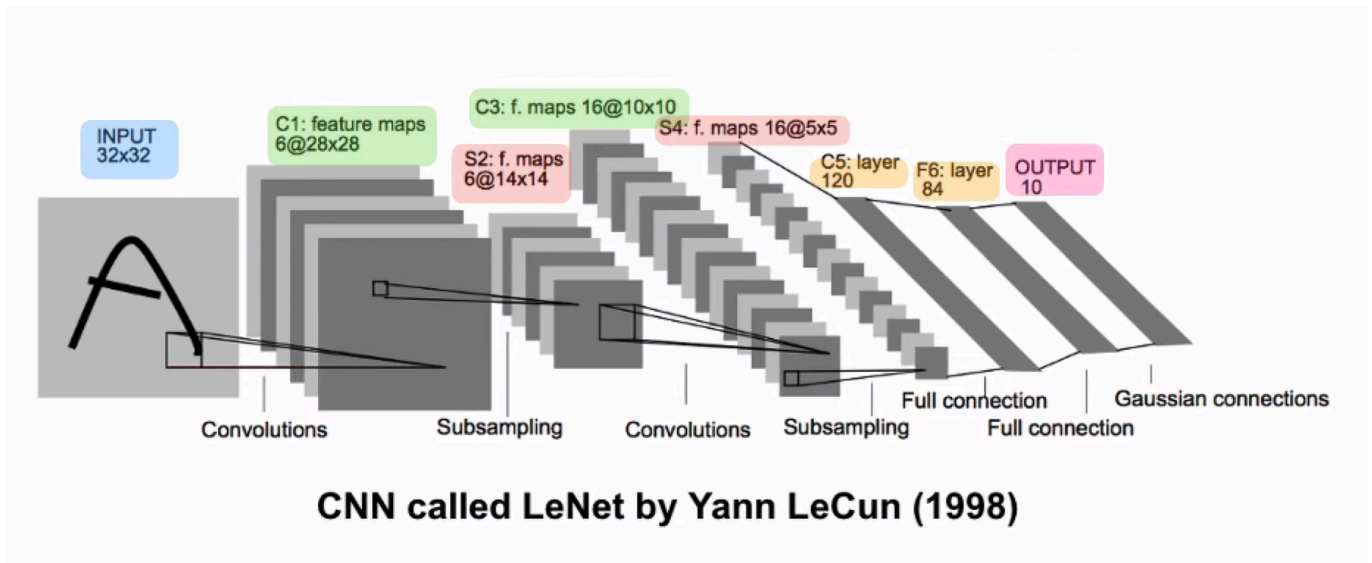


LeNet 5

CNN 모델중 손글씨 이미지를 학습한 얀 르네 교수의 르네 파이브의 그림을 보겠습니다.



수업을 시작해서 잠깐 이 모델을 구경 했었는데요. 이제 우리는 이 모델을 완성할 준비가 끝났습니다. 한번 천천히 살펴 보겠습니다. 입력으로 (32, 32) 이미지가 (좌측 첫번째 파란색네모) 사용 되었어요. 우리가 사용한 리스트는 (28, 28)이었는데요. 이 모델의 입력은 크기가 조금 더 큼니다. 첫번째 컨벌루션 레이어는 6개의 특징맵을 만들었습니다.(좌측 첫번째 초록색 네모) 각 맵은 (28, 28)사이즈로 되어 있죠. 이렇게 6개의 특징 맵을 만들려면 6개의 필터를 사용하면 되고, 사이즈가 4 줄었으니 (5, 5)사이즈로 필터를 만들면 되겠습니다. 그다음 이미지 갯수는 6개 그대로 있고 사이즈만 (14,14) 절반으로 줄었네요(첫번째 빨간색네모). 맥스 풀링을 한거죠. 두번째 컨벌루션 레이어가 나왔습니다(두번째 초록색 네모). 총 16개의 특징맵을 만들었고, 이미지의 사이즈는 (10,10) 입니다. 그다음 맥스풀링이구요(두번째 빨간색 네모). 이미지의 사이즈가 (5,5)로 줄었네요. 이후에는 플래튼을 하여 400개의 변수(5 X 5 X 16) 로 펼친 후에 120개의 노드(첫번째 노란색)와 84개의 노드(두번째 노란색)를 가진 히든 레이어를 추가 했습니다. 마지막으로 10개의 출력(빨간색)을 만들어 주었습니다.

르넷으로 구성한 코드입니다.

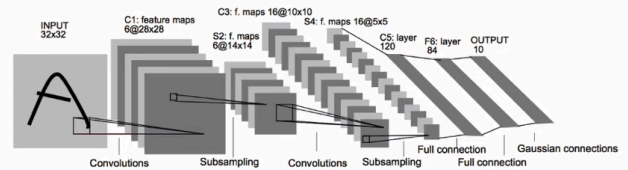
1.과거의 데이터를 준비합니다.

```
(독립, 종속), _ = tf.keras.datasets.mnist.load_data()
독립 = 독립.reshape(60000, 28, 28, 1)
종속 = pd.get_dummies(종속)
print(독립.shape, 종속.shape)
```

2. 모델의 구조를 만듭니다

```
X = tf.keras.layers.Input(shape=[28, 28, 1])
H = tf.keras.layers.Conv2D(6, kernel_size=5, padding='same', activation='swish')(X)
H = tf.keras.layers.MaxPool2D()(H)
H = tf.keras.layers.Conv2D(16, kernel_size=5, activation='swish')(H)
H = tf.keras.layers.MaxPool2D()(H)

H = tf.keras.layers.Flatten()(H)
H = tf.keras.layers.Dense(120, activation='swish')(H)
H = tf.keras.layers.Dense(84, activation='swish')(H)
Y = tf.keras.layers.Dense(10, activation='softmax')(H)
model = tf.keras.models.Model(X, Y)
model.compile(loss='categorical_crossentropy', metrics='accuracy')
```



CNN called LeNet by Yann LeCun (1998)

컨벌루션 레이어는 각각 6개의 필터와 16개(초록색)의 필터를 주었네요. 각 컨벌루션 레이어 후에는 맥스풀링으로 이미지 사이즈를 절반(빨간색)으로 줄여 주었습니다. 플래튼이후 노드 갯수는 120, 84(노란색)로 주었어. 한가지 우리가 배우지 않았던 부분이 있는데 컨벌루션 레이어의 패딩이라는 옵션(파란색)을 추가해 놓았어. 패딩을 same으로 주면 컨벌루션의 결과인 특징맵의 사이즈가 입력 이미지와 동일한 크기로 출력이 됩니다. LeNet 5는 32 X 32를 입력으로 하여 28 X 28 특징맵을 출력했는데요. 우리는 입력이 28 X 28 이니 처음 컨벌루션 에서 입력한 똑같은 사이즈의 특징맵을 출력하도록 조정한 거예요.

완성된 모델의 써머리를 확인해 봅시다.

CNN called LeNet by

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d (Conv2D)	(None, 28, 28, 6)	156
max_pooling2d (MaxPooling2D)	(None, 14, 14, 6)	0
conv2d_1 (Conv2D)	(None, 10, 10, 16)	2416
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 16)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 120)	48120
dense_1 (Dense)	(None, 84)	10164
dense_2 (Dense)	(None, 10)	850
=====		
Total params: 61,706		
Trainable params: 61,706		
Non-trainable params: 0		

컨벌루션 output은 동일하고(초록색), 풀링 레이어 결과의 모습도 동일함(빨간색)을 확인할 수 있습니다. 두번째 컨벌루션 레이어(두번째 초록색)와 두번째 맥스 풀링 레이어(두번째 빨간색)도 같네요. 플래튼 이후에 히든 레이어(파란색)도 모두 동일합니다. 자 어떨까요 CNN 구성하기가 이제 어렵지 않죠. 이제 르네5를 만들러 가보겠습니다.

비행기



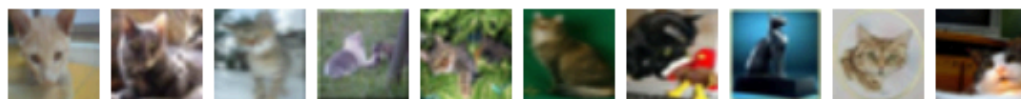
자동차



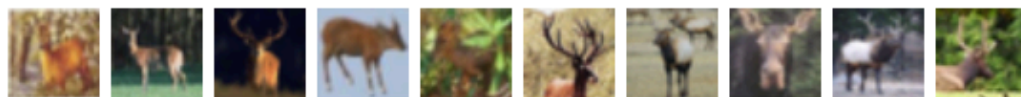
새



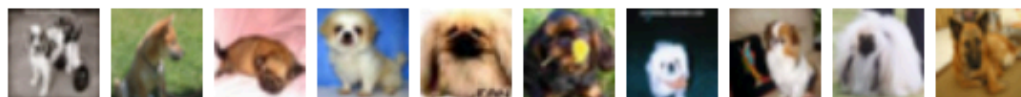
고양이



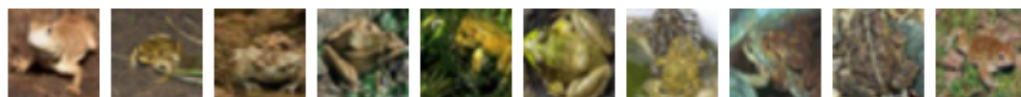
사슴



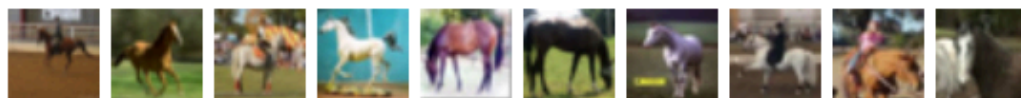
개



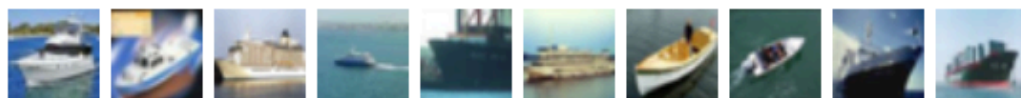
개구리



말



배



트럭



싸이파 이미지도 학습을 하겠습니다.

```

1  # -*- coding: utf-8 -*-
2  import tensorflow as tf
3  import pandas as pd
4
5  (독립, 종속), _ = tf.keras.datasets.mnist.load_data()
6  독립 = 독립.reshape(60000, 28, 28, 1)
7  종속 = pd.get_dummies(종속)
8  print(독립.shape, 종속.shape)
9
10 X = tf.keras.layers.Input(shape=[28, 28, 1])
11
12 H = tf.keras.layers.Conv2D(6, kernel_size=5, padding='same', activation='swish')(X)
13 H = tf.keras.layers.MaxPool2D()(H)
14
15 H = tf.keras.layers.Conv2D(16, kernel_size=5, activation='swish')(H)
16 H = tf.keras.layers.MaxPool2D()(H)
17
18 H = tf.keras.layers.Flatten()(H)
19 H = tf.keras.layers.Dense(120, activation='swish')(H)
20 H = tf.keras.layers.Dense(84, activation='swish')(H)
21 Y = tf.keras.layers.Dense(10, activation='softmax')(H)
22
23 model = tf.keras.models.Model(X, Y)
24 model.compile(loss='categorical_crossentropy', metrics=['accuracy'])
25
26 model.fit(독립, 종속, epochs=10)
27
28 model.summary()
29
30 (독립, 종속), _ = tf.keras.datasets.cifar10.load_data()
31 print(독립.shape, 종속.shape)
32
33 종속 = pd.get_dummies(종속.reshape(50000))

```

```
34 print(독립.shape, 종속.shape)
35
36 X = tf.keras.layers.Input(shape=[32, 32, 3])
37
38 H = tf.keras.layers.Conv2D(6, kernel_size=5, activation='swish')(X)
39 H = tf.keras.layers.MaxPool2D()(H)
40
41 H = tf.keras.layers.Conv2D(16, kernel_size=5, activation='swish')(H)
42 H = tf.keras.layers.MaxPool2D()(H)
43
44 H = tf.keras.layers.Flatten()(H)
45 H = tf.keras.layers.Dense(120, activation='swish')(H)
46 H = tf.keras.layers.Dense(84, activation='swish')(H)
47 Y = tf.keras.layers.Dense(10, activation='softmax')(H)
48
49 model = tf.keras.models.Model(X, Y)
50 model.compile(loss='categorical_crossentropy', metrics=['accuracy'])
51
52 model.fit(독립, 종속, epochs=10)
53
54 model.summary()
```
