개념정리 #8

딥러닝 파이토치 교과서 6장

6. 합성곱 신경망 II

6.1. 이미지 분류를 위한 신경망

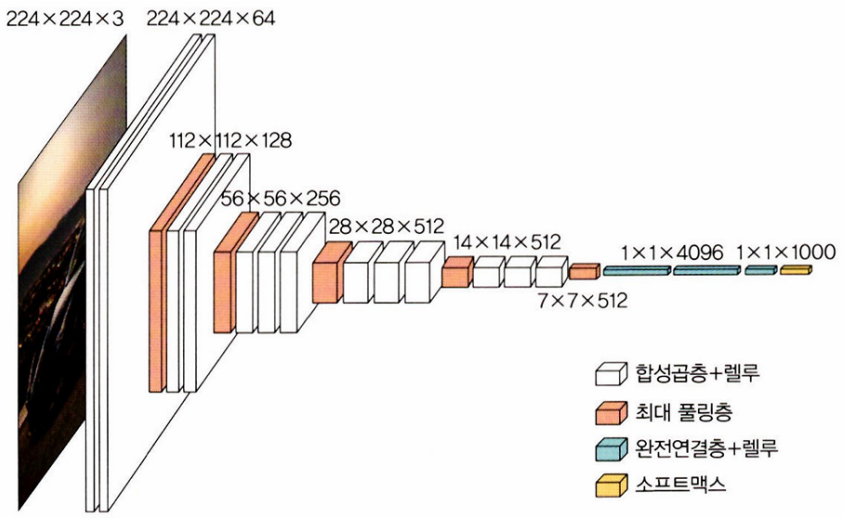
입력 데이터가 이미지인 분류(classification) : 특정 대상이 영상 내에 존재하는지 여부를 판단하는 것

→ Image classification

VGGNet

- "*Very deep convolutional networks for large-scale image recognition*" 논문
- 합성곱층의 파라미터 수를 줄이고 훈련 시간을 개선하려고 탄생함 — 네트워크를 깊게 만드는 것이 성능에 어떤 영향을 미치는지 확인하고자 함
- VGG 연구 팀은 깊이의 영향만 최대한 확인하고자 합성곱층에서 사용하는 커널의 크기를 가장 작은 3x3으로 고정함

- VGG16 (네트워크 계층이 16개)
    - 파라미터가 총 1억 3300만 개 있음
    - 모든 합성곱 커널 크기가 3x3
    - max pooling 커널 크기는 2x2, 스트라이드는 2
    - 마지막 16번째 계층 제외 전부 ReLU 활성화 함수 적용



VGG16 구조

	Feature map	Size	Kernel size	Stride	Activation funct.
Input image	1	224 × 224			
Conv1	64	224 × 224	3 × 3	1	ReLU
Conv2	64	224 × 224	3 × 3	1	ReLU
MaxPool1	64	112 × 112	2 × 2	2	
Conv3	128	112 × 112	3 × 3	1	ReLU
Conv4	128	112 × 112	3 × 3	1	ReLU
MaxPool2	128	56 × 56	2 × 2	2	
Conv5	256	56 × 56	3 × 3	1	ReLU
Conv6	256	56 × 56	3 × 3	1	ReLU
Conv7	256	56 × 56	3 × 3	1	ReLU
Conv8	256	56 × 56	3 × 3	1	ReLU
MaxPool3	256	28 × 28	2 × 2	2	
Conv9	512	28 × 28	3 × 3	1	ReLU
Conv10	512	28 × 28	3 × 3	1	ReLU
Conv11	512	28 × 28	3 × 3	1	ReLU
Conv12	512	28 × 28	3 × 3	1	ReLU
MaxPool4	512	14 × 14	2 × 2	2	
Conv13	512	14 × 14	3 × 3	1	ReLU
Conv14	512	14 × 14	3 × 3	1	ReLU
Conv15	512	14 × 14	3 × 3	1	ReLU
Conv16	512	14 × 14	3 × 3	1	ReLU
MaxPool5	512	7 × 7	2 × 2	2	
FC1		4096			ReLU
FC2		4096			ReLU
FC3		1000			Softmax

Copy

1. 단순한 객체 복사

```
original = [1, 2, 3]
copy_o = original
```

`copy_o` 의 값을 바꾸면 `original`의 값도 변경됨

2. 얕은 복사 (shallow copy)

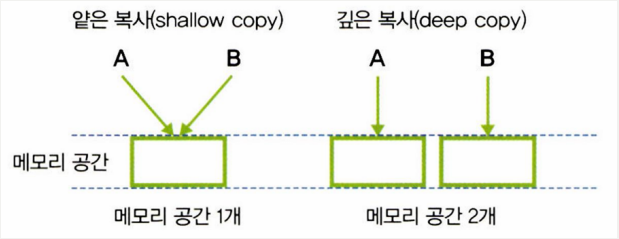
```
import copy

original = [1, 2, 3]
copy_o = copy.copy(original)
# copy_o = original[:]
```

- 요소 값 변경 → 원본에는 반영 X
- 요소 추가 → 원본에도 반영 O
- 리스트 자체는 별도의 객체이지만, 리스트 내의 원소들이 동일한 참조
- 리스트는 **가변 객체**라서 원본 리스트의 구조를 변경하는 연산(요소 추가, 제거 등)은 **참조된 모든 곳에 영향**을 미

3. 깊은 복사 (deep copy)

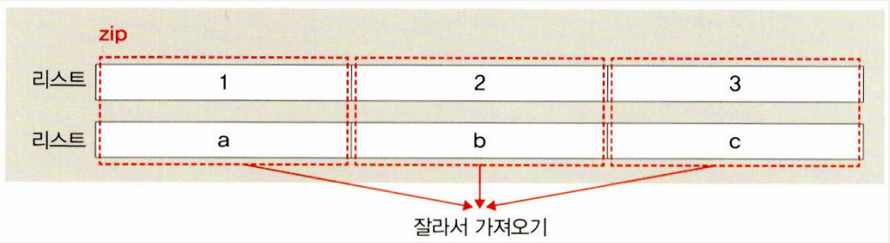
```
import copy

original = [1, 2, 3]
copy_o = copy.deepcopy(original)
```

- 요소 값 변경, 요소 추가 모두 원본에 반영 X
- 내부에 객체들까지 모두 새롭게 copy



`zip()`

여러 개의 리스트(혹은 튜플)을 합쳐서 새로운 튜플 타입으로 반환



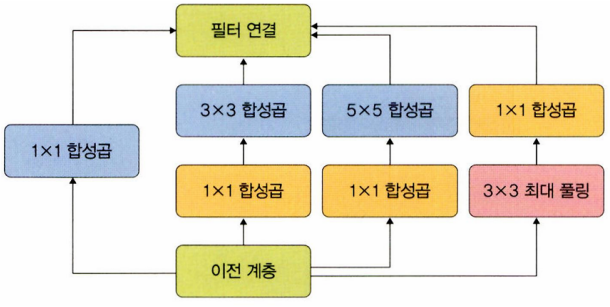

```
a = [1, 2, 3]
b = ['a', 'b', 'c']

for x, y in zip(a,b):
    print(x, y)

# 1 a
# 2 b
# 3 c
```

메서드에 _ 표시

기존의 메모리 공간에 있는 값을 새로운 값으로 대체하겠다는 의미

GoogLeNet

- 주어진 하드웨어 자원을 최대한 효율적으로 이용하면서, 학습 능력은 극대화할 수 있는 깊고 넓은 신경망
- Inception module : 특징을 효율적으로 추출하기 위해 1 x 1, 3 × 3, 5 × 5 합성곱 연산을 각각 수행함
  - 3 × 3 max pooling은 입력과 출력의 높이, 너비가 같아야 하므로 드물게 패딩을 추가해야 함
  - → Sparse connectivity : 빽빽하게 연결된 신경망 대신 correlation이 높은 노드끼리만 연결하는 방법 (연산량이 적어지고 overfitting 방지)



인셉션 모듈

- 인셉션 모듈의 4가지 연산
  - 1 × 1 conv
  - 1 × 1 conv + 3 × 3 conv
  - 1 × 1 conv + 5 × 5 conv
  - 3 × 3 max pooling + 1 × 1 conv

심층 신경망의 아키텍처에서 layer가 넓고 깊으면 인식률은 좋아지지만, overfitting이나 vanishing gradient 문제를 비롯한 학습 시간 지연과 연산 속도 등의 문제가 있음

특히 CNN에서 이러한 문제들이 자주 나타나는데, GoogLeNet으로 이러한 문제를 해결할 수 있음