



## 6-1장 정리

### 6-1장

#### 6.1.1 LeNet-5

합성곱과 다운 샘플링(풀링)을 반복적으로 거치면서 마지막에 FC layer에서 분류를 수행한다.

c1: 5x5 합성곱

s2: 다운 샘플링하여 14x14로 줄인다.

c3: 5x5 합성곱

s4: 다운 샘플링하여 5x5로 줄인다.

c5: 5x5 합성곱

f6: 완전연결층으로 84개에 연결

출력: 분류

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

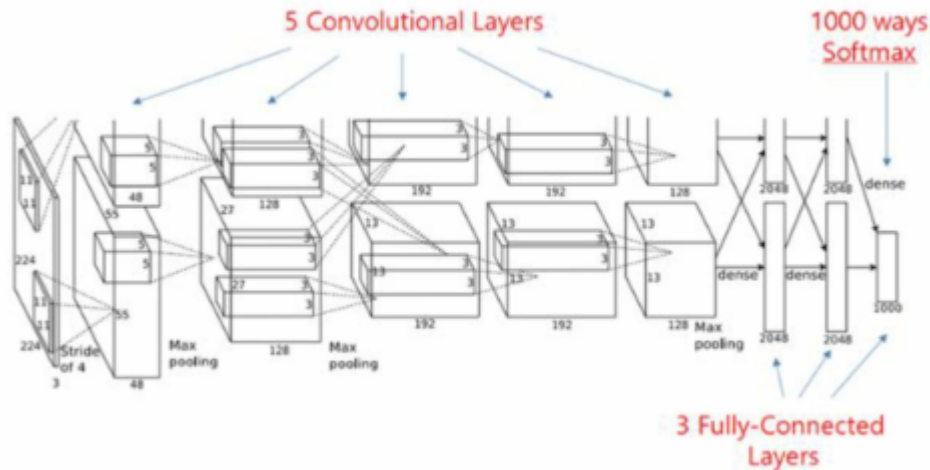
activation 함수는 relu등으로 해도 됨

### 6.1.2 AlexNet

imagenet 화상 인식 대회에서 우승한 cnn구조

합성곱 층 5개+FC layer 3개+마지막은 softmax

GPU 두 개를 기반으로 한 병렬 구조이다. (활성화 함수로는 렐루를 이용)



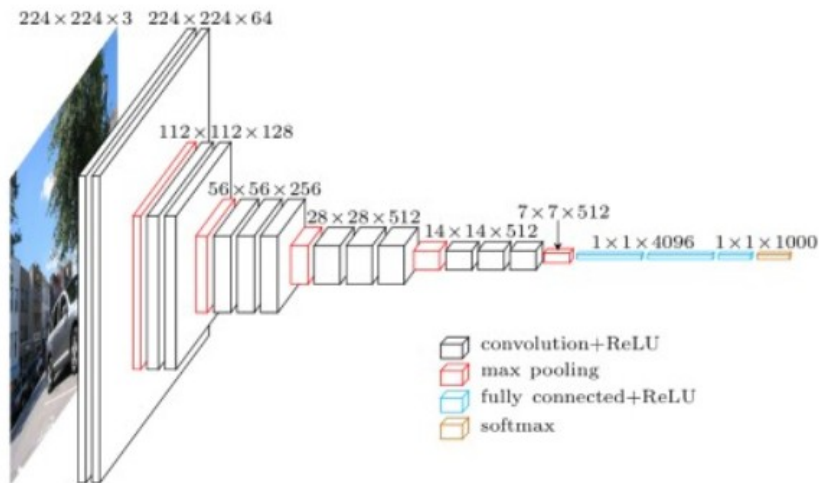
GPU-1에서는 주로 컬러와 상관없는 정보를 추출, GPU-2에서는 주로 컬러와 관련된 정보를 추출하기 위한 커널이 학습된다.

### 6.1.3 VGGNet

합성곱층의 파라미터 수를 줄이고 훈련 시간을 개선하고자 탄생함.

- 네트워크를 깊게 만드는 것이 성능에 어떤 영향을 미치는지 확인하고자 만들어짐
- 깊이의 영향만을 확인하고자 필터 사이즈를 가장 작은 3x3으로 고정함.

네트워크 계층의 총 개수에 따라 VGG16, VGG19등으로 나타난다. 활성화 함수로는 ReLU를 이용함.



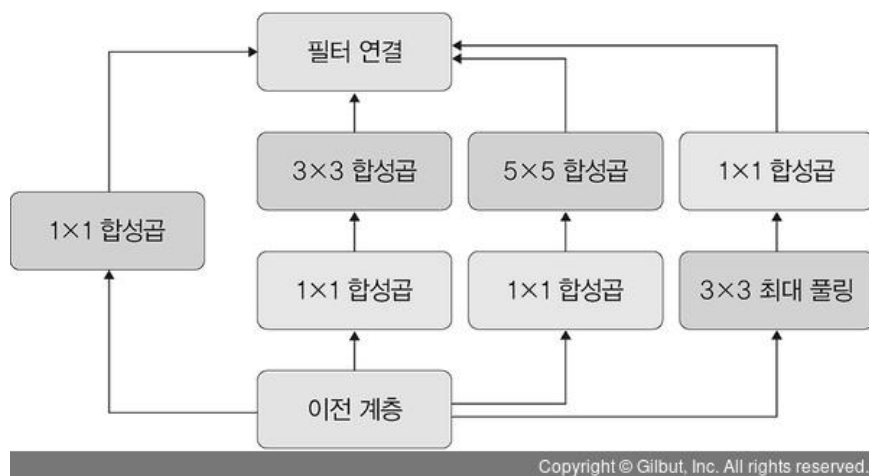
VGG 11의 구조

#### 6.1.4 GoogLeNet

깊고 넓은 신경망을 위해 인셉션 모듈을 추가함, 뾰뾰하게 연결된 신경망 대신 관련성이 높은 노드끼리 연결하는 방식(희소 연결-sparse connectivity)

\*인셉션 모듈: 특징을 효율적으로 추출하기 위해  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ 의 합성곱 연산을 각각 수행함

- $1 \times 1$  합성곱
- $1 \times 1$  합성곱 +  $3 \times 3$  합성곱
- $1 \times 1$  합성곱 +  $5 \times 5$  합성곱
- $3 \times 3$  max pooling+ $1 \times 1$  합성곱 ( $3 \times 3$  maxpooling은 입력과 출력의 높이와 너비가 같아야 하므로 패딩을 추가한다.)



Copyright © Gilbut, Inc. All rights reserved.

### 6.1.5 ResNet

깊어진 신경망을 효과적으로 학습하기 위해 레지듀얼(residual)개념을 고안함.

네트워크의 깊이가 깊을수록 무조건 성능이 좋은 것이 아니다.

residual block: 기울기가 잘 전파될 수 있도록 shortcut을 만들어 준다. 층이 152개로 구성 되어 Googlenet(22층)에 비해 기울기 소멸 문제가 발생할 수 있기 때문이다.

블록: 합성곱층을 하나의 블록으로 묶은 것.

bottleneck block(병목 블록): 계층의 깊이가 깊어질수록 파라미터가 무제한으로 커지는 것을 해결하고자 도입.

병목 블록을 사용하면 깊이가 깊어졌음에도 파라미터 수는 감소한다.

ex) ResNet50에서는 3x3합성곱 앞뒤로 1x1합성곱층이 붙어 있다.

