Machine learning 08

Byung Chang Chung

Gyeongsang National University

bcchung@gnu.ac.kr

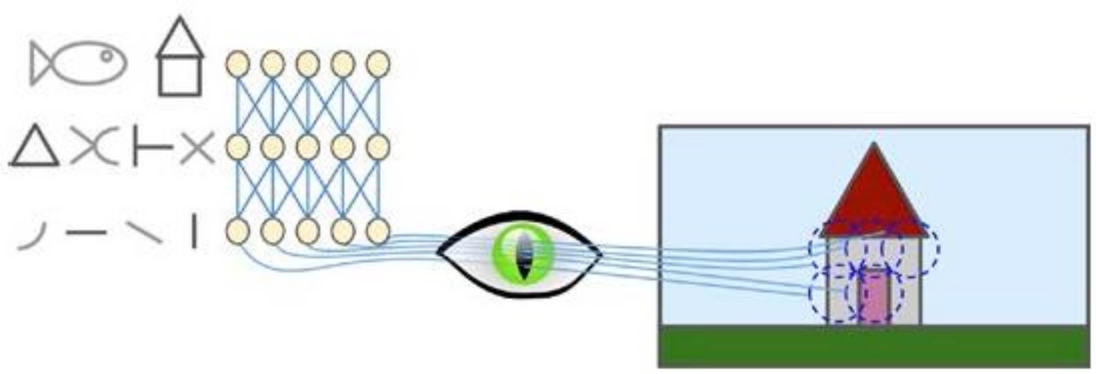
Contents

- Convolutional neural network (CNN)
  - basic intuition
  - convolutional layer
  - pooling layer
  - entire structure
  - implementation

Gyeongsang National University 경상국립대학교

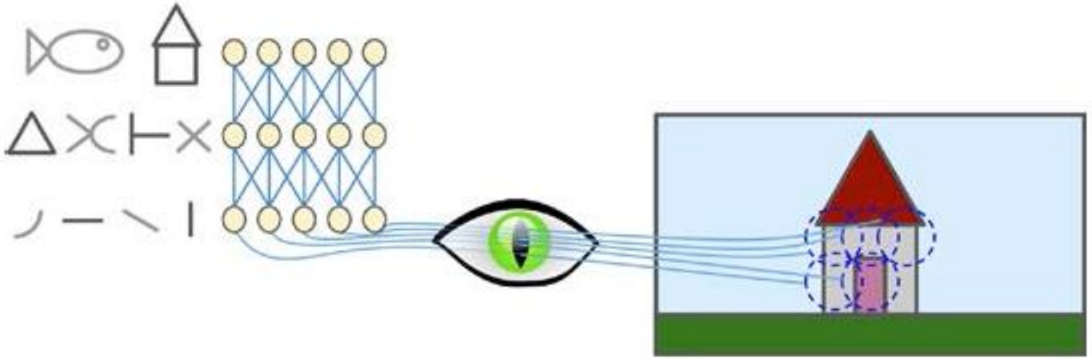
Bio-inspired structure

- Vision of human
  - lots of local receptive field
  - some neurons are activated by horizontal line, some neurons are activated by vertical line, and so on



Bio-inspired structure

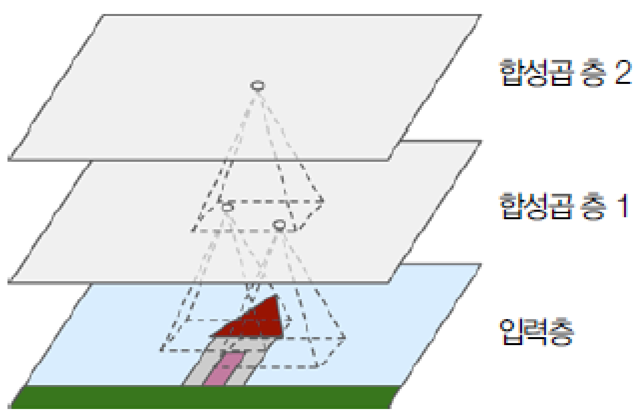
- Vision of human
  - some neurons are activated by the combination of low-level pattern (horizonal, vertical, and so on)
  - high-level neurons are activated by the output of low-level neurons



Gyeongsang
National
University
경상국립대학교

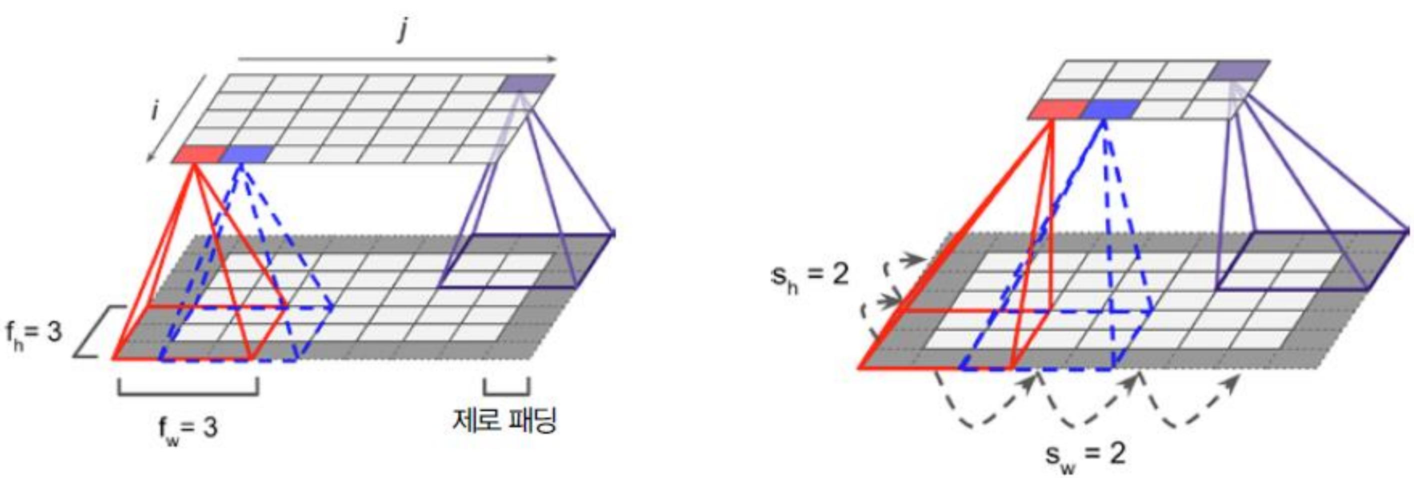
Convolutional layer

- Core part of CNN
  - the neurons in the first convolutional layer do not connect to all the pixels in the input image, but only to the pixels in the local receptable field



Convolutional layer

- Single convolutional layer
  - same convolution kernel moves by its strides
  - for edge, zero padding is used

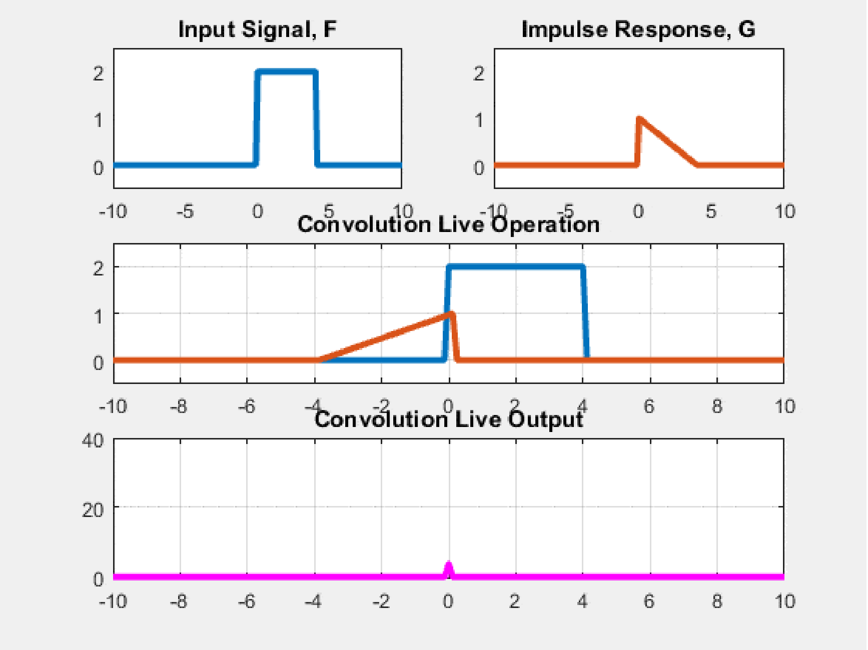


Convolutional layer

- Convolution
  - convolution is a mathematical operation on two functions (f and g) that produces a third function $(f \circ g)$ that expresses how the shape of one is modified by the other

Convolutional layer

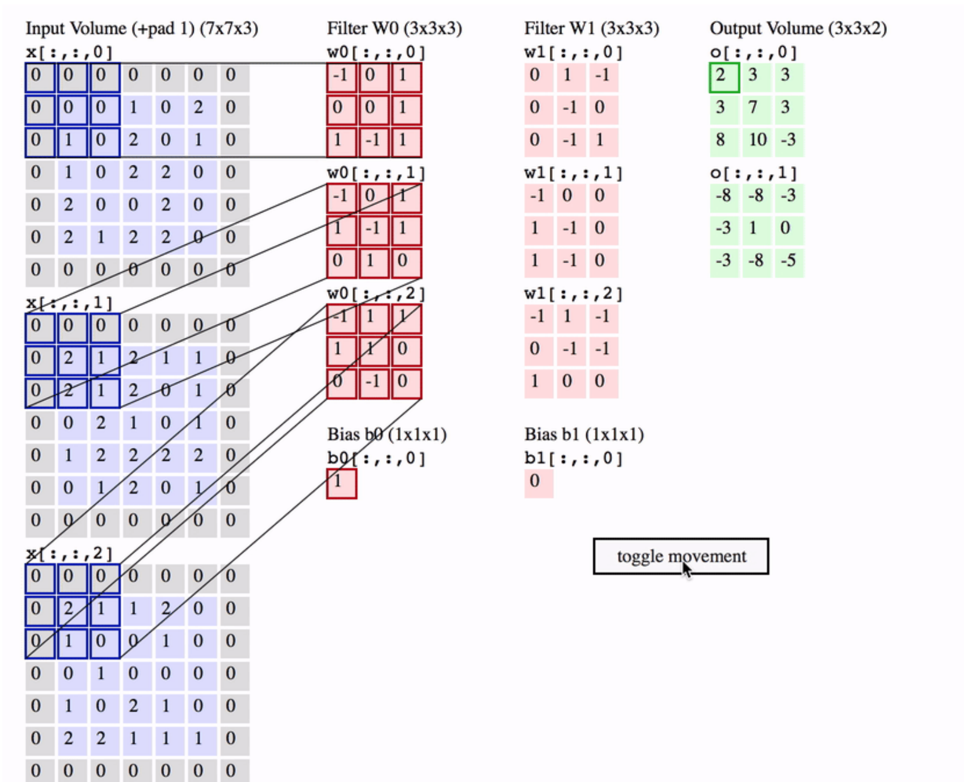
- Convolution
  - visualization



Gyeongsang National University 경상국립대학교

Convolutional layer

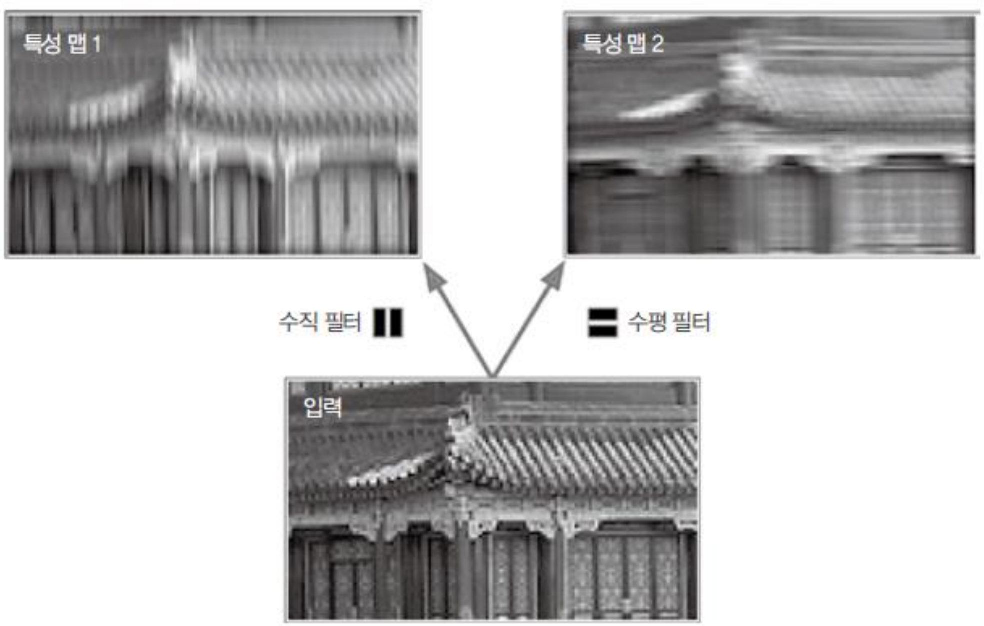
- Visualization of convolution kernel (filter)



http://naoki.io/dlt/cnn.html

Convolutional layer

- Convolution kernel (filter)
  - a set of weights based on convolution calculation



Convolutional layer

- Effects of convolution in image

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

http://aishack.in/tutorials/image-convolution-examples/

Convolutional layer

- Effects of convolution in image

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal lines

-1	2	-1
-1	2	-1
-1	2	-1

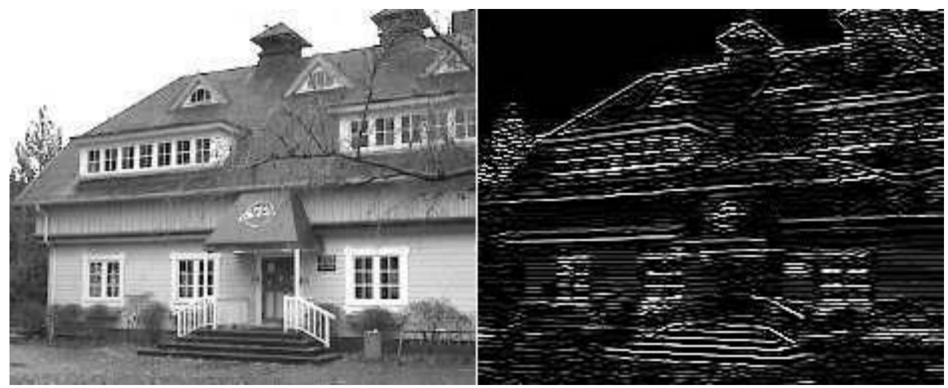
Vertical lines

-1	-1	2
-1	2	-1
2	-1	-1

45 degree lines

2	-1	-1
-1	2	-1
-1	-1	2

135 degree lines

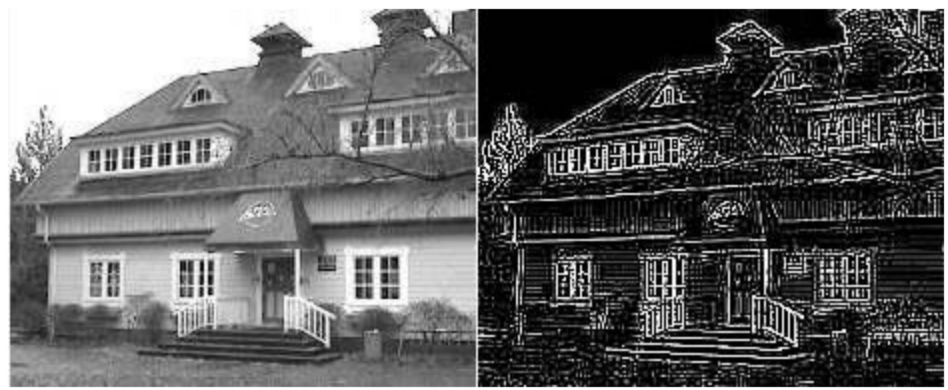


http://aishack.in/tutorials/image-convolution-examples/

Convolutional layer

- Effects of convolution in image

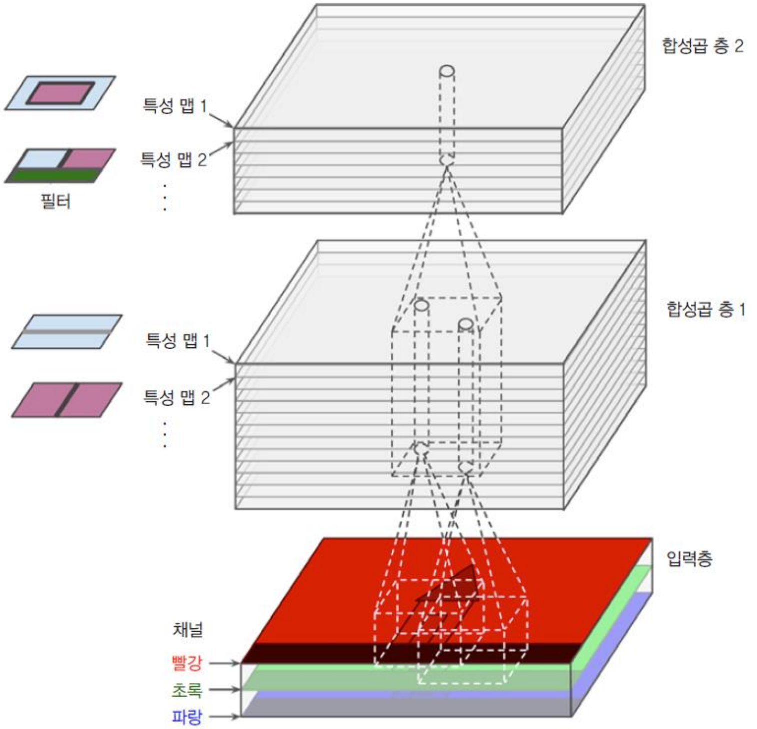
-1	-1	-1
-1	8	-1
-1	-1	-1



http://aishack.in/tutorials/image-convolution-examples/

Convolutional layer

- Accumulation of convolutional layer
  - like the vision of human

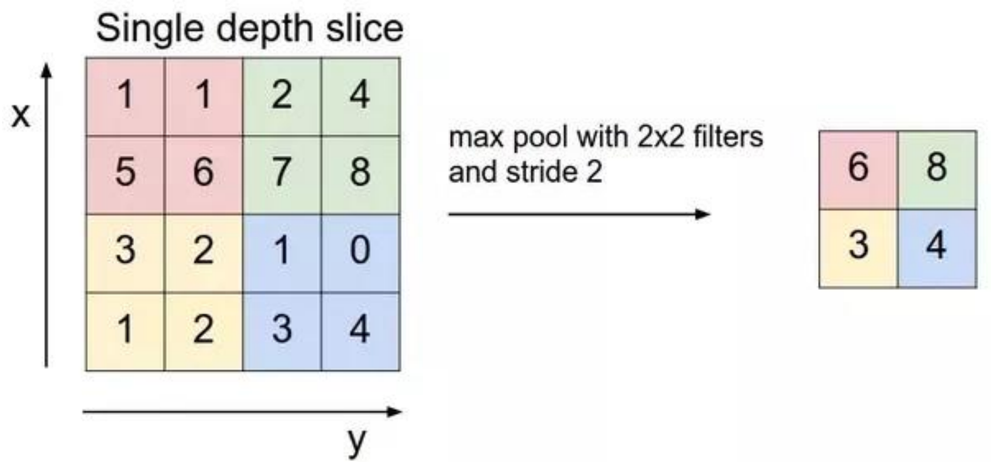


Pooling layer

- Making subsampling image
  - reduce the number of parameter
  - for time and computing complexity
  - can expect the avoidance of overfitting

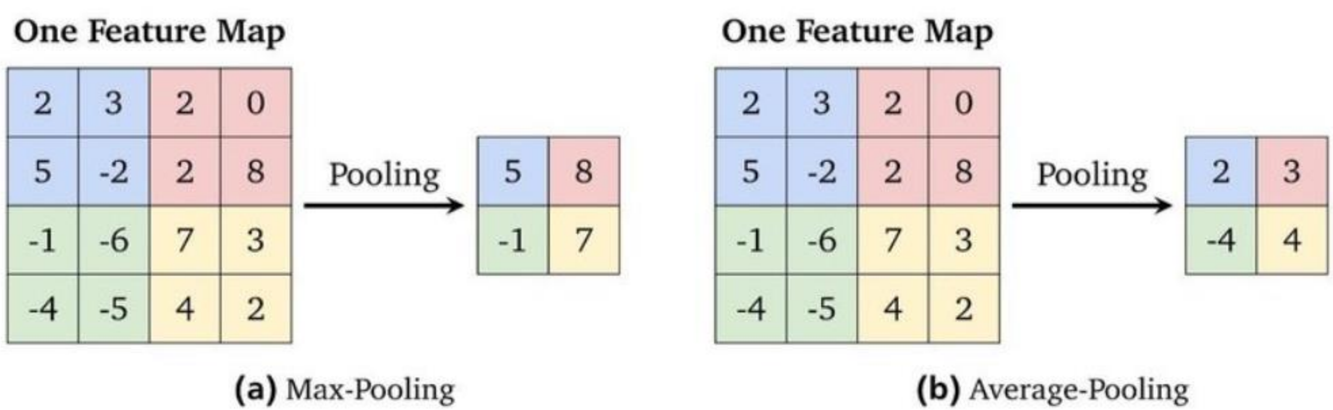
Pooling layer

- Max pooling layer
  - pooling operation that calculates the maximum value for patches of a feature map, and uses it to create a downsampled (pooled) feature map



Pooling layer

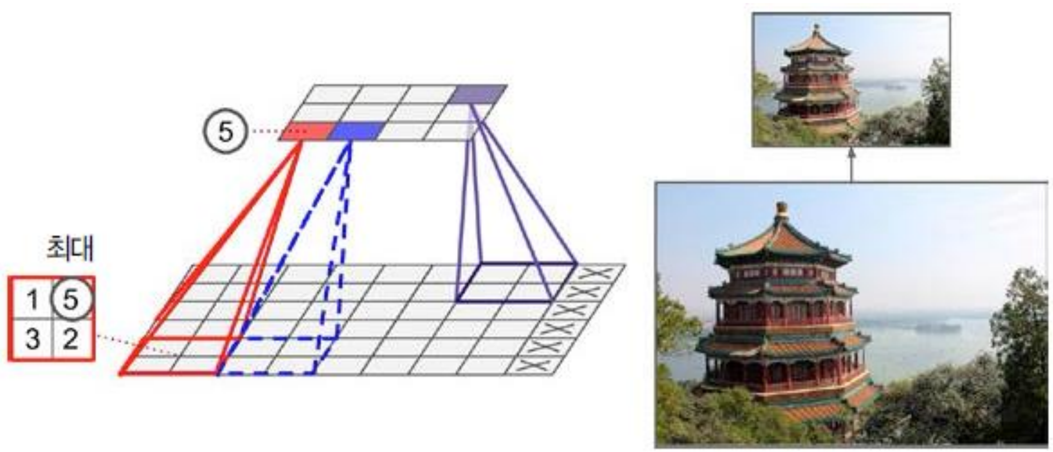
- Average pooling layer
  - pooling operation that calculates the average value for patches of a feature map



Gyeongsang National University 경상국립대학교

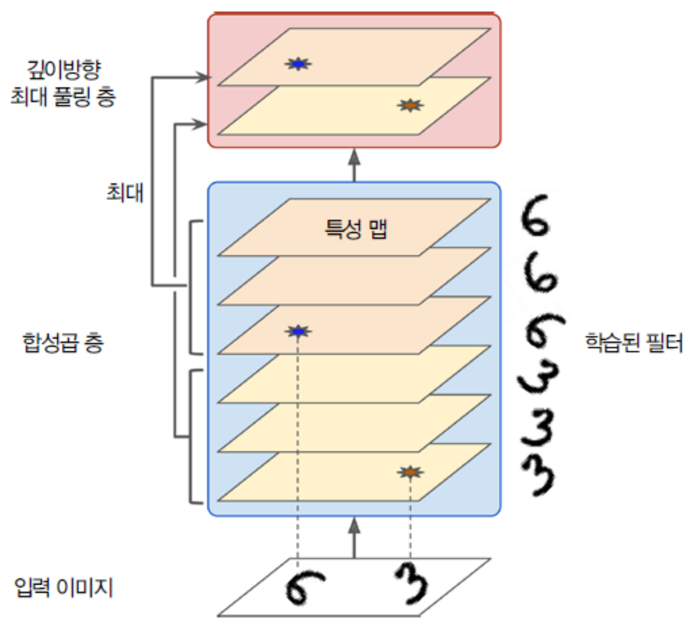
Pooling layer

- Learning invariance
  - can make invariance of input data
  - maintaining robustness with a little change



Pooling layer

- Learning invariance
  - a convolution for rotational transformation
  - after pooling, it can be generalized to the image rotation



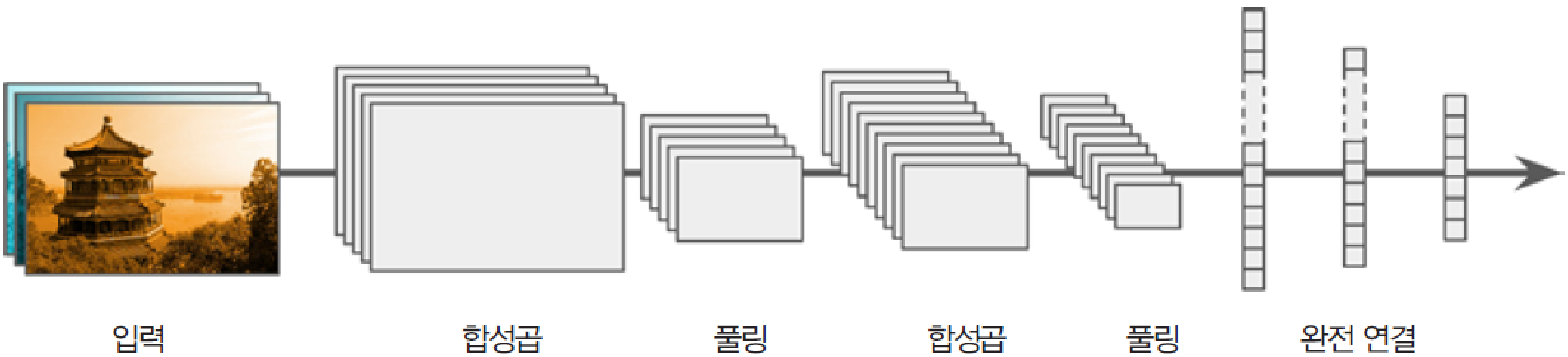
CNN structure

- General CNN structure
  - few layers of convolution each with ReLU activation
  - pooling layers
  - few layers of convolution each with ReLU activation
  - pooling layers
  - fully connected layers

GNU Gyeongsang National University 경상국립대학교

CNN structure

- General CNN structure
  - visualization



CNN structure

- Famous CNN structure
  - LeNet-5 in 1998 (for MNIST data)

층	종류	특성 맵	크기	커널 크기	스트라이드	활성화 함수
출력	완전 연결	–	10	–	–	RBF
F6	완전 연결	–	84	–	–	tanh
C5	합성곱	120	1×1	5×5	1	tanh
S4	평균 풀링	16	5×5	2×2	2	tanh
C3	합성곱	16	10×10	5×5	1	tanh
S2	평균 풀링	6	14×14	2×2	2	tanh
C1	합성곱	6	28×28	5×5	1	tanh
입력	입력	1	32×32	–	–	–

Gyeongsang National University 경상국립대학교

http://yann.lecun.com/exdb/lenet

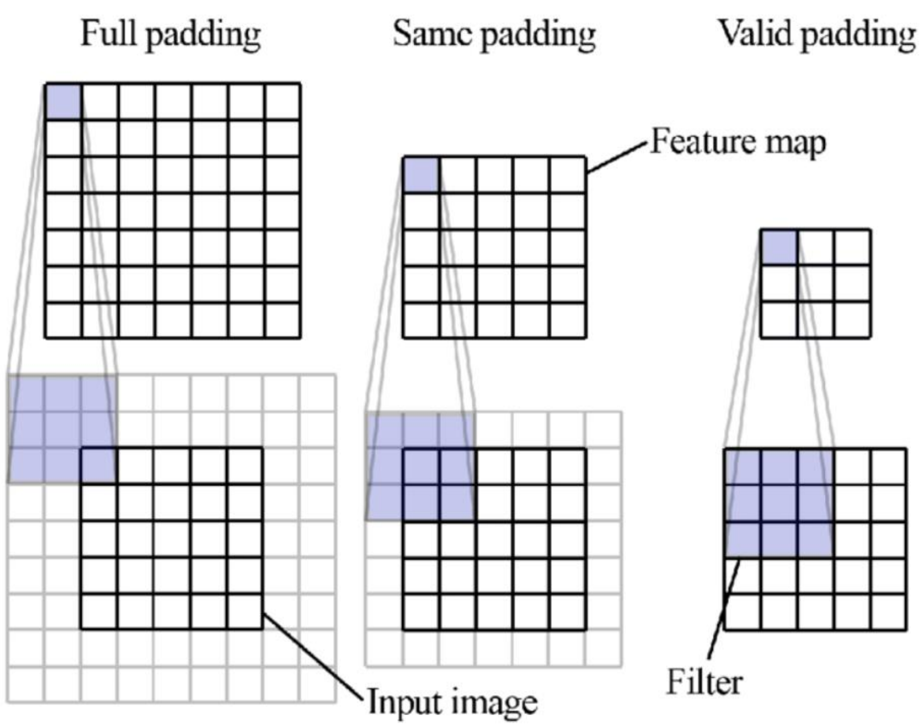
CNN structure

- Famous CNN structure
  - AlexNet in 2012 (in ILSVRC ImageNet)

층	종류	특성 맵	크기	커널 크기	스트라이드	패딩	활성화 함수
출력	완전 연결	–	1,000	–	–	–	Softmax
F10	완전 연결	–	4,096	–	–	–	ReLU
F9	완전 연결	–	4,096	–	–	–	ReLU
F8	최대 풀링	256	6×6	3×3	2	valid	–
C7	합성곱	256	13×13	3×3	1	same	ReLU
C6	합성곱	384	13×13	3×3	1	same	ReLU
C5	합성곱	384	13×13	3×3	1	same	ReLU
S4	최대 풀링	256	13×13	3×3	2	valid	–
C3	합성곱	256	27×27	5×5	1	same	ReLU
S2	최대 풀링	96	27×27	3×3	2	valid	–
C1	합성곱	96	55×55	11×11	4	valid	ReLU
입력	입력	3 (RGB)	227×227	–	–	–	–

Implementation

- Padding strategy
  - valid vs same vs full padding



Implementation

- Fashion MNIST with CNN

```
from functools import partial

DefaultConv2D = partial(keras.layers.Conv2D,
                        kernel_size=3, activation='relu', padding="SAME")

model = keras.models.Sequential([
    DefaultConv2D(filters=64, kernel_size=7, input_shape=[28, 28, 1]),
    keras.layers.MaxPooling2D(pool_size=2),
    DefaultConv2D(filters=128),
    DefaultConv2D(filters=128),
    keras.layers.MaxPooling2D(pool_size=2),
    DefaultConv2D(filters=256),
    DefaultConv2D(filters=256),
    keras.layers.MaxPooling2D(pool_size=2),
    keras.layers.Flatten(),
    keras.layers.Dense(units=128, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(units=64, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(units=10, activation='softmax'),
])
```

```
model.compile(loss="sparse_categorical_crossentropy", optimizer="nadam", metrics=["accuracy"])
history = model.fit(X_train, y_train, epochs=10, validation_data=(X_valid, y_valid))
score = model.evaluate(X_test, y_test)
X_new = X_test[:10] # 새로운 이미지처럼 사용합니다
y_pred = model.predict(X_new)
```

Feel free to question

Through e-mail & LMS

본 자료의 연습문제는 수업의 본교재인
한빛미디어, Hands on Machine Learning(2판)에서 주로 발췌함

Gyeongsang National University 경상국립대학교