The "Hello World" of machine learning

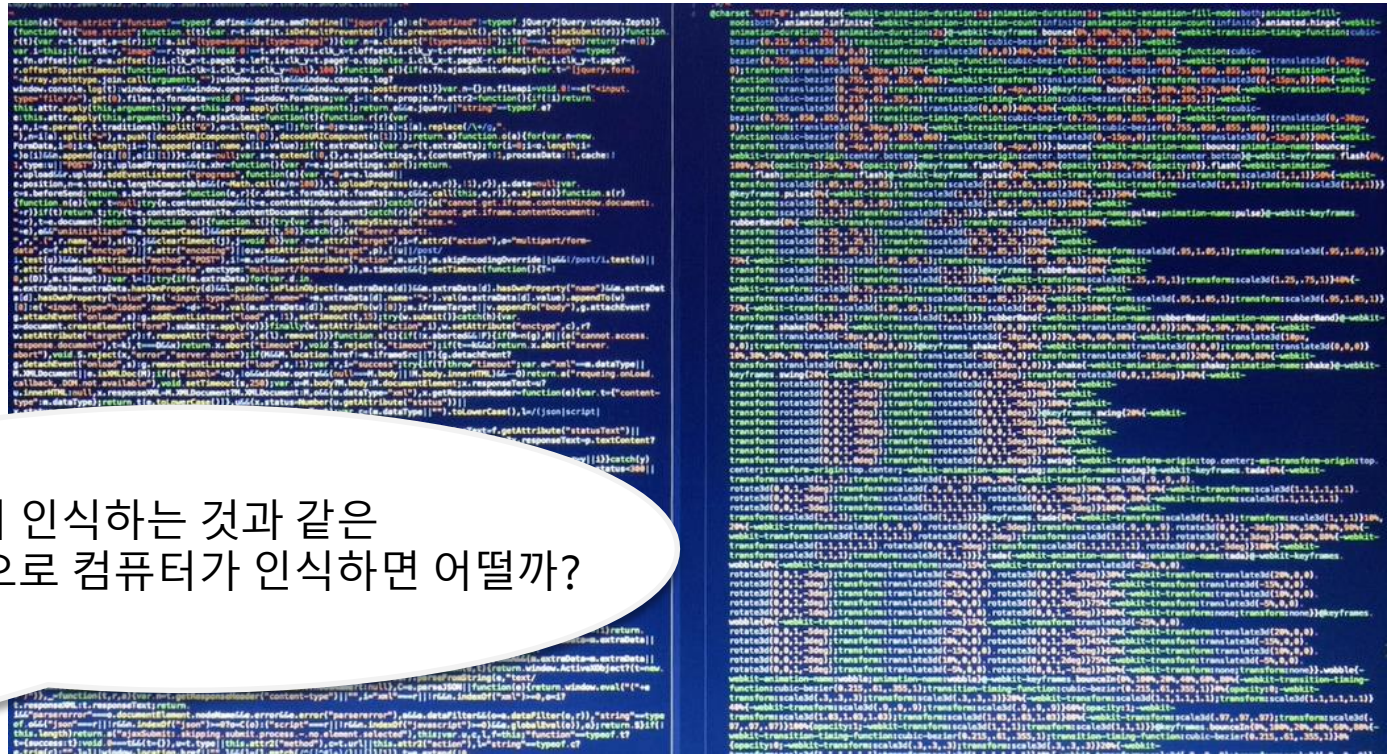
JEJ

context

- Intro to Machine Learning
- Basic computer vision with ML
- Introducing convolutional neural networks
- Build an image classifier for rock,paper,scissors

Intro to Machine Learning

Intro to Machine Learning



사람이 인식하는 것과 같은
방식으로 컴퓨터가 인식하면 어떨까?

Intro to Machine Learning

사람이 인식하는 것과 같은 방식으로
컴퓨터가 인식하면 어떨까?

머신러닝의 핵심, 인공지능이 추구하는 것

Intro to Machine Learning

동그란것= 묵..
손가락 세개 또는 두개=가위..
등등..

Intro to Machine Learning

동그란것= 묵..
손가락 세개 또는 두개=가위..
등등..

Rules

Data

Traditional Programming

Answers

묵,찌,빠

묵,찌,빠

Answers

Data

Machine Learning

Rules

동그란것= 묵..
손가락 세개 또는 두개=가위..
등등..

X= -1, 0, 1, 2, 3, 4

Y= -2, 1, 4, 7, 10, 13

Intro to Machine Learning

X= -1, 0, 1, 2, 3, 4
Y= -2, 1, 4, 7, 10, 13

Y= 3X + 1

Intro to Machine Learning

모델 정의

모델 컴파일

x 데이터
y 데이터

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
# 한개의 레이어가 있고, 그 레이어는 한개의 뉴런을 지녔고, 인풋쉐입은 1이다.

model.compile(optimizer='sgd', loss='mean_squared_error')
# 추측 -> 평가 -> 추측 -> 평가....
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

model.fit(xs, ys, epochs=500)
# x값을 y의 값에다가 핏시킴

print(model.predict([10.0]))
# x에 10이 들어가면 뭐가 나올까?
```

모델이란? 학습된 뉴런 네트워크

Intro to Machine Learning

```python
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
# 한개의 레이어가 있고, 그 레이어는 한개의 뉴런을 지녔고, 인풋쉐입은 1이다.

model.compile(optimizer='sgd', loss='mean_squared_error')
# 추측 -> 평가 -> 추측 -> 평가....
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

model.fit(xs, ys, epochs=500)
# x값을 y의 값에다가 핏시킴

print(model.predict([10.0]))
# x에 10이 들어가면 뭐가 나올까?
```

31가 나와야되는데 실제로는
31.00321가 나온다 왜일까?

Intro to Machine Learning

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
# 한개의 레이어가 있고, 그 레이어는 한개의 뉴런을 지녔고, 인풋쉐입은 1이다.

model.compile(optimizer='sgd', loss='mean_squared_error')
# 추측 -> 평가 -> 추측 -> 평가....
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

model.fit(xs, ys, epochs=500)
# x값을 y의 값에다가 핏시킴

print(model.predict([10.0]))
# x에 10이 들어가면 뭐가 나올까?
```

훈련데이터가 6쌍 뿐이기 때문

두 관계가 일직선처럼 보이지만

그 밖의 값은 일직선이 아닐수도 있는 확률이 존재

Intro to Machine Learning

Hello ML world

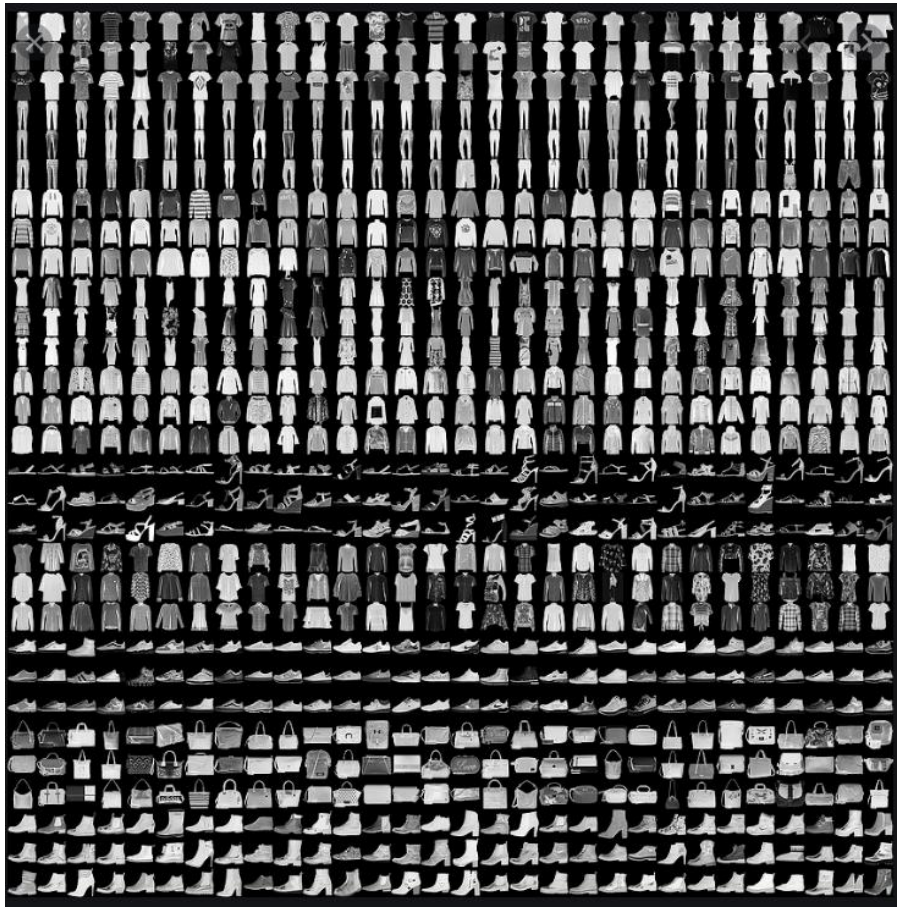
https://colab.research.google.com/drive/1eS0IaRdvRMejyh1jYxj12zsj6YEuDAG-

Basic computer vision with ML

신발은 몇개일까?

Basic computer vision with ML

Basic computer vision with ML



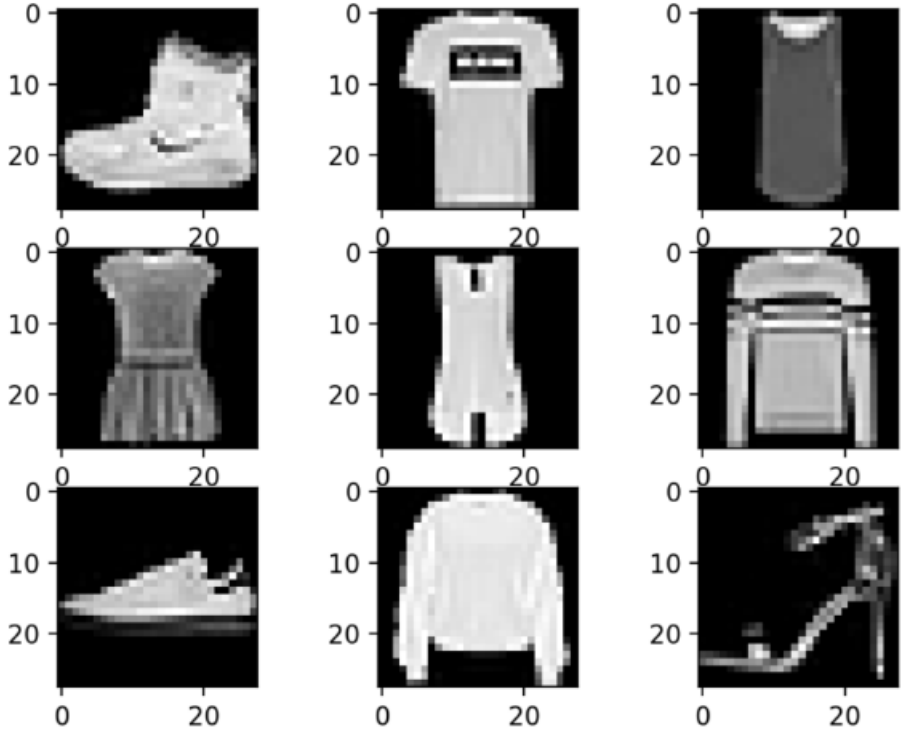
0 T-shirt/top
1 Trouser
2 Pullover
3 Dress
4 Coat
5 Sandal
6 Shirt
7 Sneaker
8 Bag
9 Ankle boot

Fashion MNIST

- 총 70000개의 이미지
- 10개의 카테고리
- 28X28 크기의 이미지

https://github.com/zalandoresearch/fashion-mnist

Basic computer vision with ML



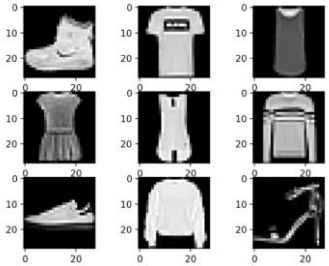
0 T-shirt/top
1 Trouser
2 Pullover
3 Dress
4 Coat
5 Sandal
6 Shirt
7 Sneaker
8 Bag
9 Ankle boot

Fashion MNIST

- 총 70000개의 이미지
- 10개의 카테고리
- 28X28 크기의 이미지

Basic computer vision with ML

```
import tensorflow as tf
from tensorflow import keras
# fashion_mnist 데이타는 tensorflow에 내장되어있어서
fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels),(test_images, test_labels) = fashion_mnist.load_data()
# 쉽게 불러올 수 있다
```

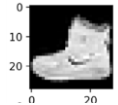


0 T-shirt/top
1 Trouser
2 Pullover
3 Dress
4 Coat
...

훈련 이미지는 60,000개

테스트 이미지는 10,000개

Basic computer vision with ML

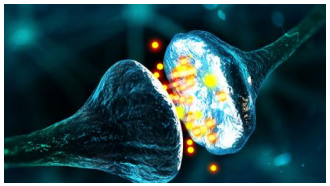



```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(input_shape=(28,28)),
                                    tf.keras.layers.Dense(128,  activation='relu'),
                                    tf.keras.layers.Dense(10, activation='softmax')])
```

0 T-shirt/top
1 Trouser
2 Pullover
3 Dress
4 Coat
...
10

Basic computer vision with ML

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(input_shape=(28,28)),
                                    tf.keras.layers.Dense(128,  activation='relu'),
                                    tf.keras.layers.Dense(10, activation='softmax')])
```

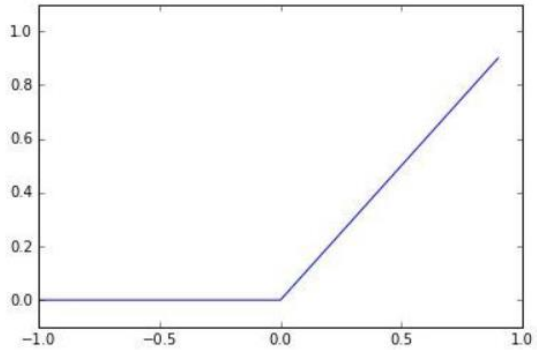


Firing of a neuron

Activation function(활성화 함수)
Input data -> {activation function} -> output data

<ReLU>

```
if (x>0){
    return x;
}
else{
    return 0;
}
```

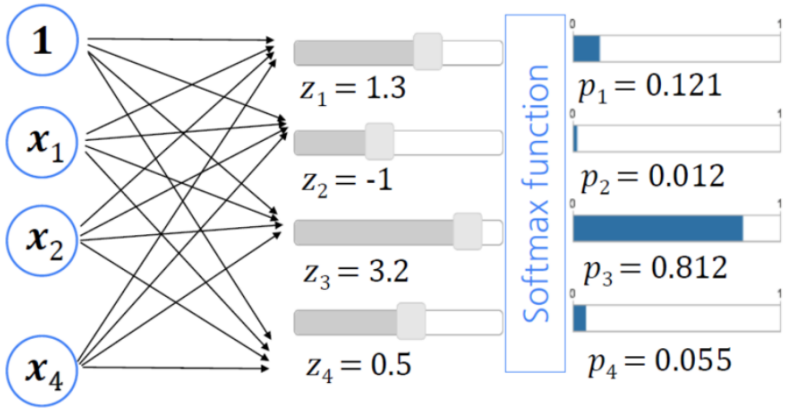



```
import numpy as np
def relu(x):
    return np.maximum(0, x)
```

Basic computer vision with ML

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(input_shape=(28,28)),
                                    tf.keras.layers.Dense(128,  activation='relu'),
                                    tf.keras.layers.Dense(10, activation='softmax')])
```

```
import numpy as np
def softmax(arr):
    arr = np.exp(arr)
    return arr / np.sum(arr)
```



0.121
0.012
0.812
+ 0.055
1

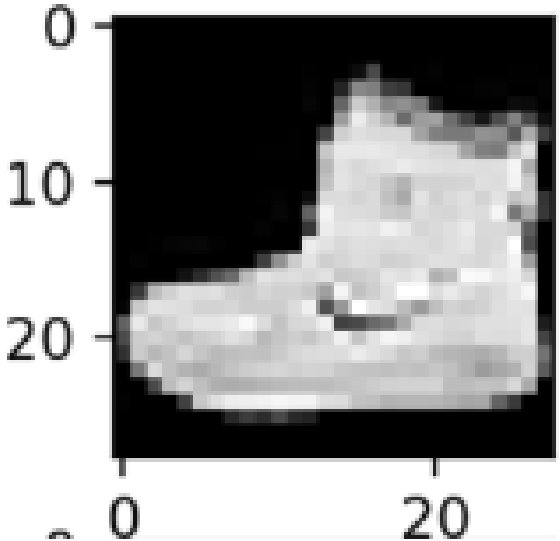
편차란? 관측값과 평균의 차이
Softmax는 최종 출력 단에서 N가지 범주로 분류하는 Multi-class classification 에 쓰임

Basic computer vision with ML

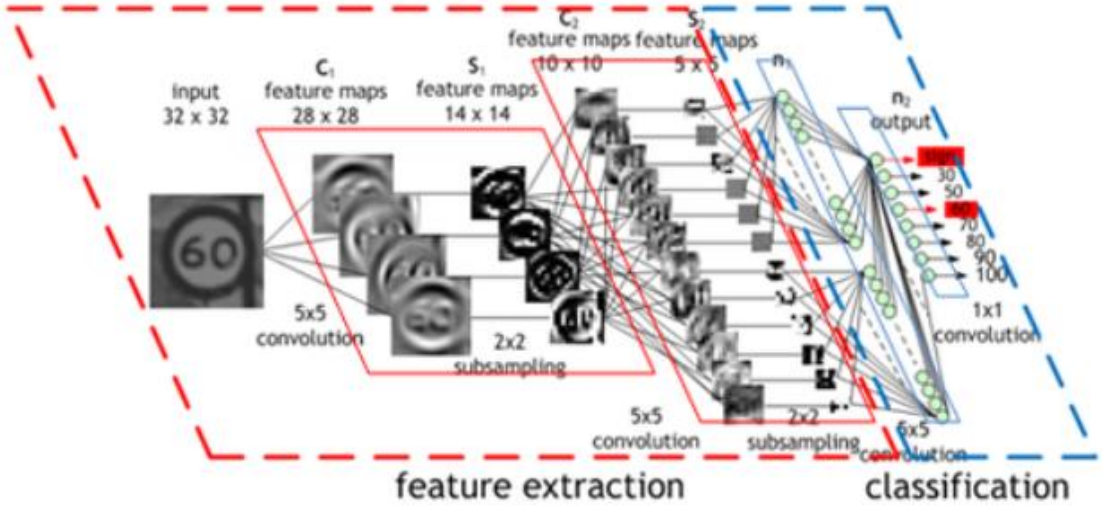
2_Computer_Vision_FashionMNIST_end

https://colab.research.google.com/drive/1eRRtGqB0396cqkrdCCWBukVJ433tVLWJ

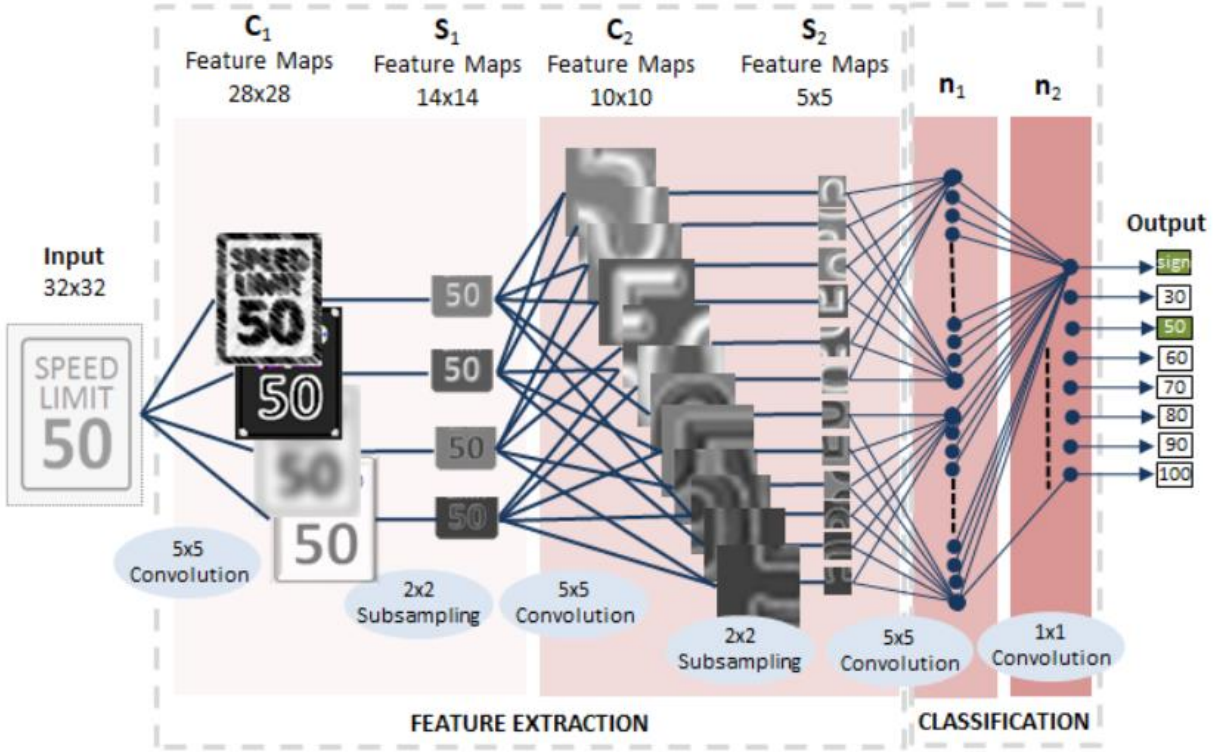
convolutional neural networks



convolutional neural networks



convolutional neural networks



convolutional neural networks

```
model = tf.keras.models.Sequential([

                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(128,  activation='relu'),
                                    tf.keras.layers.Dense(10, activation='softmax')])
```

convolutional neural networks

```
model = tf.keras.models.Sequential([
                                    tf.keras.layers.Conv2D(64,(3,3), activation='relu',
                                                           input_shape=(28,28,1)),
                                    tf.keras.layers.MaxPooling2D(2,2),

                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(128,  activation='relu'),
                                    tf.keras.layers.Dense(10, activation='softmax')])
```

convolutional neural networks

풀링 (Sub sampling or Pooling)

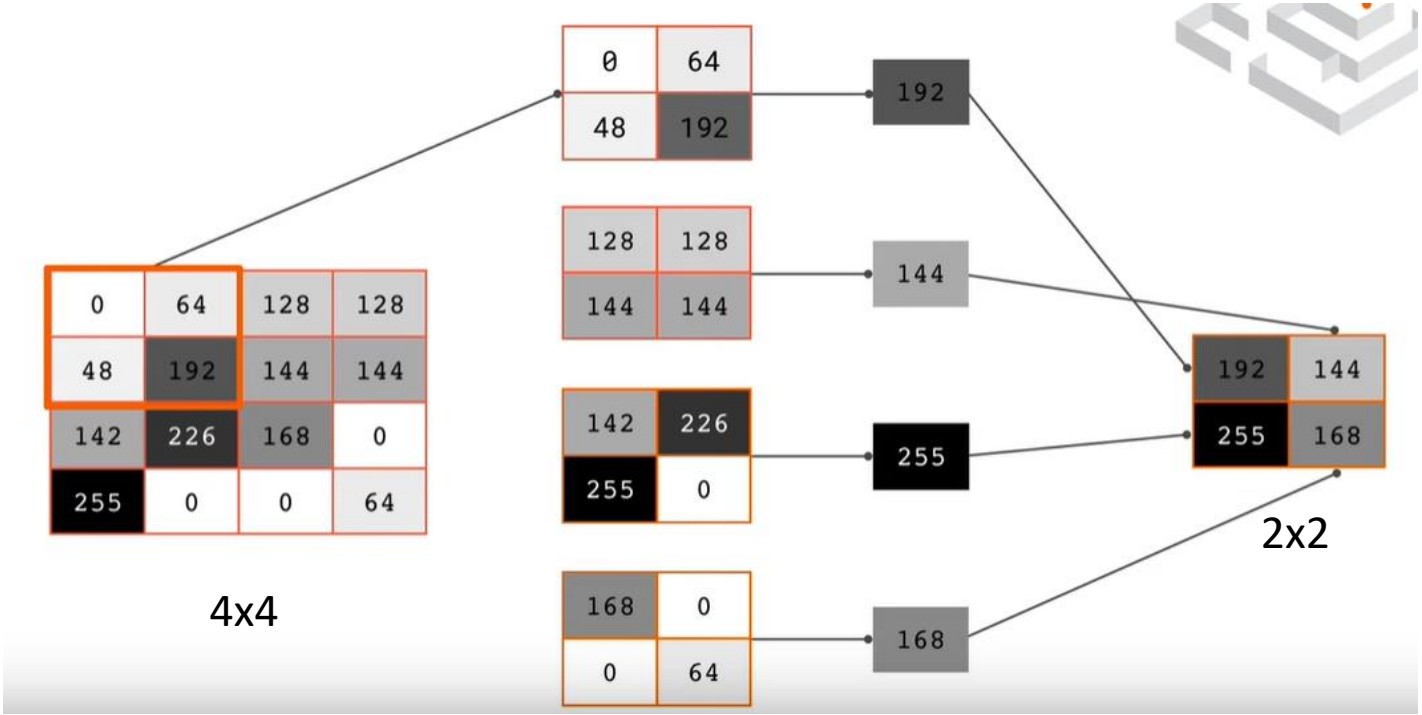
컨볼루셔널 레이어를 통해 어느정도 특징이 추출되었으면 이 모든 특징을 가지고 판단을 할 필요가 없다. (고해상도 저해상도 사진)
그래서, 인위로 줄이는 작업을 하는데, 이 작업을 sub sampling 도는 pooling 이라고 한다.

[Pooling의 종류]
- Max pooling
- Average pooling,

등이 있고, 그중에서 max pooling 이라는 기법이 많이 사용된다.
이미지의 크기는 줄어들지만 특징들은 강조되게 된다.

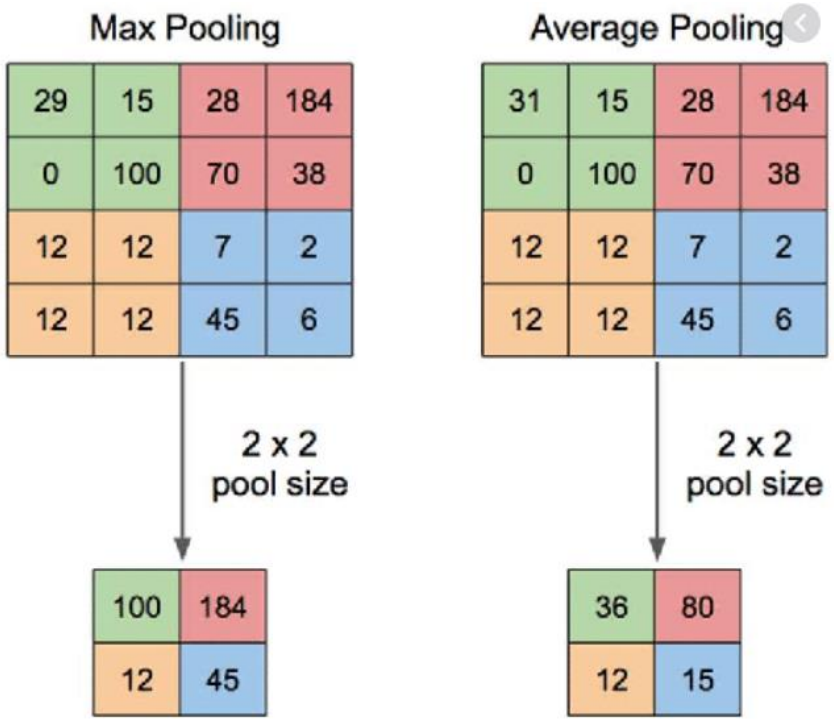
convolutional neural networks

Max pooling(맥스풀링) : 가장 큰 값을 뽑아냄



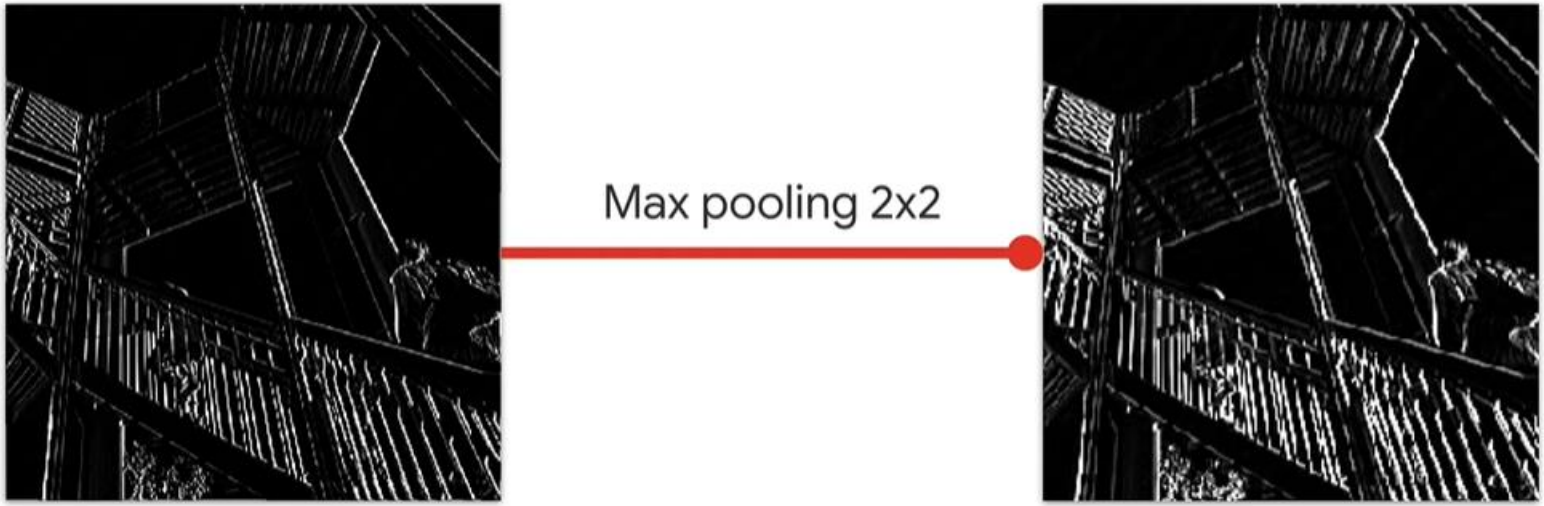
convolutional neural networks

Average pooling(에버리지풀링) : 평균 값을 뽑아냄



(31+15+0+100)/4 =36
(28+184+70+38)/4=80

convolutional neural networks



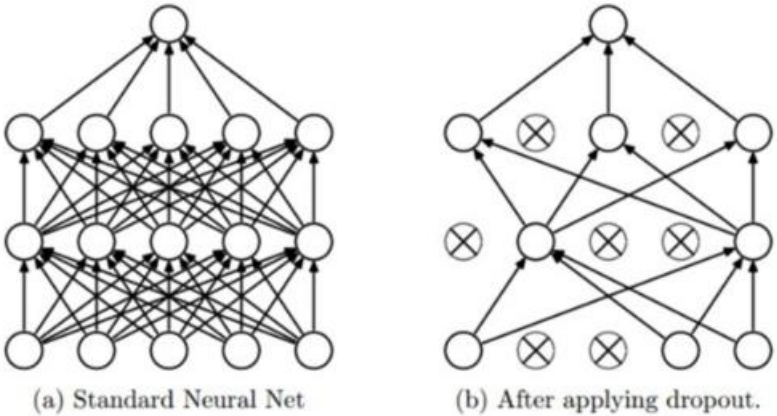
convolutional neural networks



https://colab.research.google.com/drive/1yw9rwtDNo45mJ1XcVT_8W2hG--9Gt2vt

3_cnn

Drop out : 드롭 아웃은 오버피팅(over-fit)을 막기 위한 방법으로 뉴럴 네트워크가 학습중일때, 랜덤하게 뉴런을 꺼서 학습을 방해함으로써, 학습이 학습용 데이타에 치우치는 현상을 막아준다.



(a) Standard Neural Net

(b) After applying dropout.

Hyperparameters:

f : filter size

s : stride

Max or average pooling

~~p : padding~~

f=2, s=2

f=3, s=2

$n_H \times n_w \times n_c$

$\downarrow$

$\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_w - f}{s} + 1 \right\rfloor$

$\times n_c$