

<인공지능 3차 과제 2>

<과제 3.2: VGGNet 설계>

1. 개요

VGGNet의 개발자들은 모델의 깊이가 성능에 얼마나 영향을 끼칠지에 집중하여 연구하였다고 논문에서 밝혔다. 깊은 네트워크를 가지고 있지만, GoogLeNet과 비교하면, 구조가 매우 간단하다. 깊이에 따른 변화를 비교하기 위해, 3x3의 작은 필터 크기를 사용했고, 모델 깊이와 구조에 변화를 주어 실험하였다. 논문에서 언급한 것은 총 6개의 모델로 내용은 다음 표와 같다. D 구조를 VGG16, E 구조를 VGG19라고 부른다. 다음 표에서, "conv 3, 64" 는 3x3 convolution 연산에 출력 피쳐맵 갯수는 64개라는 뜻이다.

(<https://datascienceschool.net/view-notebook/47c57f9446224af08f13e1f3b00a774e/> 에서 발췌)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

2. 과제(30점)

3차 과제 1의 hw3-VGGNet.ipynb의 VGGNet(A번), VGGNet16(C번) 클래스를 hw3_submit.ipynb를 참고해 수정해서 위의 표의 A와 C를 구현하여라. 단 들어가는 데이터는 CIFAR-10(32*32*3 클래스 10개)라 가정하고서 구현을 해야 한다. Maxpooling을 통해서 데이터의 width와 height가 반씩 줄어들고, 각 Conv와 FC4096 부분엔 무조건 ActivateFunction인 Relu가 들어가야 한다.

해당하는 과제는 트레이닝 시킬 필요 없이 만들어진 VGGNet11, VGGNet16클래스 안의 layers의 정보만으로 채점을 할 것이다. VGGNet의 주요한 특징은 ppt Week9-2와 인터넷을 참고하면 된다. 클래스넷에 hw3-VGGNet.ipynb 코드와 아래와 같은 출력결과와 50000*32*32*3의 데이터가 들어갔을 때, 데이터가 해당 레이어를 나오면 어떤 형태로 바뀌는지에 대한 순서도를 추가한 보고서를 제출해야한다.(컴퓨터 자필 스캔 상관없음.)[순서도 양식은 알아만 볼 수 있으면 됨]

```
In [81]: VGG11ConvLayerList = [['C', 5, 5, 3, 2, 2], ['C', 3, 3, 5, 1, 1], ['P', 2, 2, 0],
                             ['C', 10, 3, 5, 1, 1], ['P', 2, 2, 0], ['C', 3, 1, 10, 1, 1], ['P', 2, 2, 0]]
VGG11FullLayerList = [[3*3*10, 50], [50, 10]]

VGGNet11 = VGGNet(VGG11ConvLayerList, VGG11FullLayerList)

In [92]: VGG16ConvLayerList = [['C', 5, 5, 3, 2, 2], ['C', 3, 3, 5, 1, 1], ['P', 2, 2, 0],
                             ['C', 10, 3, 5, 1, 1], ['P', 2, 2, 0], ['C', 3, 1, 10, 1, 1], ['P', 2, 2, 0]]
VGG16FullLayerList = [[3*3*10, 50], [50, 10]]

VGGNet16 = VGGNet16(VGG16ConvLayerList, VGG16FullLayerList)

In [90]: print("VGG11에 대한 내용")
print("전체적인 Layer", VGGNet11.layers.keys())
print("-----")
for key, layer in VGGNet11.layers.items():

    if type(layer) == Convolution:
        print("이 레이어는 ", key, "이고, #의 Shape는 ", layer.W.shape, "이다")
    elif type(layer) == LinearLayer:
        print("이 레이어는 ", key, "이고, #의 Shape는 ", layer.W.shape, "이다")

VGG11에 대한 내용
전체적인 Layer odict_keys(['C1', 'R1', 'C2', 'R2', 'P3', 'C4', 'R4', 'P5', 'C6', 'R6', 'P7', 'F8', 'R8', 'F9', 'R9', 'F10'])
-----
이 레이어는 C1 이고, #의 Shape는 (5, 3, 5, 5) 이다
이 레이어는 C2 이고, #의 Shape는 (3, 5, 3, 3) 이다
이 레이어는 C4 이고, #의 Shape는 (10, 5, 3, 3) 이다
이 레이어는 C6 이고, #의 Shape는 (3, 10, 1, 1) 이다
이 레이어는 F8 이고, #의 Shape는 (90, 50) 이다
이 레이어는 F9 이고, #의 Shape는 (50, 10) 이다
이 레이어는 F10 이고, #의 Shape는 (10, 10) 이다

In [93]: print("VGG16에 대한 내용")
print("전체적인 Layer", VGGNet16.layers.keys())
print("-----")
for key, layer in VGGNet16.layers.items():

    if type(layer) == Convolution:
        print("이 레이어는 ", key, "이고, #의 Shape는 ", layer.W.shape, "이다")
    elif type(layer) == LinearLayer:
        print("이 레이어는 ", key, "이고, #의 Shape는 ", layer.W.shape, "이다")

VGG16에 대한 내용
전체적인 Layer odict_keys(['C1', 'R1', 'C2', 'R2', 'P3', 'C4', 'R4', 'P5', 'C6', 'R6', 'P7', 'F8', 'R8', 'F9', 'R9', 'F10'])
-----
이 레이어는 C1 이고, #의 Shape는 (5, 3, 5, 5) 이다
이 레이어는 C2 이고, #의 Shape는 (3, 5, 3, 3) 이다
이 레이어는 C4 이고, #의 Shape는 (10, 5, 3, 3) 이다
이 레이어는 C6 이고, #의 Shape는 (3, 10, 1, 1) 이다
이 레이어는 F8 이고, #의 Shape는 (90, 50) 이다
이 레이어는 F9 이고, #의 Shape는 (50, 10) 이다
이 레이어는 F10 이고, #의 Shape는 (10, 10) 이다
```

3. 제출물

1. 클래스넷에 6월 17일 월 01:00AM 까지 hw3-VGGNet.ipynb 와 해당코드에 대한 print문, 해당 코드에 CIFAR10 training_data(50000, 3, 32, 32)가 들어갔을 때 어떤 식으로 shape가 바뀌는 지에 대한 순서도 그림을 간단히 그려서 제출.
2. 자신의 인터뷰 시간에 해당코드에 대한 print문과 순서도를 하드카피해 들고 와야한다. 안 들고 올 시 클래스넷 제출여부와 상관없이 0점 처리한다.