

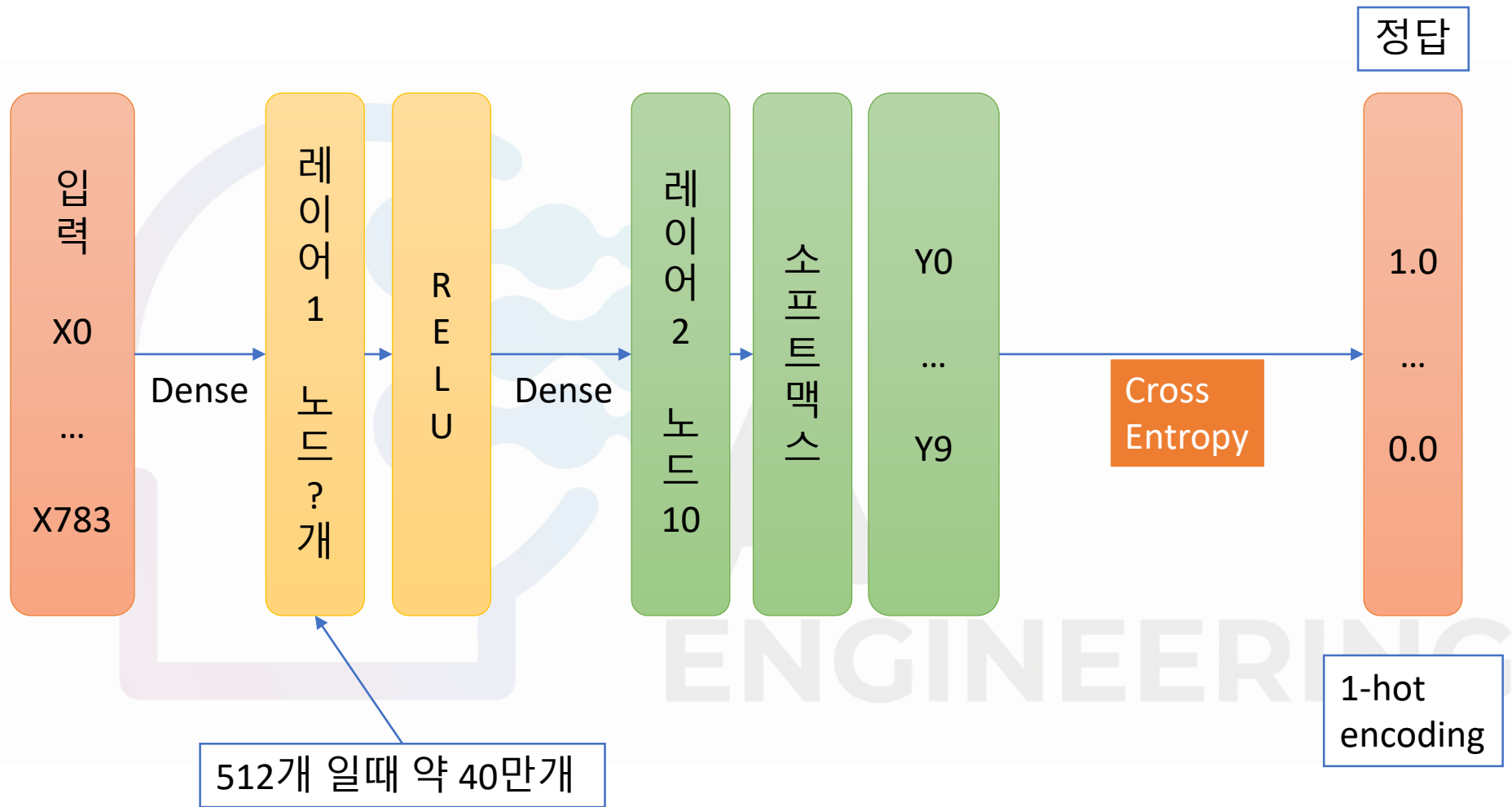
AI+x 인재양성 확대를 위한 교수자 연수

한국폴리텍대학 대구캠퍼스
SI엔지니어링학과 강현우

Convolutional Neural Network

-CNN을 이용하여 기계학습을 진행해본다

지난 시간 MLP 모델



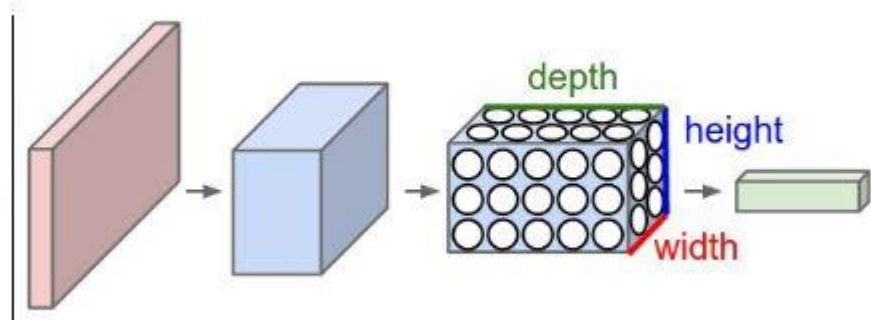
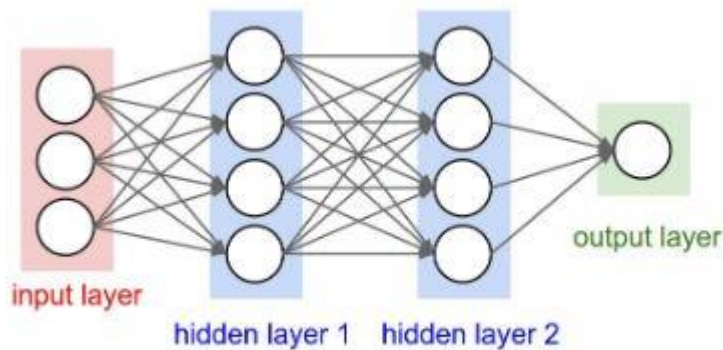
MLP의 문제점

◆ MLP

- 공간 정보가 소실 된다.
- MNIST 28x28 → 784 flatten

◆ Conv Layer

- 형상을 유지한다.



Convolution

◆ 다음 이미지에 3x3 커널을 컨볼루션 해보자

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

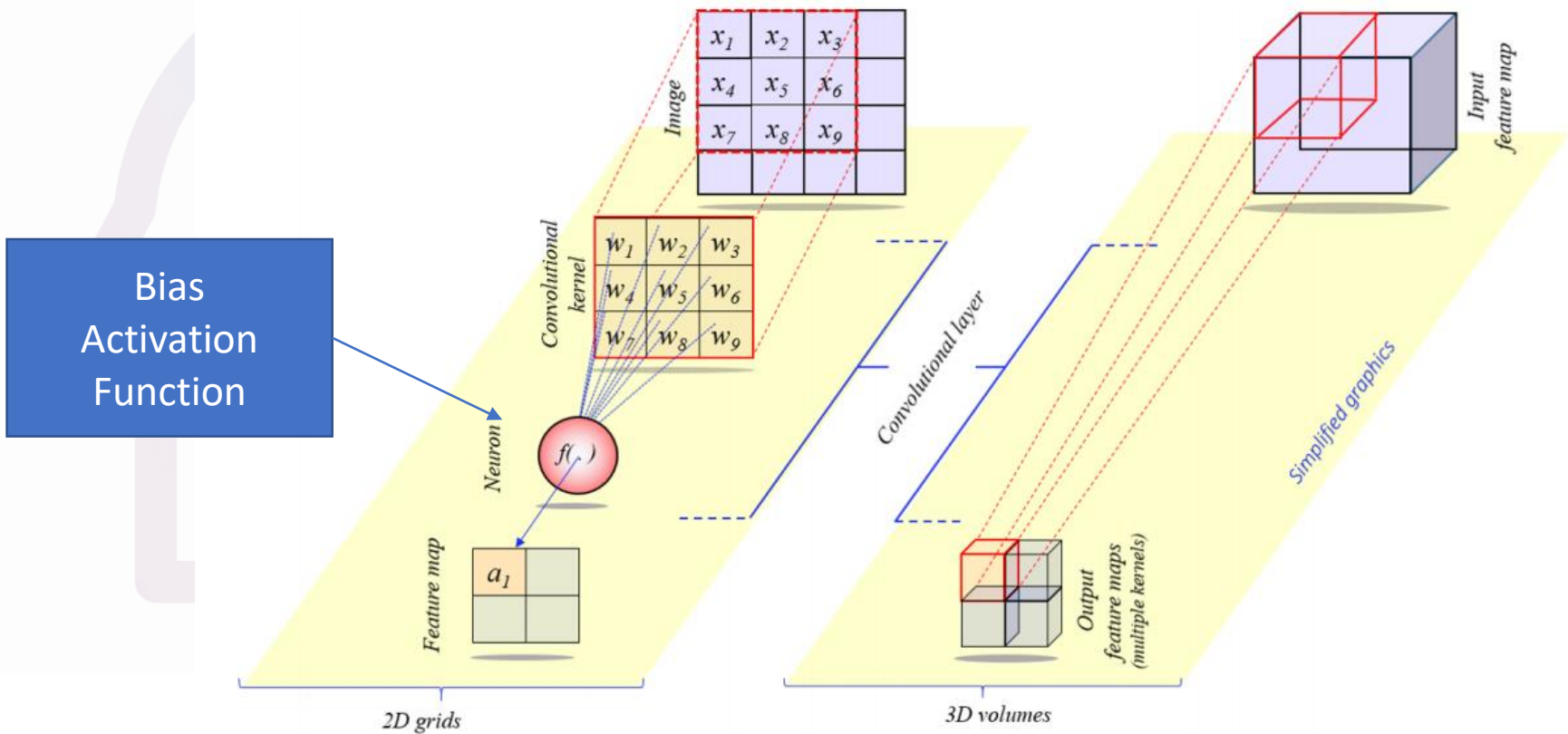
4		

Convolved
Feature

<http://deeplearning.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/>

CNN 에서의 Weight

◆ Locally Connected Networks



CNN 에서의 Weight

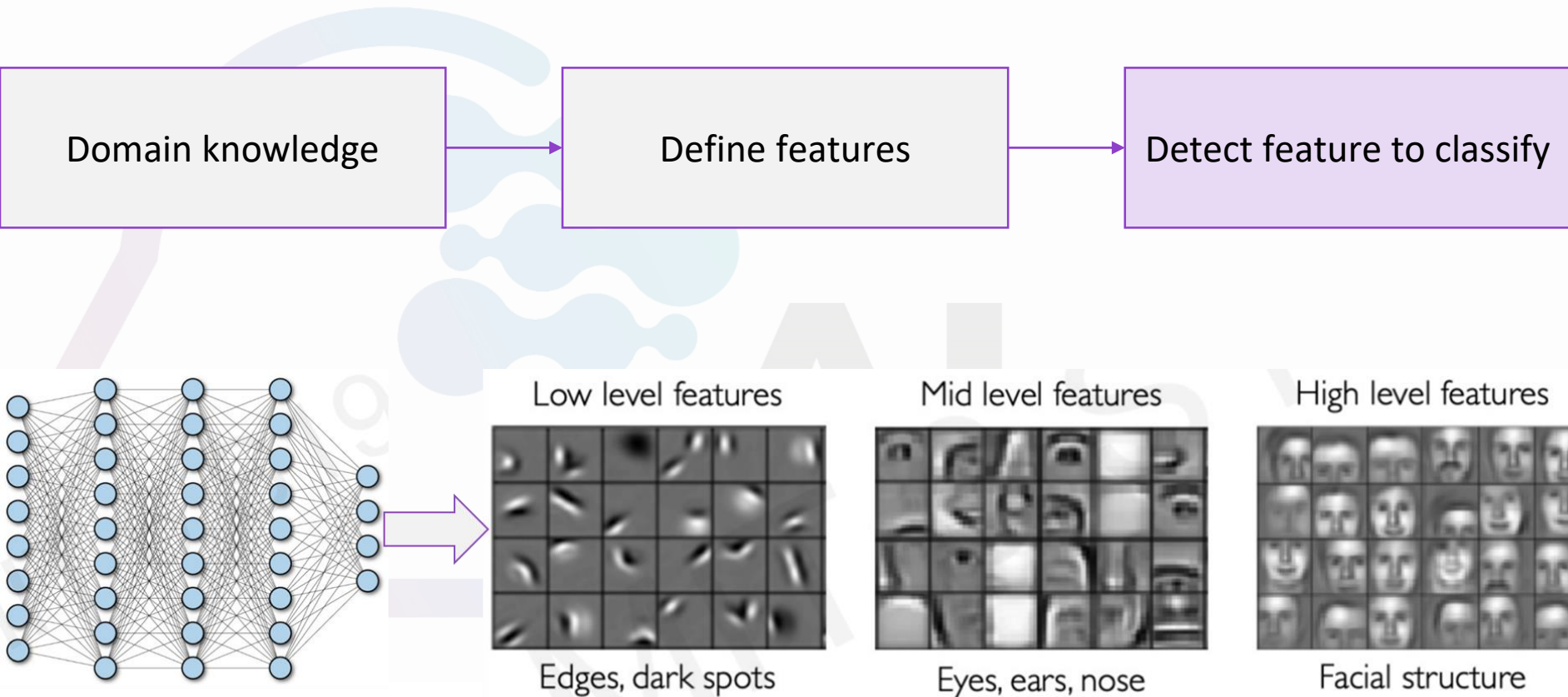
◆ 수용영역 (receptive field)

- MLP처럼 전체 입력과 연결되지 않는다.
- 합성곱 층 뉴런의 수용영역 안의 픽셀만 연결된다.
- convolution 필터가 이미지의 특징을 뽑아내는 특징이 된다!



<https://cs231n.github.io/convolutional-networks/>

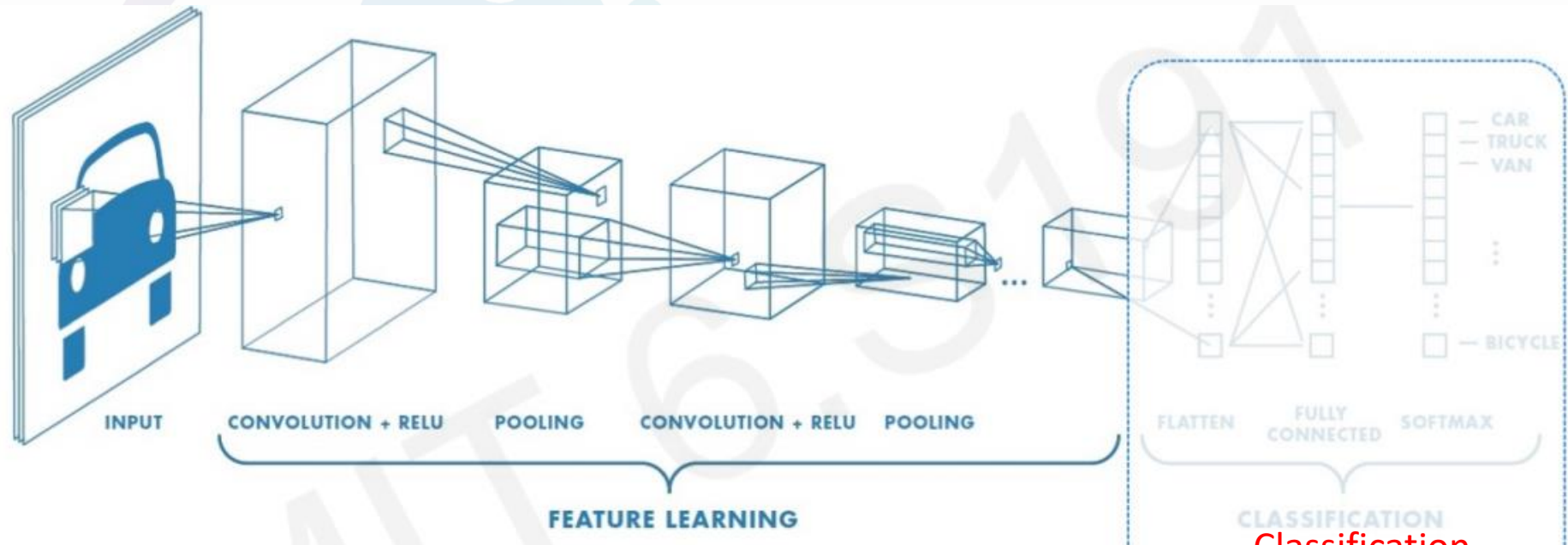
전통적인 기계학습과의 비교



CNN의 표상 학습

◆ Representation Learning

➤ 이미지는 무조건 CNN이라 생각하면 됨

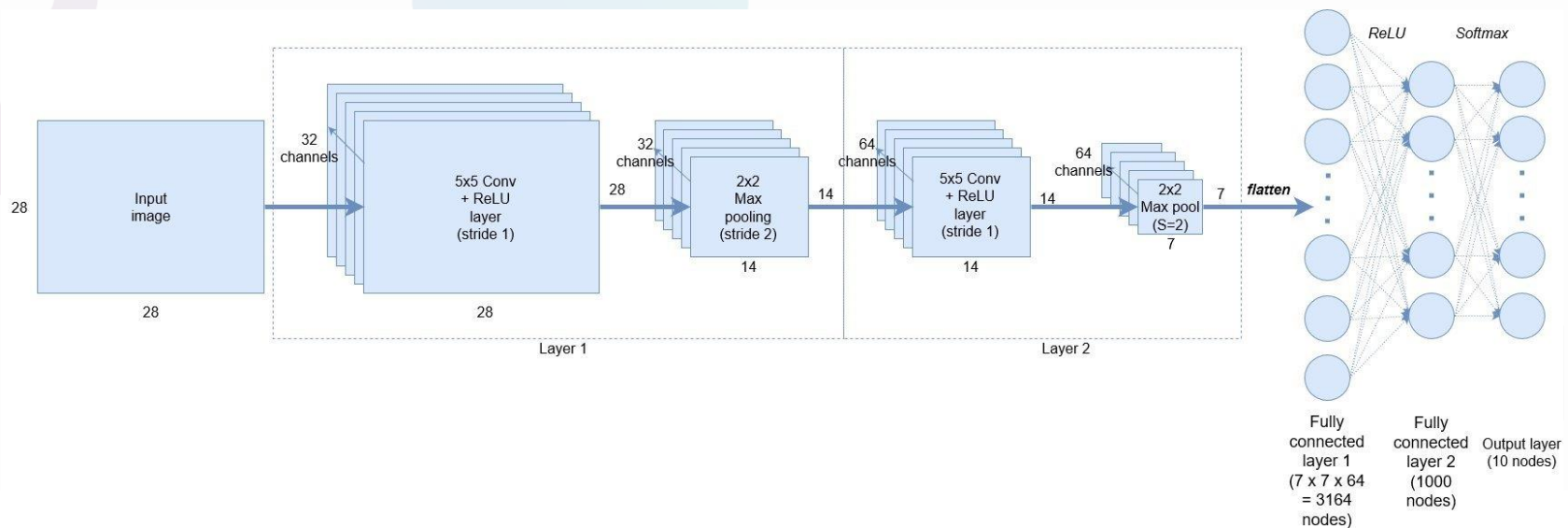


Classification
Object detection
Segmentation
Probabilistic control

TensorFlow CNN 예제

◆ CNN Tutorial in TensorFlow

➤ <https://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/>



Maxpool

◆ Max-pooling

- 해당 영역에서 최대값을 찾는다.
- 데이터의 크기를 줄이는 효과
- 어쩌면 제일 중요한 성분일지도...?
- 평균을 취하는 Average Pooling도 있음

Max-pooling

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	

Stride = 2

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	2

구현

```
import tensorflow as tf

def load_dataset(online=False):
    if online:
        (tr_data, tr_label), (te_data, te_label) = tf.keras.datasets.mnist.load_data()

    else:
        path = "D:/Project/Mnist/dataset/mnist.npz"
        (tr_data, tr_label), (te_data, te_label) = tf.keras.datasets.mnist.load_data(path)

    print("학습 데이터 {0}개 로드".format(tr_data.shape[0]))
    print("테스트 데이터 {0}개 로드".format(te_data.shape[0]))

    return (tr_data, tr_label), (te_data, te_label)

if __name__ == "__main__":

    (train_data, train_label), (test_data, test_label) = load_dataset()
```



Model 구현

```
def make_model():
    model = Sequential()
    model.add(Conv2D(filters=1, kernel_size=(5, 5), strides=(1, 1),
                     activation='relu',
                     input_shape=(28, 28, 1),
                     padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(1, (5, 5), activation='relu', padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Flatten())
    model.add(Dense(1000, activation='relu'))
    model.add(Dense(10, activation='softmax'))

    model.summary()

    model.compile(loss="categorical_crossentropy",
                  optimizer="adam",
                  metrics=["accuracy"])

    return model
```

빠른 테스트를
위해 채널을
1개만 사용

ERING

Model Summary

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 28, 28, 1)	26

max_pooling2d (MaxPooling2D)	(None, 14, 14, 1)	0

conv2d_1 (Conv2D)	(None, 14, 14, 1)	26

max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 1)	0

flatten (Flatten)	(None, 49)	0

dense (Dense)	(None, 1000)	50000

dense_1 (Dense)	(None, 10)	10010
=====		
Total params: 60,062		
Trainable params: 60,062		
Non-trainable params: 0		

학습 구현

```
# 하이퍼 파라미터
MY_EPOCH = 10
MY_BATCHSIZE = 200

def train(model, x, y):
    history = model.fit(x, y, epochs=MY_EPOCH, batch_size=MY_BATCHSIZE)
    filename = "./model/cnn_e({0}).h5".format(MY_EPOCH)
    model.save(filename)

    return history
```

```
if __name__ == "__main__":
    (train_data, train_label), (test_data, test_label) = load_dataset()

    cnn = make_model()

    train_data = train_data.reshape(train_data.shape[0], 28, 28, 1)
    train_label = tf.keras.utils.to_categorical(train_label, 10)

    train(cnn, train_data, train_label)
```

중요!!
28x28x1 이라고
명시 해야함

테스트

```
test_data = test_data.reshape(test_data.shape[0], 28, 28, 1)
test_label = tf.keras.utils.to_categorical(test_label, 10)

filename = "./model/cnn_e({0}).h5".format(MY_EPOCH)
cnn = load_model(filename)
cnn.evaluate(test_data, test_label)
```

313/313 [=====] - 3s 9ms/step
- loss: 0.1568 - accuracy: 0.9488

고작 6만개로 했을 뿐인데?!

채널 숫자 증가

◆ 채널의 숫자를 설계대로!

```
def make_model():
    model = Sequential()
    model.add(Conv2D(filters=32, kernel_size=(5, 5), strides=(1, 1),
                     activation='relu',
                     input_shape=(28, 28, 1),
                     padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(64, (5, 5), activation='relu', padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Flatten())
    model.add(Dense(1000, activation='relu'))
    model.add(Dense(10, activation='softmax'))

    model.summary()

    model.compile(loss="categorical_crossentropy", optimizer="adam",
                  metrics=["accuracy"])

    return model
```

Model Summary

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	832
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	51264
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 1000)	3137000
dense_1 (Dense)	(None, 10)	10010

Total params: 3,199,106
Trainable params: 3,199,106

학습 및 테스트

Epoch 7/7

300/300 [=====] - 52s 175ms/step - loss: 0.0176 -

accuracy: 0.9941

313/313 [=====] - 6s 20ms/step - loss: 0.0587 -

accuracy: 0.9864

◆ 99% 이상의 정확도

- 데이터가 이상한 것이 있어서 100은 나오기 힘들
- overfitting 에 주의할 것

◆ MLP 에 비하여 적은 파라미터

- 지역적 특징을 뽑는 Convolution 이 영상에 적합

Summary

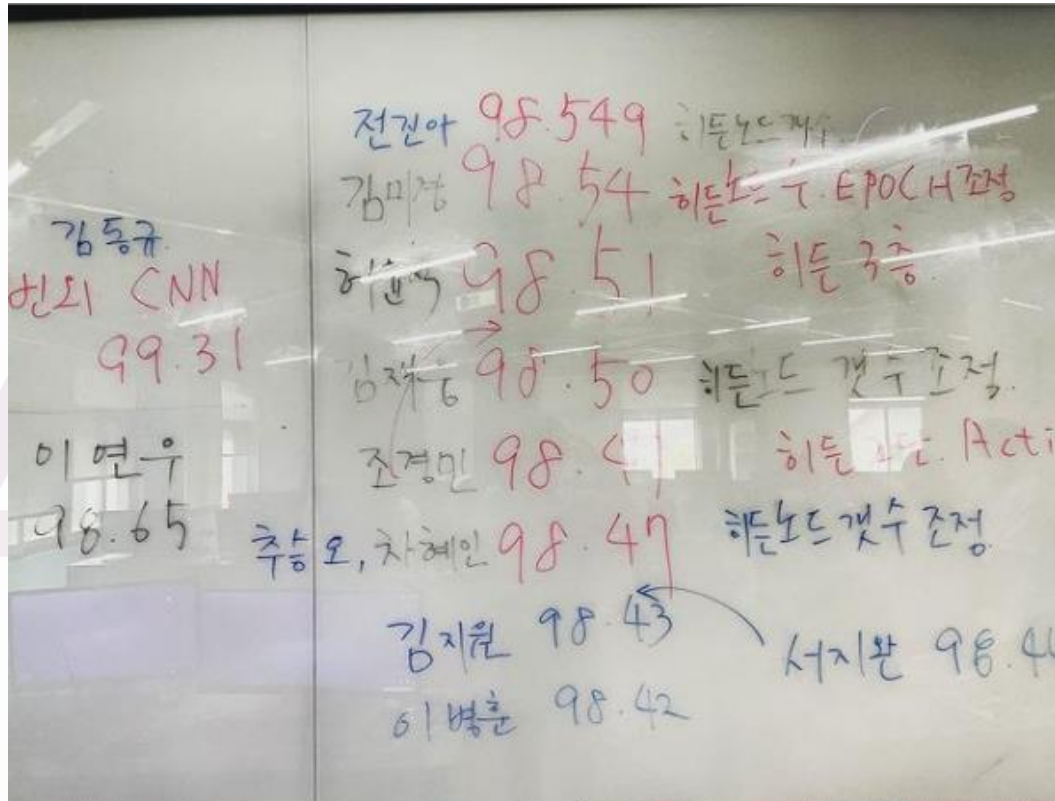
◆ Convolutional Neural Network

- Convolution 필터를 학습된다.
 - ✓ 영상에서 필터가 어떤 의미를 가지는지?
- 지역적으로 연결된 신경망이다.
 - ✓ Fully Connected 된 MLP와 어떤 차이가 있는가?

◆ 참고 사이트

- TensorFlow 튜토리얼
- <https://www.tensorflow.org/tutorials/images/cnn?hl=ko>

Competition



hyunwoo.kang.98

...



hyunwoo.kang.98 #MLP (다층 신경망) 까지만 배운 상태로 #MNIST 손글씨 인식 미니 대회를 학과내에서 실시 하였는데 몇 개 안 배운거 가지고 다들 넘 잘 함

안 가르쳐준 #CNN 쓴 학생도 있었고 시간은 살짝 넘겼지만 1등보다 더 점수 잘 나온 학생도 있었고, 유익한 시간이었던듯 ^^



tae.gyeong_lee님 외 5명이 좋아합니다

4월 20일



댓글 달기...

게시

https://www.instagram.com/p/CN4N4NgFnXY/?utm_medium=copy_link

CIFAR-10

◆ 데이터 셋

- <https://www.cs.toronto.edu/~kriz/cifar.html>
- 32x32 컬러 이미지 60000개

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



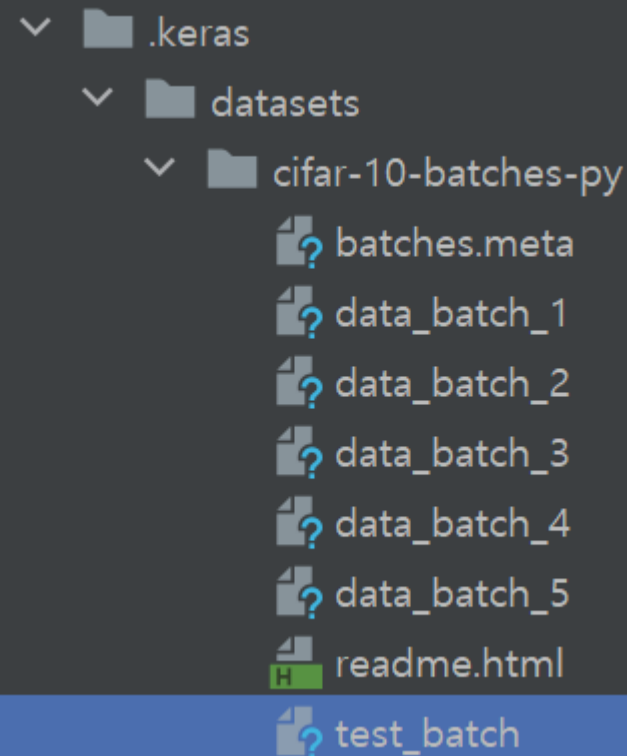
RING

Dataset download

batches.meta	2009-03-31 오후 1:45	META 파일	1K
data_batch_1	2009-03-31 오후 1:32	파일	30,309K
data_batch_2	2009-03-31 오후 1:32	파일	30,308K
data_batch_3	2009-03-31 오후 1:32	파일	30,309K
data_batch_4	2009-03-31 오후 1:32	파일	30,309K
data_batch_5	2009-03-31 오후 1:32	파일	30,309K
readme.html	2009-06-05 오전 5:47	Microsoft Edge H...	1K
test_batch	2009-03-31 오후 1:32	파일	30,309KB

프로젝트 폴더 안에
.keras/datasets
폴더를 생성한다.

압축을 풀어서
.keras/datasets
안에 넣어둔다



Data download and load

인터넷으로 다운
받아도 된다.
첨에 한 번 받는다.

```
import tensorflow as tf
from tensorflow.keras.datasets import cifar10
import matplotlib.pyplot as plt

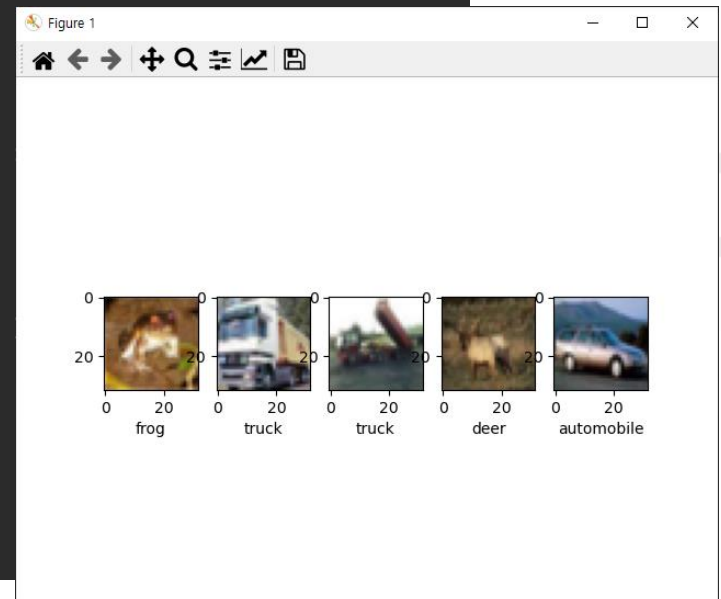
# data 로드. 전역으로 로드한다.
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
print("학습 데이터 {0}개 로드".format(train_images.shape[0]))
print("테스트 데이터 {0}개 로드".format(test_images.shape[0]))

class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse',
               'ship', 'truck']

def show_data_sample():
    for i in range(5):
        plt.subplot(1, 5, i+1)
        plt.imshow(train_images[i])
        plt.xlabel(class_names[train_labels[i][0]])

    plt.show()

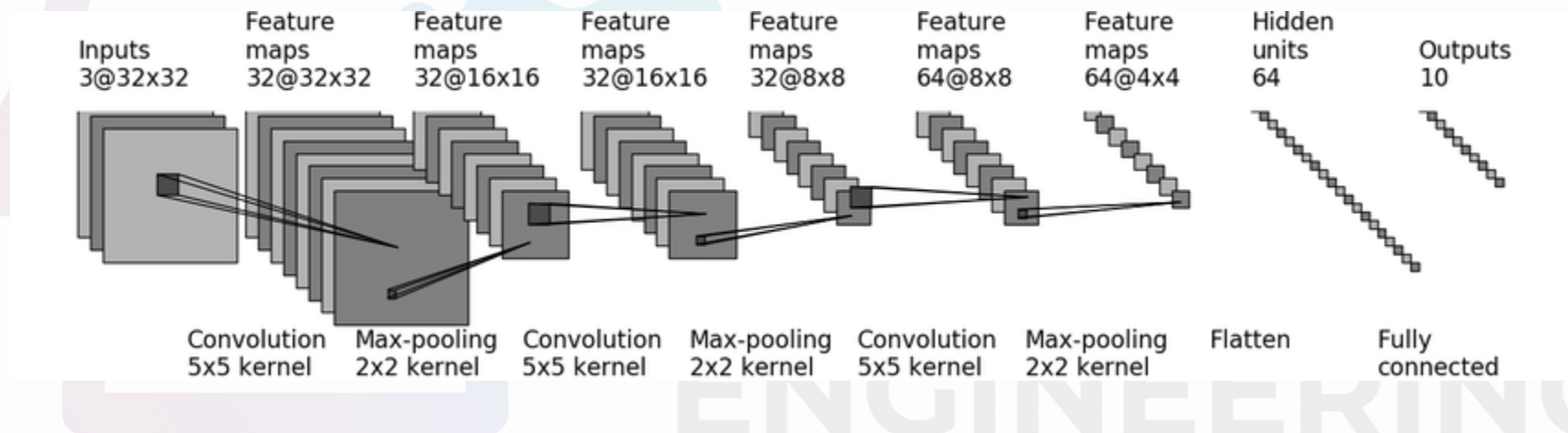
if __name__ == "__main__":
    show_data_sample()
```



모델 설계

◆ Conv + max pool 3회

◆ 64개 hidden node + 10개 출력



모델 구현

```
def make_model():
    model = Sequential()

    model.add(Conv2D(32, (5, 5), activation='relu',
                    input_shape=(32, 32, 3),
                    padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(32, (5, 5), activation='relu', padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(64, (5, 5), activation='relu', padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Flatten())
    model.add(Dense(64, activation='relu'))
    model.add(Dense(10, activation='softmax'))
    model.summary()

    model.compile(loss="categorical_crossentropy",
                  optimizer="adam",
                  metrics=["accuracy"])

    return model
```

ING

Model Summary

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	2432
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_1 (Conv2D)	(None, 16, 16, 32)	25632
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 32)	0
conv2d_2 (Conv2D)	(None, 8, 8, 64)	51264
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 64)	65600
dense_1 (Dense)	(None, 10)	650
Total params: 145,578		
Trainable params: 145,578		
Non-trainable params: 0		

학습 구현

```
# 하이퍼 파라미터
MY_EPOCH = 10
MY_BATCHSIZE = 200
filename = f"./model/cnn_e({MY_EPOCH}).h5"

def train(model, x, y):
    x = x.astype("float32")
    x = x / 255
    y = tf.keras.utils.to_categorical(y, 10)
    history = model.fit(x, y, epochs=MY_EPOCH, batch_size=MY_BATCHSIZE)
    model.save(filename)

    return history
```

Epoch 9/10

250/250 [=====] - 47s 190ms/step - loss: 0.6965 -
accuracy: 0.7577

Epoch 10/10

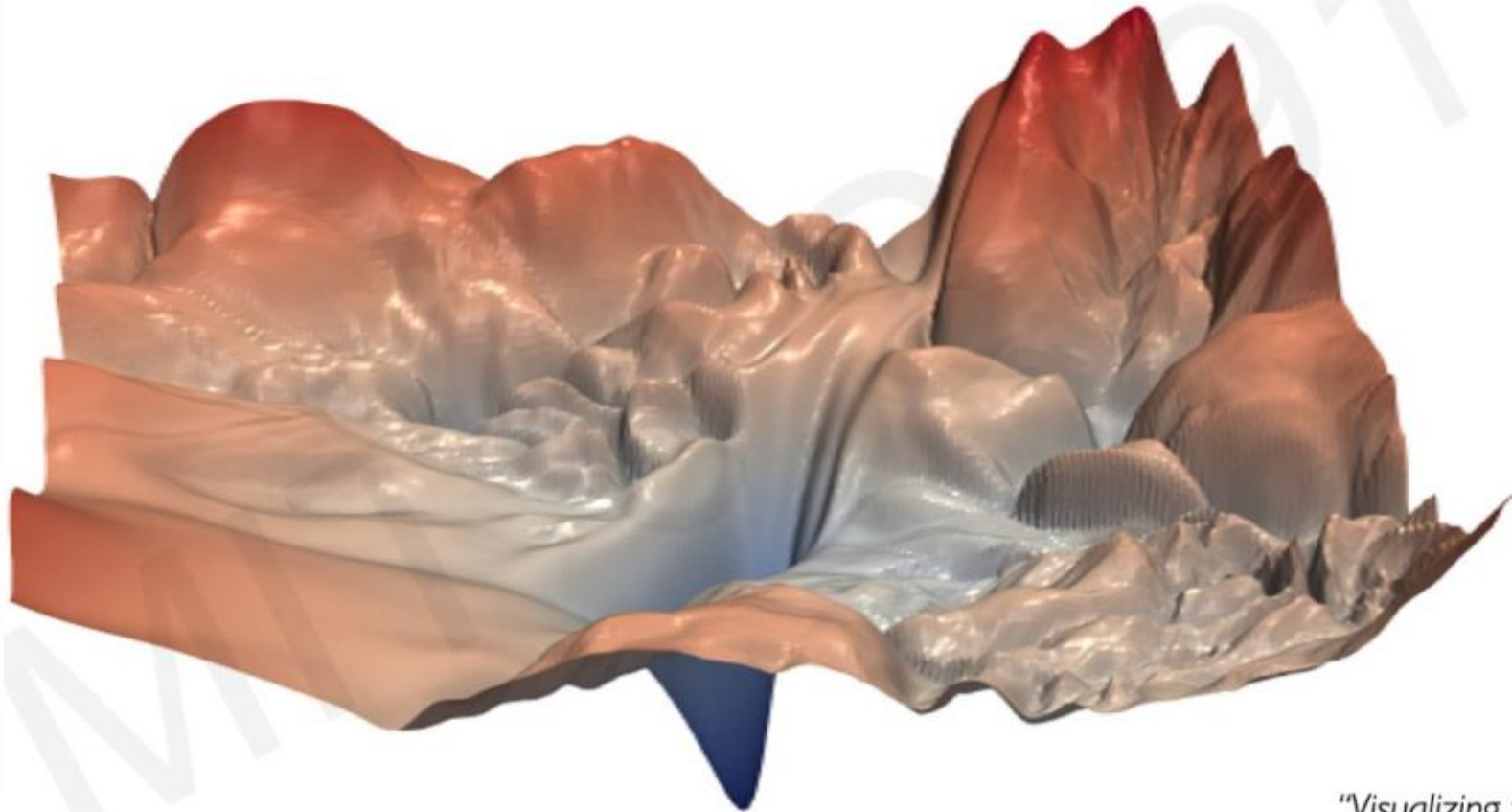
250/250 [=====] - 47s 187ms/step - loss: 0.6459 -
accuracy: 0.7757

테스트

```
def test_all(x, y):  
    model = load_model(filename)  
    x = x / 255.  
    y = tf.keras.utils.to_categorical(y, 10)  
    model.evaluate(x, y)
```

```
if __name__ == '__main__':  
    print(train_images.shape)  
    print(train_labels.shape)  
  
    cnn = make_model()  
    #train(cnn, train_images, train_labels)  
  
    test_all(test_images, test_labels)
```

최적화의 어려움



"Visualizing the loss landscape of neural nets". Dec 2017.

Batch normalization

◆ 가중치 계산 시

1. 기울기 계산, $\frac{\partial J_i(W)}{\partial W}$

2. 가중치 갱신, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$

한번에 하나의 데이터만 처리한다면,
계산은 쉽지만 수렴이 어려움

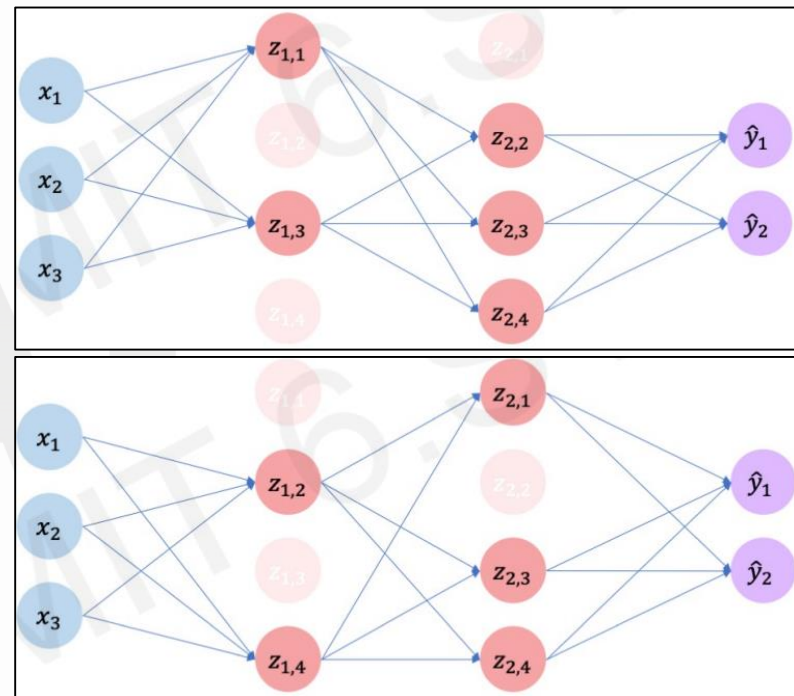
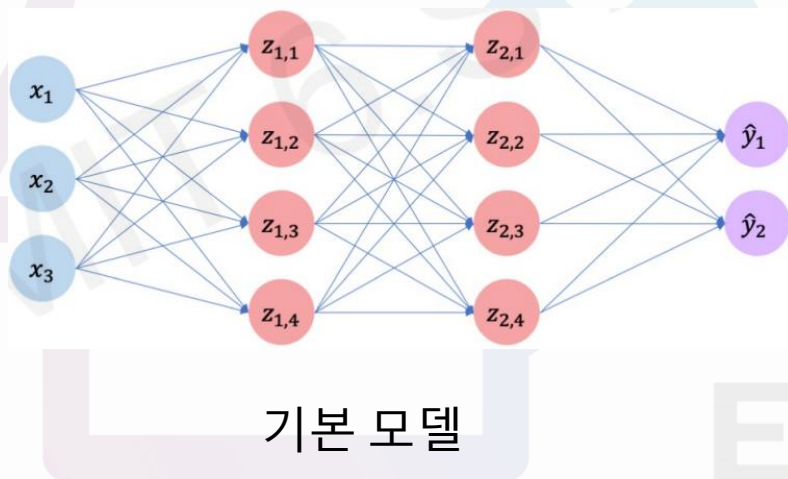
일정 개수의 데이터 묶음 단위로 처리한다면,
계산도 가능하고 수렴이 잘 될 수 있음

◆ Batch의 정규화

➤ 어려워서 패스...

Drop out

◆ 각 층마다 임의로 출력을 제거



학습 과정에서의 드롭아웃 된 모델

모델 변경

```
def make_model():
    model = Sequential()
    # 1
    model.add(Conv2D(filters=32, kernel_size=(5, 5), activation='relu',
                     input_shape=(32, 32, 3), padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Dropout(0.25))

    # 2
    model.add(Conv2D(32, (5, 5), activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Dropout(0.25))

    # 3
    model.add(Conv2D(64, (5, 5), activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Dropout(0.25))

    # 4
    model.add(Flatten())
    model.add(Dense(64, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(10, activation='softmax'))
```

각 출력 계층에
추가해보고
다시 학습 해본다!

EERING



학습 최적화

◆ 검증 데이터 오류를 잘 활용한다.

➤ 학습 조기 종료!

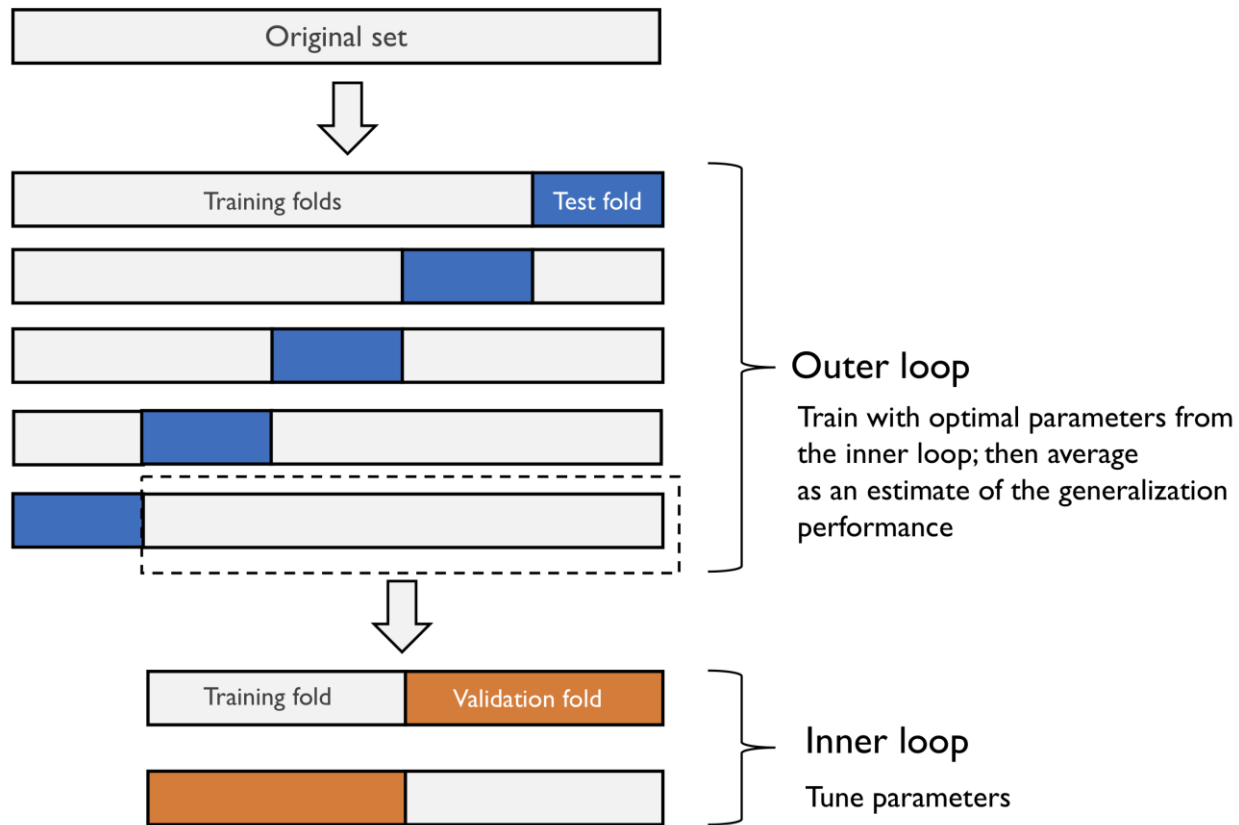


K-fold 테스트

◆ Training

◆ Validation

◆ Test



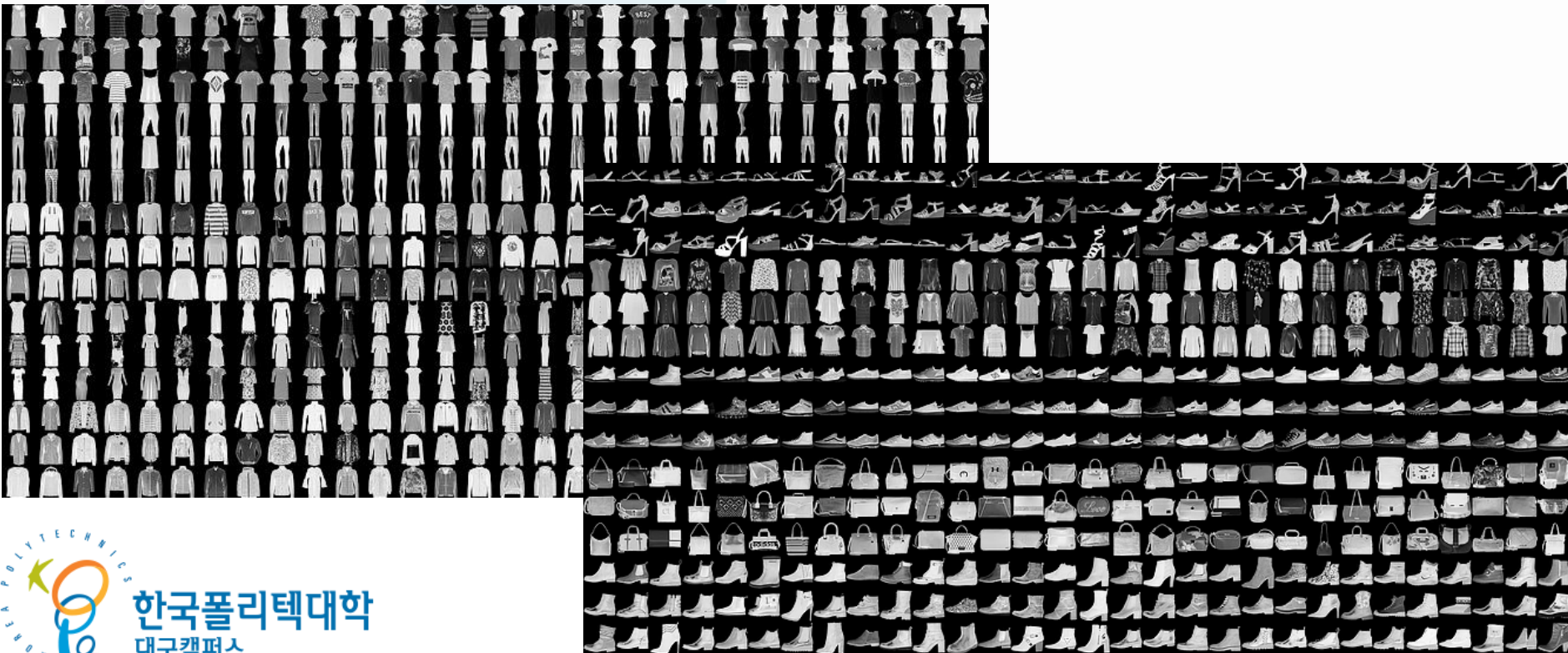
This work by Sebastian Raschka is licensed under a Creative Commons Attribution 4.0 International License.

[출처] <https://sebastianraschka.com/blog/2018/model-evaluation-selection-part4.html>

Mini Project

◆ Fashion MNIST

- 10개 범주, 70,000개 흑백 이미지 (28×28)
- T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankel boot



모두의 딥러닝



따라 하면 되는
예제가 많다!

강의 자료가 충실하다.

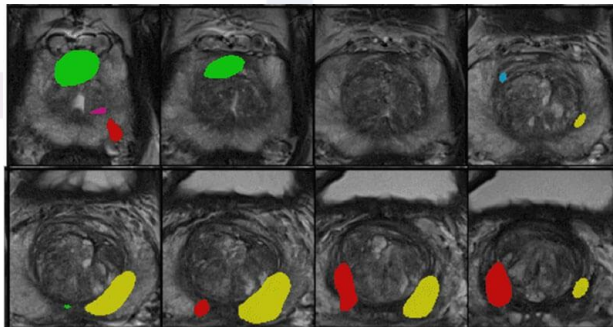
그러나 현실은...

◆ 데이터가 없다

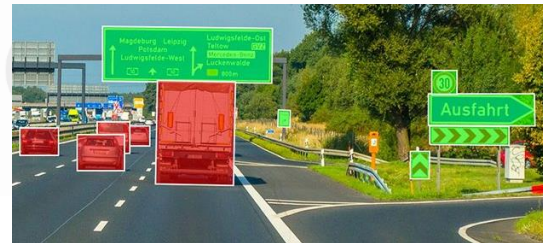
- 현실에서 우리가 만나는 문제들은 잘 정제된 데이터가 없다.

◆ 데이터는 가치가 있다.

- 만드는 데 힘이 들기 때문
- 데이터 레이블을 위한 도구들도 많이 존재



[출처] medium



[출처] clickworker

Summary

◆ CNN

- Convolution Filter 를 학습
- 이미지 학습에 적합

◆ 최적화의 어려움

- 많은 시행 착오를 겪어야 한다.

◆ 데이터!

- 연습을 위한 데이터는 많다.
- 내 문제에 필요한 데이터는 내가 만들 수 밖에...