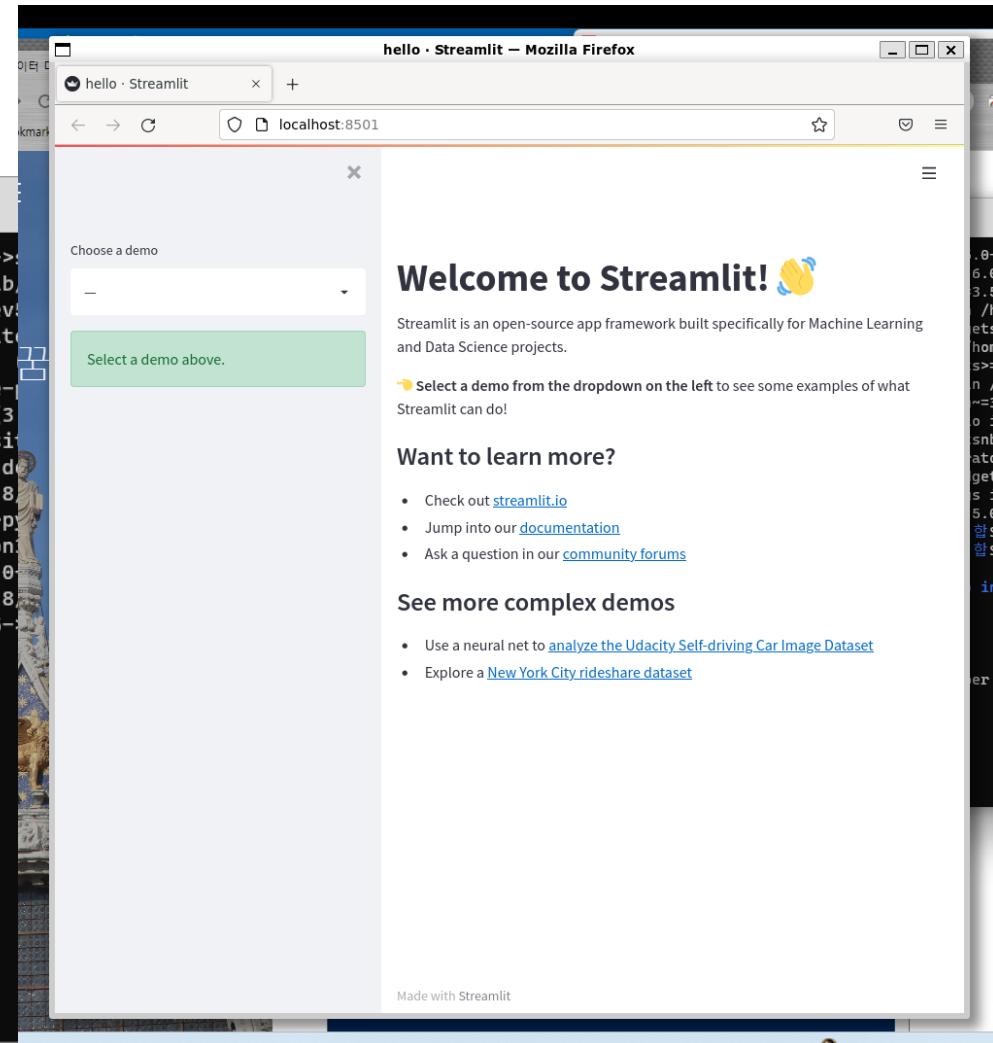
```
9 #####  
10 # [환경설정]  
11 #####  
12 # 학습률  
13 learningRate = 0.001  
14  
15 # 총 학습 횟수  
16 totalEpochs = 20  
17 # 학습데이터를 나누기 위한 값  
18 # 학습데이터 총수 / batch_size = 한번의 epoch 쓰이는 데이터 수  
19 batch_size = 200  
20  
21 # W, b 변수 생성 타입 (1 : random_normal, 2: truncated_normal, 3: random_uniform)  
22 randomVariableType = 1  
23  
24 # input Layer 크기  
25 # 입력 데이터 크기 784 (손글씨 이미지는 28 * 28 픽셀로 총 784개)  
26 inputDataSize = 28 * 28 # 입력 데이터 고정값(수정불가)  
27  
28 # hidden Layer 크기  
29 hiddenLayer1Size = 1024  
30 hiddenLayer2Size = 512  
31 hiddenLayer3Size = 256  
32  
33 # output Layer 크기  
34 # 출력값 크기 (Output Layer에서 출력되 데이터(0~9까지 숫자)  
35 outputLayerSize = 128  
36 outputDataSize = 10 # 출력값 크기 고정(수정불가)  
37  
203 print("-----")  
204 print("Train(Optimization) Start ")  
205 for epoch in range(totalEpochs):  
206     average_costFunction = 0  
207     # 전체 batch 사이즈 구하기 (55000 / 200 = 275)  
208     totalBatch = int(mnist.train.num_examples / batch_size)  
209  
210     for step in range(totalBatch):  
211         batchX, batchY = mnist.train.next_batch(batch_size)  
212         cost_val, acc_val, _ = sess.run([costFunction, accuracy, train],  
213                                         feed_dict={X: batchX, Y: batchY})  
214         train_accuracy.append(acc_val)  
215         average_costFunction = cost_val / totalBatch  
216  
217     print("epoch : {}, cost = {}".format(epoch, average_costFunction))  
218  
219     # 정확도 결과 확인 그래프  
220     plt.plot(range(len(train_accuracy)),  
221                 train_accuracy,  
222                  linewidth=2,  
223                  label='Training')  
224     plt.legend()  
225     plt.title("Accuracy Result")  
226     plt.show()  
227  
228 print("Train Finished")
```

STREAMLIT

Streamlit

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

```
oykwon@Ohyoung-K506:/mr ~ + ~  
->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in /home/oykwon/anaconda3/lib/  
erty->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
Requirement already satisfied: testpath in /home/oykwon/anaconda3/lib/python3.8/site-  
=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit  
Requirement already satisfied: bleach in /home/oykwon/anaconda3/lib/python3.8/site-  
.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->streamlit (3  
Requirement already satisfied: pyparser in /home/oykwon/anaconda3/lib/python3.8/site-  
-cffi->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
Requirement already satisfied: nest-asyncio in /home/oykwon/anaconda3/lib/python3.8/site-  
=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
Requirement already satisfied: async-generator in /home/oykwon/anaconda3/lib/python3.8/site-  
0,>=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
Requirement already satisfied: webencodings in /home/oykwon/anaconda3/lib/python3.8/site-  
t->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->pydeck>=0.1.dev5->  
(base) oykwon@Ohyoung-K506:/mnt/c/Users/용한$ pip install streamlit^C  
(base) oykwon@Ohyoung-K506:/mnt/c/Users/용한$ 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

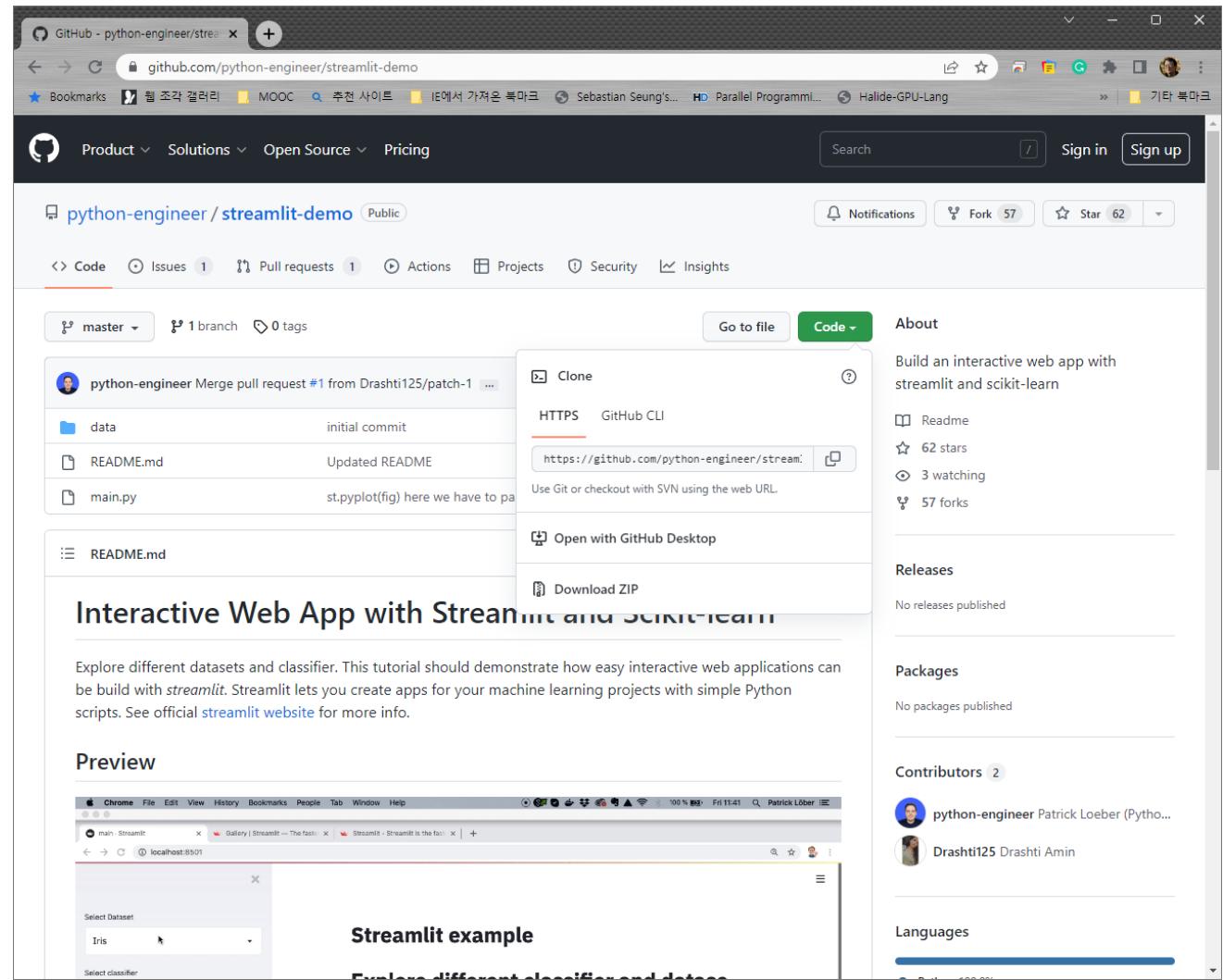
You need these dependencies:

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

Usage

Run

```
streamlit run main.py
```

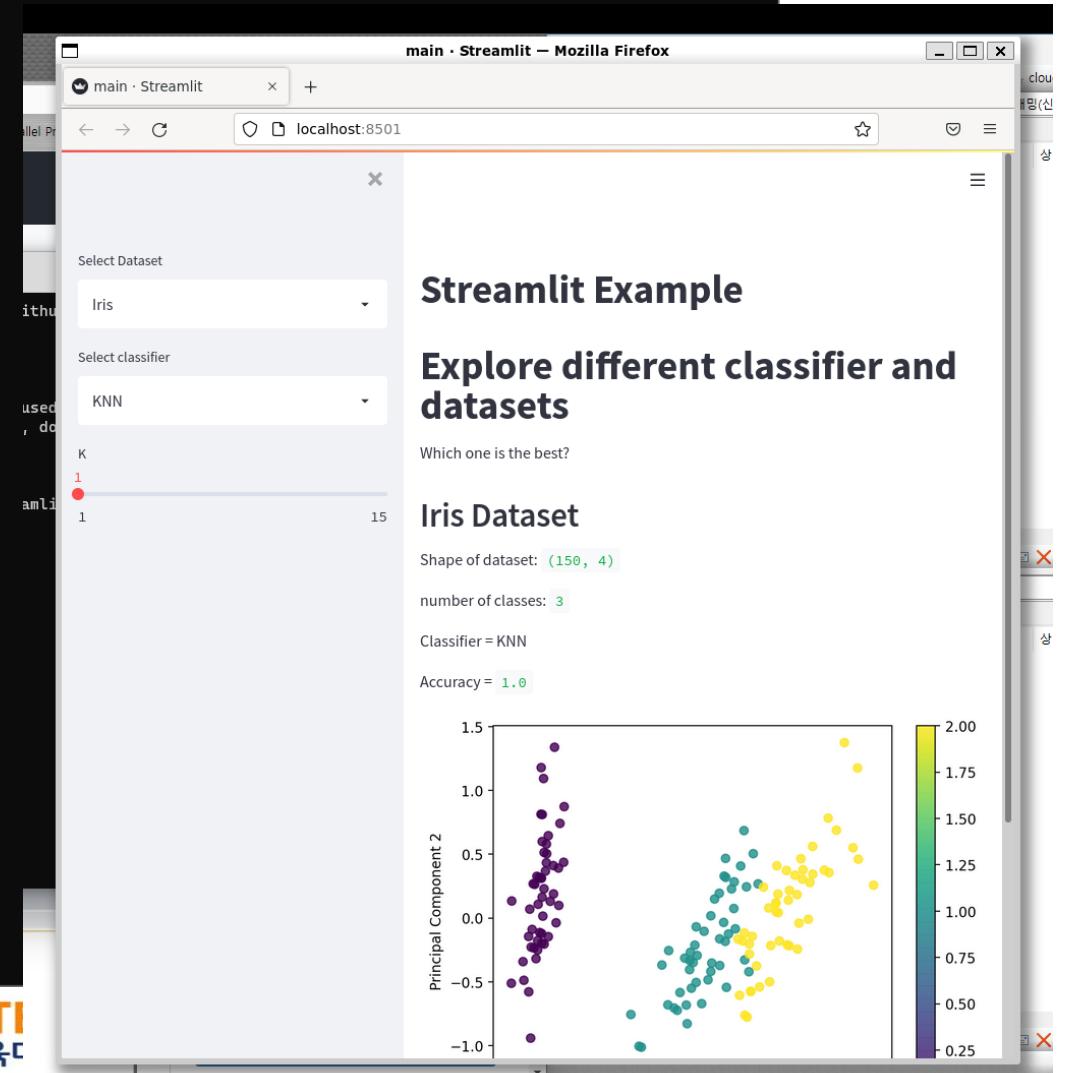


Streamlit Demo

```
oykwon@Ohyoung-K506:~/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 mast x +

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

Bookmarks 웹 조각 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's...

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 )
```

streamlit-demo/main.py at mast 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

```
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
64 return params
```

```
streamlit-demo/main.py at mast + https://github.com/python-engineer/streamlit-demo/blob/master/main.py
62     params['n_estimators'] = n_estimators
63
64
65     return params
66
67 def get_classifier(clf_name, params):
68     clf = None
69
70     if clf_name == 'SVM':
71         clf = SVC(C=params['C'])
72     elif clf_name == 'KNN':
73         clf = KNeighborsClassifier(n_neighbors=params['K'])
74     else:
75         clf = RandomForestClassifier(n_estimators=params['n_estimators'],
76                                     max_depth=params['max_depth'], random_state=1234)
77
78     return clf
79
80
81 #### CLASSIFICATION ####
82
83 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234)
84
85 clf.fit(X_train, y_train)
86 y_pred = clf.predict(X_test)
87
88 acc = accuracy_score(y_test, y_pred)
89
90 st.write(f'Classifier = {classifier_name}')
91 st.write(f'Accuracy = ', acc)
92
93 #### PLOT DATASET ####
94 # Project the data onto the 2 primary principal components
95 pca = PCA(2)
96 X_projected = pca.fit_transform(X)
97
98 x1 = X_projected[:, 0]
99 x2 = X_projected[:, 1]
100
101 fig = plt.figure()
102 plt.scatter(x1, x2,
103             c=y, alpha=0.8,
104             cmap='viridis')
105 plt.xlabel('Principal Component 1')
106 plt.ylabel('Principal Component 2')
107 plt.colorbar()
108
109 st.pyplot(fig)
```

Streamlit 참고사이트

- ❖ 유튜브 Python Engineer
 - streamlit 검색 상위 4개

Google Calendar - 2022년 10월 x Python Engineer - YouTube x +

youtube.com/c/PythonEngineer/search?query=streamlit

Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's... HD Parallel Programming Halide-GPU-Lang 기타 북마크

YouTube KR 검색

python-engineer.com Python and Machine Learning Tutorials

Python Engineer 구독자 19.1만명 구독

구독

Machine Learning Web App Streamlit 38:45

Build A Beautiful Machine Learning Web App With Streamlit And Scikit-learn | Python Tutorial Python Engineer 조회수 11만회 · 2년 전 Get my Free NumPy Handbook: <https://www.python-engineer.com/numpybook> In this tutorial we build an interactive machine learning app with Streamlit and Scikit-learn to explore different datasets...

Build A PyTorch Style Transfer Web App With Streamlit Python Engineer 조회수 7.4천회 · 2년 전 In this tutorial we build an interactive deep learning app with Streamlit and PyTorch to apply style transfer. This tutorial should demonstrate how easy interactive web applications can be...

MACHINE LEARNING WEB APP FROM SCRATCH 53:20

Build A Machine Learning Web App From Scratch Python Engineer 조회수 8.8만회 · 1년 전 Build a Machine Learning web application from scratch in Python with Streamlit. We use real world data to build a machine learning model. In the first part you learn how we analyze the data...

STOCK PREDICTION WEB APP

Build A Stock Prediction Web App In Python Python Engineer 조회수 9.1만회 · 1년 전 In this tutorial we build a stock prediction web app in Python using streamlit, yahoo finance, and Facebook...

Python Streamlit 사용법 - 프로토타입 만들기

07 Feb 2021 in MLOps on Basic

- Python Streamlit에 대한 글입니다
 - python streamlit tutorial, python streamlit dashboard, python streamlit install, python streamlit vs dash, python dashboard, python streamlit example
- 목차
 - 회사 또는 프로젝트하며 생길 수 있는 일
 - 대표적인 프로토타이핑 도구
 - Streamlit 소개
 - Streamlit 맛보기
 - Streamlit API 소개
 - Streamlit Component
 - Streamlit 배포하기

회사 또는 프로젝트하며 생길 수 있는 일

- 1) 머신러닝 모델 학습
- 2) 머신러닝 모델 배포
- 3) 머신러닝 모델이 잘 동작하는지 Jupyter Notebook 등에서 확인
- 4) 모델을 만든 당사자는 노트북 파일을 기반으로 여러 동작을 할 수 있음
 - 그러나 자신만 디버깅하는게 아닌 다른 분들에게 공유하고 싶은 경우엔 이슈가 존재

Python Streamlit 사용법 - 프로토타입 만들기

Backend.AI 0:0 - "main1" /bin/bash -c

zszsa.github.io/mlops/2021/02/07/python-streamlit-dashboard/ Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's... 기타 북마크

어쩐지 오늘은

Machine Learning Engineer

Diary

Data

MLOps

Google Cloud Platform

Development

ETC

About

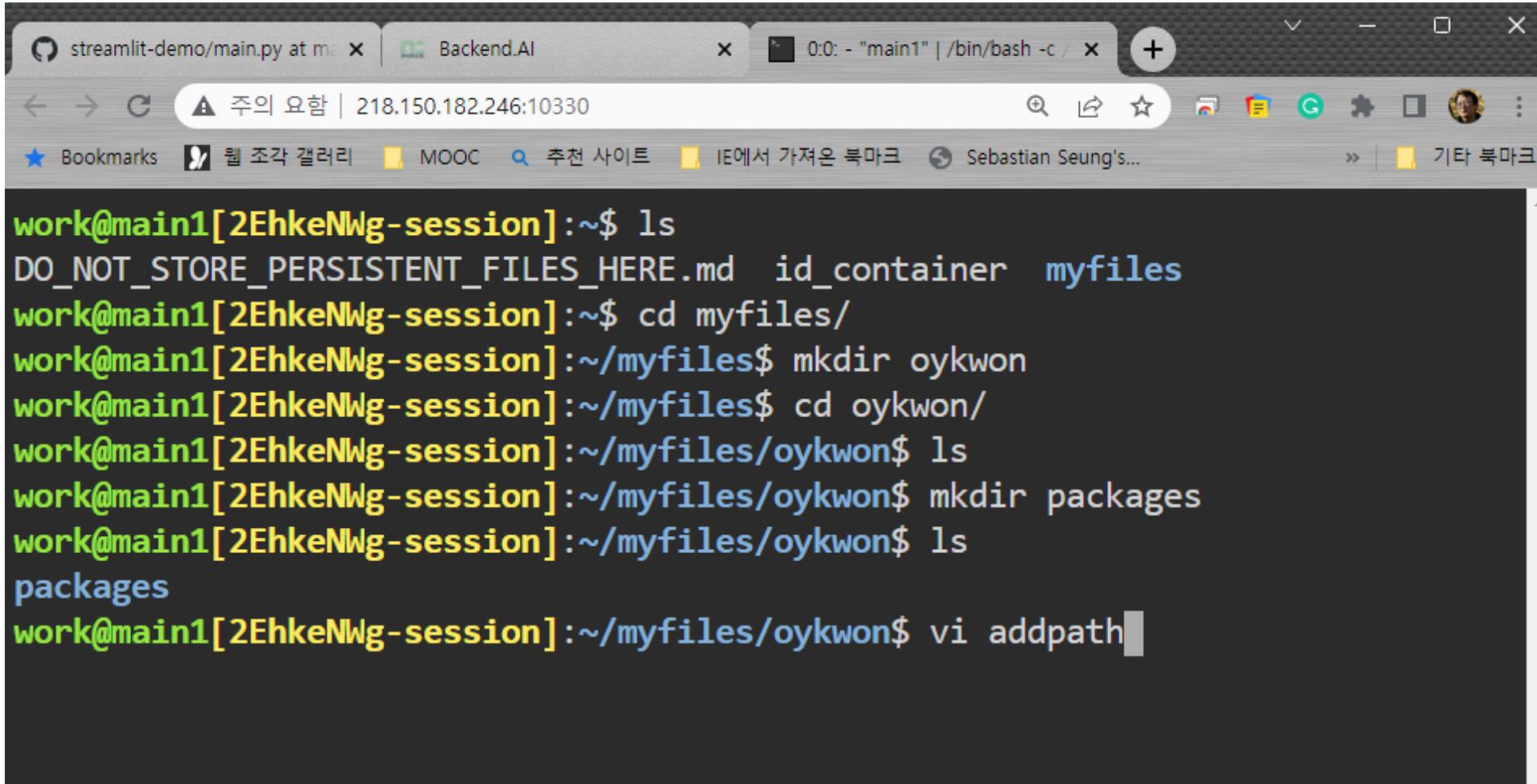
Search

메모가 습관인 데이터쟁이입니다

f YouTube Instagram GitHub

교내 클라우드에서 Streamlit 활용

- ❖ 반드시 자신의 파일폴더를 마운트 (eg. myfiles)



The screenshot shows a terminal window with a dark background and light-colored text. The terminal title is "streamlit-demo/main.py at main · Backend.AI". The URL in the address bar is "주의 요함 | 218.150.182.246:10330". The terminal content is a series of bash commands:

```
work@main1[2EhkeNwg-session]:~$ ls
DO_NOT_STORE_PERSISTENT_FILES_HERE.md  id_container  myfiles
work@main1[2EhkeNwg-session]:~$ cd myfiles/
work@main1[2EhkeNwg-session]:~/myfiles$ mkdir oykwon
work@main1[2EhkeNwg-session]:~/myfiles$ cd oykwon/
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ mkdir packages
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls
packages
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ vi addpath
```

The screenshot shows a terminal window with the following details:

- Terminal title: streamlit-demo/main.py at master
- Current directory: Backend.AI
- Process ID: 0:0 - "main1" | /bin/bash -c

The terminal content is as follows:

```
1 #!/bin/bash
2
3 export PYTHONPATH="/home/work/myfiles/oykwon/packages:$PYTHONPATH"
4 export PATH="/home/work/myfiles/oykwon/packages/bin:$PATH"
~
```

The bottom status bar indicates:

- addpath [+]
- 4,1
- :wq
- [0] 0:*
- "main1" 04:52 12

```
streamlit-demo/main.py at master · Backend.AI · GitHub
Backend.AI
0:0 - "main1" | /bin/bash -c
▲ 주의 요함 | 218.150.182.246:10330
Bookmarks 웹 조각 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's...
» 기타 북마크

work@main1[2EhkeNwg-session]:~$ ls
DO_NOT_STORE_PERSISTENT_FILES_HERE.md  id_container  myfiles
work@main1[2EhkeNwg-session]:~$ cd myfiles/
work@main1[2EhkeNwg-session]:~/myfiles$ mkdir oykwon
work@main1[2EhkeNwg-session]:~/myfiles$ cd oykwon/
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ mkdir packages
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls
packages
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ vi addpath
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls
addpath  packages
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ source addpath
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ env
```

```
streamlit-demo/main.py at m x Backend.AI x 0:0 - "main1" | /bin/bash -c x +  
 주의 요함 | 218.150.182.246:10330  
 Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's...  
  
TORCHVISION_VERSION=0.8.2  
NPROC=2  
TERM=screen  
SHELL=/bin/bash  
MPLBACKEND=Svg  
OPENCV_VERSION=4.5.1  
CUDA_VERSION=11.1.1  
NUMPY_VERSION=1.19.4  
BACKENDAI_CLUSTER_ROLE=main  
BACKENDAI_CLUSTER_REPLICAS=main:1  
NVIDIA_DRIVER_CAPABILITIES=compute,utility  
PYTHON_VERSION=3.8  
TMUX_PANE=%0  
SHLVL=2  
TENSORBOARDX_VERSION=2.1  
PYTHONPATH=/home/work/myfiles/oykwon/packages:/home/work/.local/lib/python3/site-packages  
NVIDIA_REQUIRE_CUDA=cuda>=11.1 brand=tesla,driver>=418,driver<419 brand=a,driver>=440,driver<441 brand=tesla,driver>=450,driver<451  
CPLUS_INCLUDE_PATH=/usr/include/gdal  
PATH=/home/work/myfiles/oykwon/packages/bin:/usr/local/mpi/bin:/usr/local/nvidia/bin:/usr/local/cuda/bin:/usr/local/sbin:/usr/bin/cmake:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/opt/tensorrt/bin:/usr/src/lightgbm/LightGBM:/usr/local/bin/mecab:~/local/bin  
PS1=\[\033[01;32m\]\u@\main1\[\033[01;33m\][2EhkeNwg-session]\[\033[00m\]\033[01;34m\]\w\[\033[00m\]$  
LD_PRELOAD=/opt/kernel/libbaihook.so  
_CUDA_COMPAT_PATH=/usr/local/cuda/compat  
OMP_NUM_THREADS=2  
_=~/bin/env  
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ [0] 0:*
```

```
streamlit-demo/main.py at m x Backend.AI x 0:0 - "main1" | /bin/bash -c x +  
 주의 요함 | 218.150.182.246:10330  
 Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's... 기타 북마크  
  
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ pip install --target=packages streamlit  
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com  
Collecting streamlit  
  Downloading streamlit-1.13.0-py2.py3-none-any.whl (9.2 MB)  
   |████████| 9.2 MB 6.9 MB/s  
Collecting rich>=10.11.0  
  Downloading rich-12.6.0-py3-none-any.whl (237 kB)  
   |████████| 237 kB 59.6 MB/s  
Collecting validators>=0.2  
  Downloading validators-0.20.0.tar.gz (30 kB)  
Collecting numpy  
  Downloading numpy-1.23.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)  
 )   |████████| 17.1 MB 51.6 MB/s
```

```
Python Streamlit 사용법 - 프트 x Backend.AI x 0:0: - "main1" | /bin/bash -c x + 주의 요함 | 218.150.182.246:10330
Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's...
es-5.10.0 jinja2-3.1.2 jsonschema-4.16.0 numpy-1.23.3 packaging-21.3[40/1.5.0 pillow-9.2.0 pkgutil-resolve-name-1.3.10 protobuf-3.20.3 pyarrow-9 pydeck-0.8.0b3 pygments-2.13.0 pympler-1.0.1 pyparsing-3.0.9 pyrsistent 8.1 python-dateutil-2.8.2 pytz-2022.4 pytz-deprecation-shim-0.1.0.post0.1 ests-2.28.1 rich-12.6.0 semver-2.13.0 six-1.16.0 smmap-5.0.0 streamlit-1.0 toml-0.10.2 toolz-0.12.0 tornado-6.2 typing-extensions-4.4.0 tzdata-20 tzlocal-4.2 urllib3-1.26.12 validators-0.20.0 watchdog-2.1.9 zipp-3.9.0
WARNING: You are using pip version 20.3.2; however, version 22.2.2 is available.
You should consider upgrading via the '/usr/bin/python3 -m pip install --upgrade pip' command.
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ ls packages/
GitPython-3.1.29.dist-info
Jinja2-3.1.2.dist-info
MarkupSafe-2.1.1.dist-info
PIL
Pillow-9.2.0.dist-info
Pillow.libs
o
Pygments-2.13.0.dist-info
Pympler-1.0.1.dist-info
__pycache__
_pyrsistent_version.py
altair
altair-4.2.0.dist-info
attr
attrs
attrs-22.1.0.dist-info
backports
backports.zoneinfo-0.2.1.dist-info
bin
[0] 0:*
```

```
Python Streamlit 사용법 - 프트 x Backend.AI x 0:0: - "main1" | /bin/bash -c x + 주의 요함 | 218.150.182.246:10330
Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's...
기타 북마크
ntimeState.STOPPING -> RuntimeState.STOPPED
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ streamlit --help
/home/work/myfiles/oykwon/packages/pandas/core/computation/expressions.py:21: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.0' currently installed).
from pandas.core.computation.check import NUMEXPR_INSTALLED
Usage: streamlit [OPTIONS] COMMAND [ARGS]...
Try out a demo with:
$ streamlit hello
Or use the line below to run your own script:
$ streamlit run your_script.py
Options:
--log_level [error|warning|info|debug]
--version                                         Show the version and exit.
--help                                            Show this message and exit.
Commands:
activate   Activate Streamlit by entering your email.
cache      Manage the Streamlit cache.
config     Manage Streamlit's config settings.
docs       Show help in browser.
hello      Runs the Hello World script.
help       Print this help message.
run        Run a Python script, piping stderr to Streamlit.
version    Print Streamlit's version number.
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ [0] 0:*
[0] 0:*
[0] 0:*
```

```
Python Streamlit 사용법 - 프로젝트 x Backend.AI x 0:0: - "main1" | /bin/bash -c /op x +  
Bookmarks 웹 조작 갤러리 MOOC 추천 사이트 IE에서 가져온 북마크 Sebastian Seung's... Parallel Programming... Hal  
주의 요함 | 218.150.182.246:10330  
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ streamlit hello  
/home/work/myfiles/oykwon/packages/pandas/core/computation/expressions.py:21: UserWarning: Pandas requires version 1.4.0 or higher. The currently installed version is 1.3.4.  
from pandas.core.computation.check import NUMEXPR_INSTALLED  
2022-10-12 05:09:11.654 WARNING streamlit.config:  
Warning: the config option 'server.enableCORS=false' is not compatible with 'server.enableXsrfProtection=true'. As a result, 'server.enableCORS' is being overridden to 'true'.  
  
More information:  
In order to protect against CSRF attacks, we send a cookie with each request.  
To do so, we must specify allowable origins, which places a restriction on cross-origin resource sharing.  
  
If cross origin resource sharing is required, please disable server.enableXsrfProtection.  
  
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.  
  
2022-10-12 05:09:11.657 DEBUG matplotlib.pyplot: Loaded backend agg version unknown.  
2022-10-12 05:09:11.657 DEBUG streamlit.watcher.event_based_path_watcher: No singleton. Registering one.  
2022-10-12 05:09:11.760 DEBUG streamlit.watcher.event_based_path_watcher: Watcher created for /home/work/myfiles/oykwon  
2022-10-12 05:09:11.761 DEBUG asyncio: Using selector: EpollSelector  
2022-10-12 05:09:11.761 DEBUG streamlit.web.server.server: Starting server...  
2022-10-12 05:09:11.762 DEBUG streamlit.web.server.server: Serving static content from the Node dev server  
2022-10-12 05:09:11.768 DEBUG streamlit.web.server.server: Server started on port 8501  
2022-10-12 05:09:11.769 DEBUG streamlit.runtime.runtime: Runtime state: RuntimeState.INITIAL -> RuntimeState.NO_SESSIONS_CONNECTED  
2022-10-12 05:09:11.802 DEBUG git.cmd: Popen(['git', 'version'], cwd=/home/work/myfiles/oykwon, universal_newlines=True)  
2022-10-12 05:09:11.809 DEBUG git.cmd: Popen(['git', 'version'], cwd=/home/work/myfiles/oykwon, universal_newlines=True)  
2022-10-12 05:09:11.817 DEBUG git.util: Failed checking if running in CYGWIN due to: FileNotFoundError(2, 'No such file or directory')  
2022-10-12 05:09:11.819 DEBUG urllib3.connectionpool: Starting new HTTP connection (1): checkip.amazonaws.com:80  
2022-10-12 05:09:12.233 DEBUG urllib3.connectionpool: http://checkip.amazonaws.com:80 "GET / HTTP/1.1" 200 16  
  
Welcome to Streamlit. Check out our demo in your browser.  
  
Network URL: http://172.17.0.17:3000  
External URL: http://218.150.182.249:3000  
  
Ready to create your own Python apps super quickly?  
Head over to https://docs.streamlit.io  
  
May you create awesome apps!  
  
2022-10-12 05:09:12.237 DEBUG streamlit.version: Skipping PyPI version check  
2022-10-12 05:09:12.238 DEBUG streamlit.web.bootstrap: Setting up signal handler  
^C Stopping...  
2022-10-12 05:10:49.729 DEBUG streamlit.runtime.runtime: Runtime stopping.  
..  
2022-10-12 05:10:49.730 DEBUG streamlit.runtime.runtime: Runtime state: RuntimeState.NO_SESSIONS_CONNECTED -> RuntimeState.STOPPING  
2022-10-12 05:10:49.730 DEBUG streamlit.runtime.runtime: Runtime state: RuntimeState.STOPPING -> RuntimeState.STOPPED  
work@main1[2EhkeNwg-session]:~/myfiles/oykwon$ [0] 0:*
```

학습정리

- ❖ 인공신경망
 - CNN, LSTM 개념
- ❖ AI에 사용되는 패키지
 - pandas, numpy, scipy, tensorflow, pytorch, keras 등
- ❖ streamlit 활용