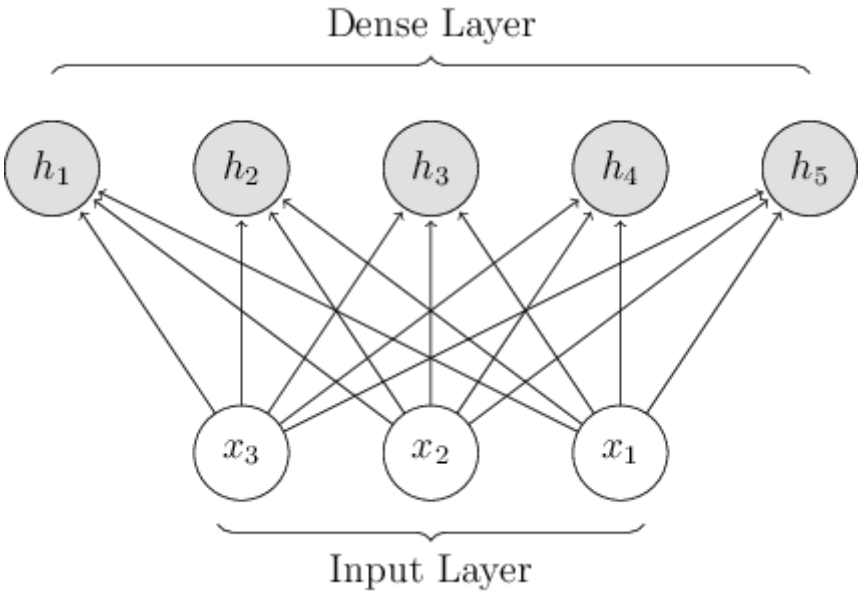
쏭노트

파이썬 딥러닝 - 07. Layer 의 종류

2020. 1. 12. 22:35 · Tech :Deep Learning

케라스에서 사용되는 레이어(Layer, 층). 핵심 데이터 구조는 모델이고, 이 모델을 구성하는 것이 Layer이다.

Dense Layer



https://slugnet.jarrodkahn.com/layers.html

다층 퍼셉트론 신경망에서 사용되는 레이어로 입력과 출력을 모두 연결해준다. 예를 들어, 입력 뉴런이 4개, 출력 뉴런이 8개라고 할때 총 연결선은 4x8=32개가 된다. 각 연결선은 가중치(weight)를 포함하고 있는데 연결강도를 의미한다. 가중치가 높을 수록 해당 입력 뉴런이 출력 뉴런에 미치는 영향이 크고, 낮을수록 미치는 영향이 작다.

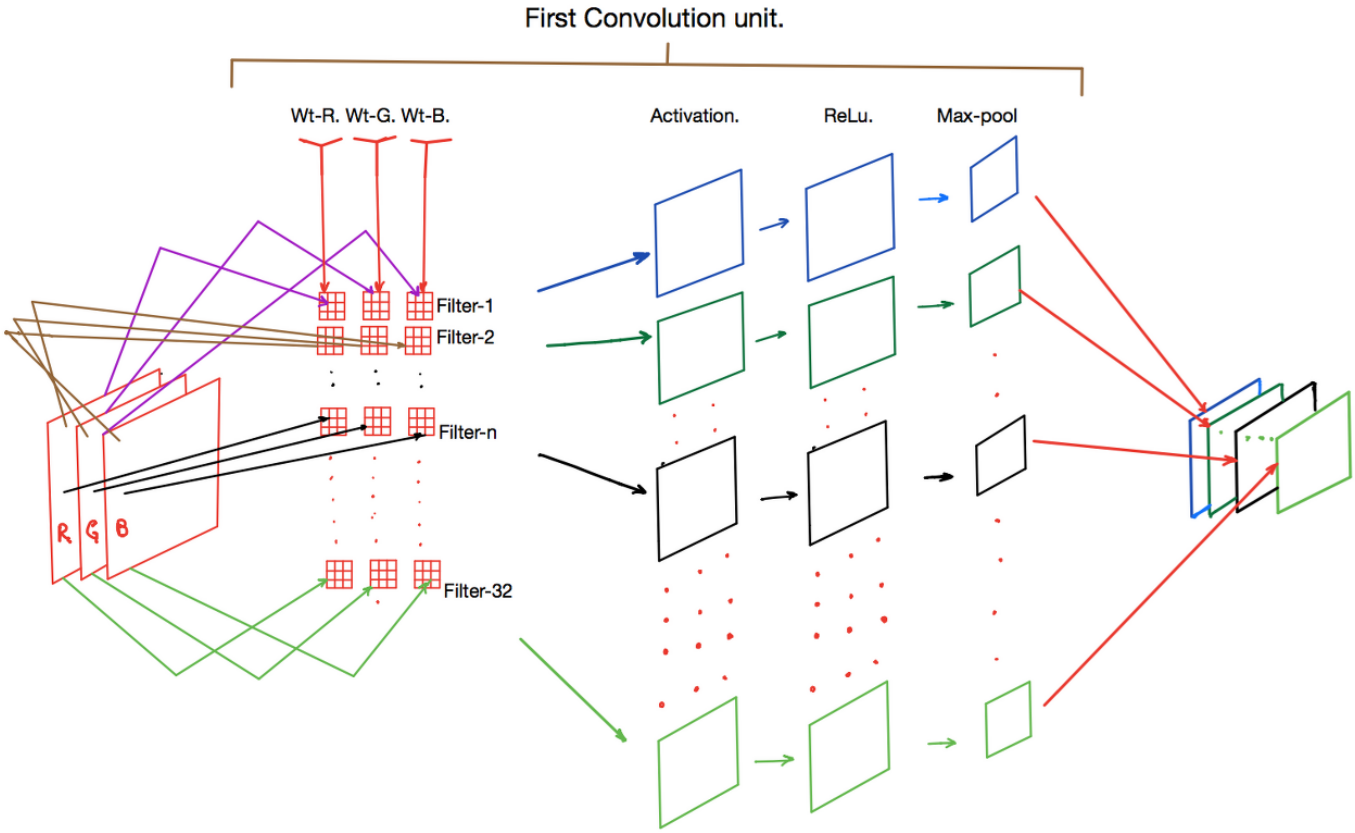
```
from tensorflow.python.keras.layers import Dense
Dense(8,input_dim=4,activation='relu')
```

- 첫번째 인자 : 출력 뉴런의 수

- input_dim : 입력 뉴런의 수

- activation : 활성화 함수를 설정. ('lenear' :Defualt 값으로 입력 뉴런과 가중치로 계산된 결과 값이 그대로 출력 값이 됨. 'relu' : rectifier 함수로 은닉층에 주로 쓰임. 'sigmoid' : 시그모이드 함수로 이진 분류 문제에서 출력 층에 주로 쓰인다. 'softmax', : 다중 클래스 분류 문제에서 출력 층에 주로 쓰인다.)

Convolution Layer



https://stats.stackexchange.com/questions/395018/why-does-each-convolution-layer-require-activation-function-and-weight-initializ

Convolution 신경망은 다층 퍼셉트론 신경망과 유사하나 이미지가 가지고 있는 특성이 고려되어 설계된 신경망으로 영상 처리에 주로 사용된다. 주로 Convolution 신경망 모델의 주요 레이어로 Convolution Layer, Max pooling Layer, Flatten Layer 가 있다. 그 중 케라스가 제공하는 Convolution Layer 종류도 여러 가지가 있으나 주로 사용되는 Conv2D Layer에 대해 보면,

```
from tensorflow.python.keras.layers.convolutional import Conv2D
Conv2D(32,(5,5), padding='valid',input_shape=
(28,28,1),activation='relu')
```

- 첫번째 인자 : Convolution filter의 수

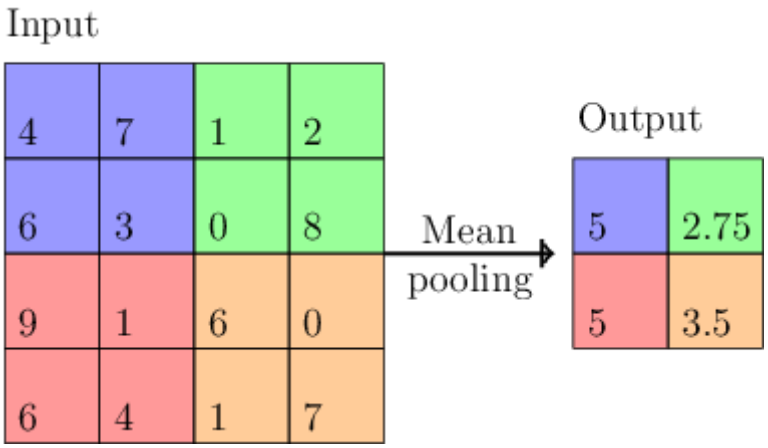
- 두번째 인자 : Convolution filter의 사이즈

- padding : 경계 처리 방법 ('valid' : 유효한 영역만 출력된다. 따라서 출력 이미지 사이즈는 입력 이미지 사이즈보다 작다. 'same' : 출력 이미지 사이즈가 입력 이미지 사이즈와 동일)

- input_shape : 샘플 수를 제외한 입력 형태를 정의. 모델에서 첫 레이어일 때만 정의하면 되는데 (행, 열, 채널 수)로 정의된다. 이 때 흑백영상일 때 channel이 1이고, 컬러(RGB) 일때는 channel이 3

- activation : Dense Layer와 동일

Max Pooling Layer



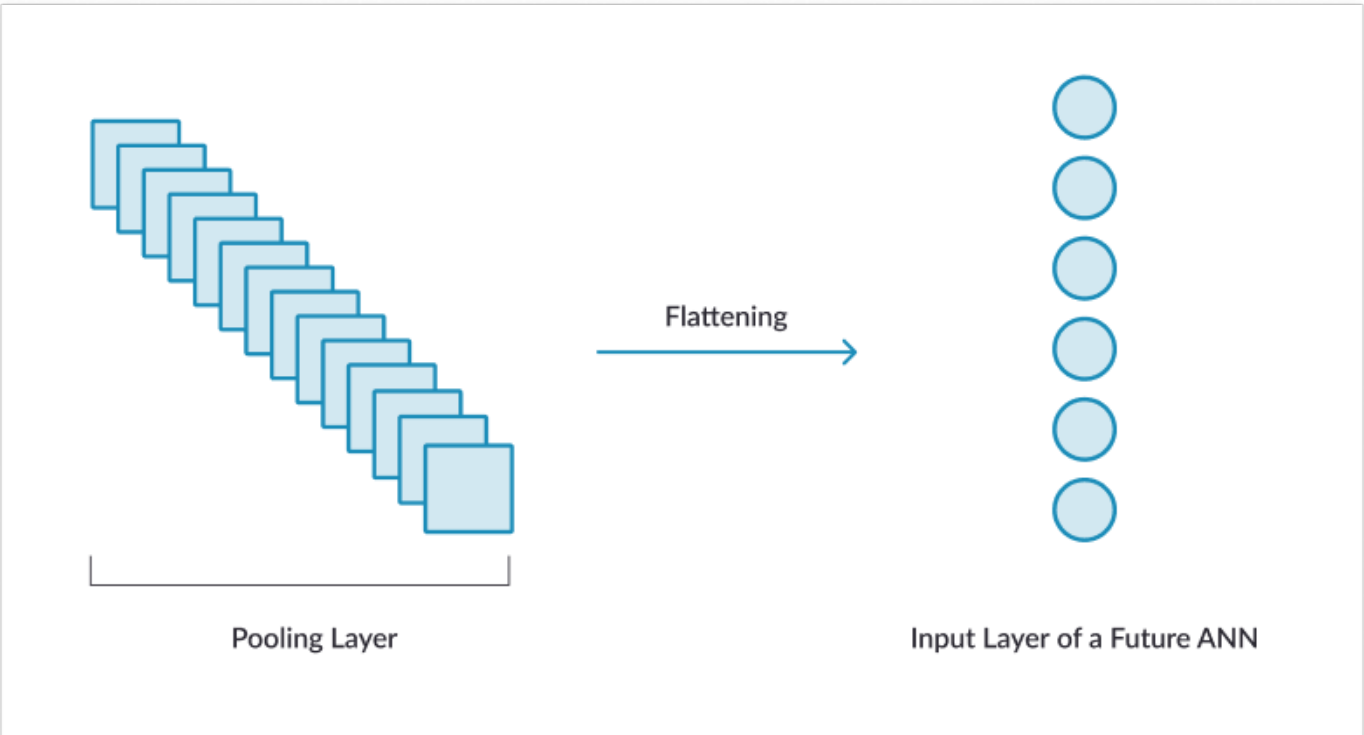
https://slugnet.jarrodkahn.com/layers.html

Convolution Layer의 출력 이미지에서 주요 값만 뽑아 크기가 작은 출력 영상을 만든다. For 지역적인 사소한 변화가 영향을 미치지 않도록

```
from tensorflow.python.keras.layers.convolutional import MaxPooling2D
MaxPooling2D(pool_size=(2,2))
```

- pool_size : 수직, 수평 축소 비율을 지정. (2,2)라면 출력 영상 크기는 입력 영상 크기의 반으로 줄어든다.

Flatten Layer



https://missinglink.ai/guides/keras/using-keras-flatten-operation-cnn-models-code-examples/

Convolution Layer나 Max Pooling Layer 를 반복적으로 거치면 주요 특징만 추출되고 추출된 주요 특징은 전결합층에 전달되어 학습된다. 전결합층에 전달하기 위해 1차원 자료로 바꿔주는데 이때 사용되는 Layer.

```
from tensorflow.python.keras.layers import Flatten
Flatten()
```

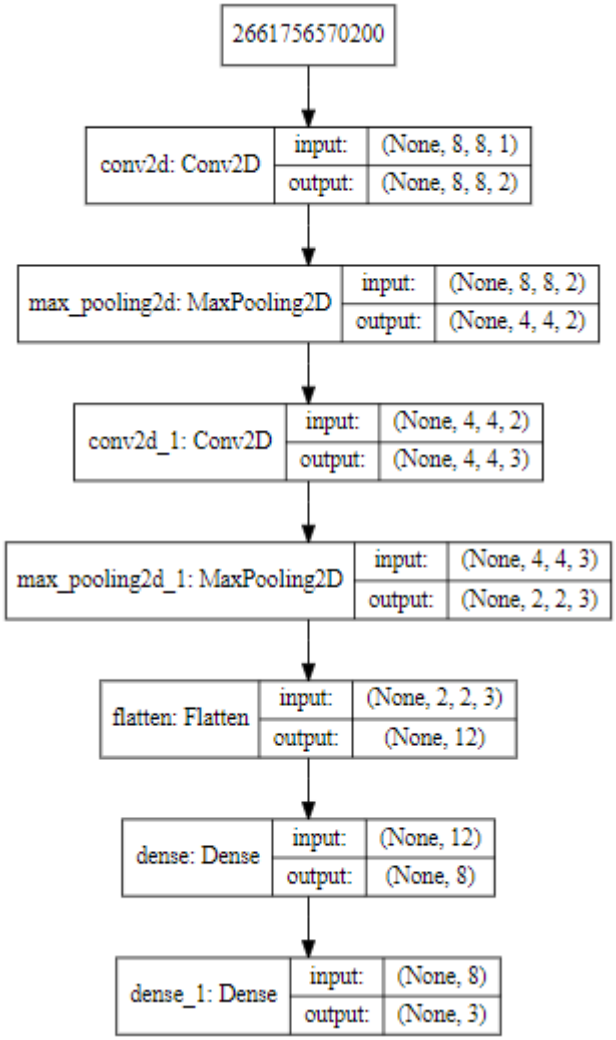
위의 Layer를 쌓아올리면 Model이 되는 셈인데,

Conv2D Layer -> MaxPooling2D Layer -> Conv2D Layer -> MaxPooling2D Layer -> Flatten Layer -> Dense Layer 를 코드로 보면

```
import numpy
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import Flatten
from tensorflow.python.keras.layers.convolutional import Conv2D
from tensorflow.python.keras.layers.convolutional import MaxPooling2D
from tensorflow.python.keras.utils import np_utils

model = Sequential()
```

```
model.add(Conv2D(2,(3,3),padding='same',activation='relu',input_shape=
(8,8,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(3,(2,2),padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(8,activation='relu'))
model.add(Dense(3,activation='softmax'))
```



위 코드를 model_to_dot을 이용하여 가시화한 결과

가시화하는 코드)

```
from IPython.display import SVG
from tensorflow.python.keras.utils.vis_utils import model_to_dot

%matplotlib inline

SVG(model_to_dot(model,show_shapes=True).create(prog='dot',format='svg')
)
```

LSTM(Long Short-Term Memory units) Layer

컨벌루션 신경망 뿐만 아니라 순환 신경망 모델이라는 것도 존재하는데 이는 순차적인 자료에서 규칙적인 패턴을 인식하거나 그 의미를 추론할 수 있다. 순차적이라는 특성 때문에 간단한 레이어로도 다양한 형태의 모델을 구성할 수 있는데 주로 사용하는 LSTM 레이어에 대해 보면

```
LSTM(3, input_dim=1, input_length=5)
```

- 첫 번째 인자 : 메모리 셀의 개수

- input_dim : 입력 속성 수

- input_length : 시퀀스 데이터의 입력 길이

- (출력) return_sequences : 시퀀시 출력 여부

Dense Layer와 비슷한데 첫 번째 인자인 메모리 셀의 개수는 기억용량 정도와 출력 형태를 결정짓는다. Dense 레이어에서의 출력 뉴런 수와 비슷하다고 보면 된다.

관련글

파이썬 딥러닝 - 06. 케라스 모델 저장하고 불러… 2020.01.12

파이썬 딥러닝 - 05. EarlyStopping 2020.01.12

파이썬 딥러닝 - 04. 케라스 학습 과정 2020.01.12

파이썬 딥러닝 - 02. 케라스 개발환경 구축 2020.01.04

댓글

ugi · 2020.01.13 11:07 신고

덕분에 많이 배우고 갑니다~!

수정/삭제|답글

댓글을 입력해주세요.

비공개 댓글 남기기

1 2 3 4 5 6 ··· 10