

Going Deeper with Convolutions

GoogLeNet

발표자 : 허진규

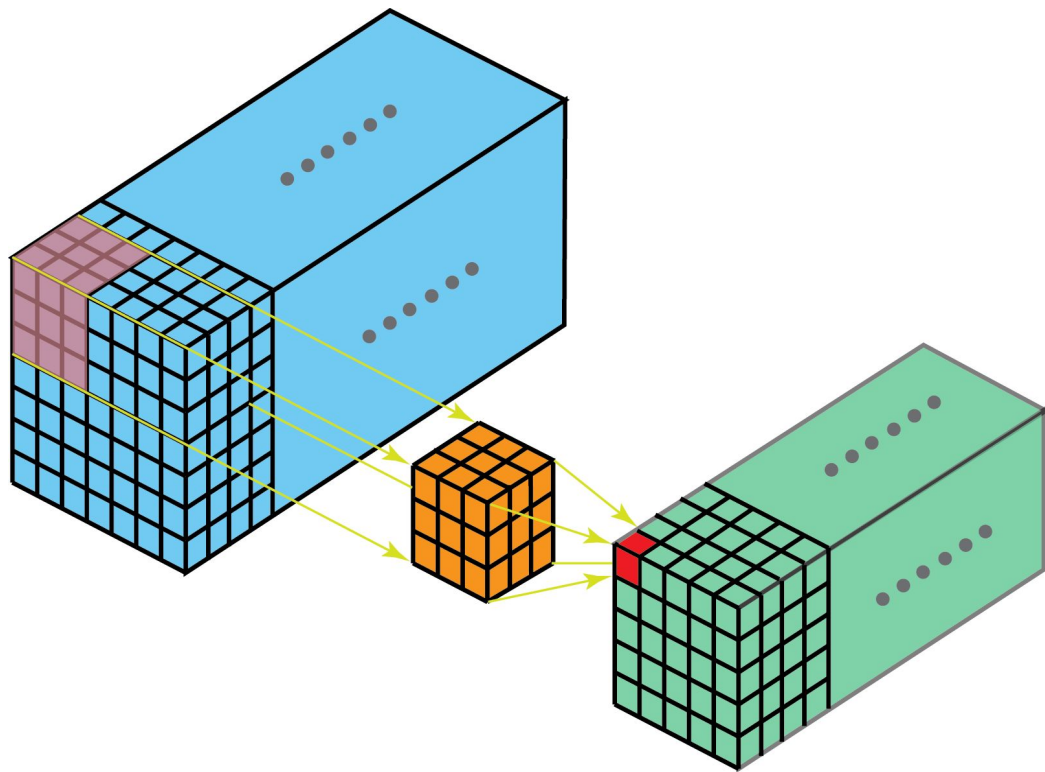
abstract

1. deep convolution neural network 구조
2. classification & detection
3. 네트워크 내에서 컴퓨터 자원 효율성 상승
4. resource budget 유지하며 depth, width 증가
5. 22 layers

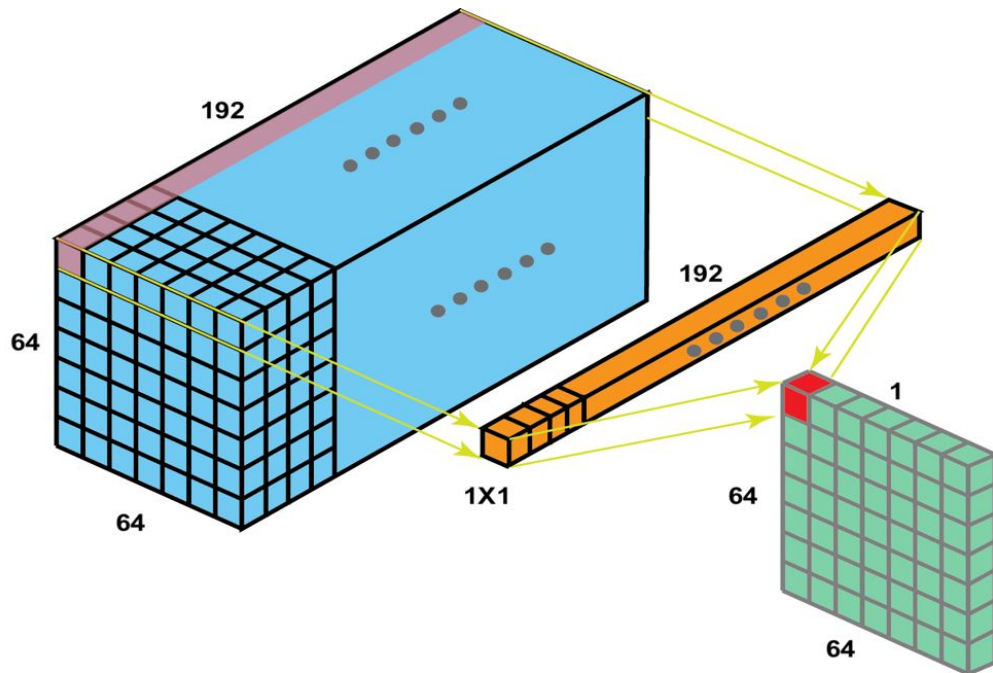
설명에 앞서 미리 이해하고 가자!

- 2014 ILSVRC Competition에서 우승한 모델
- **layer**가 깊어진다는 것은 **param**의 수가 증가 한다는 뜻이고, **overfitting** 발생 가능성과 연산량이 증가한다는 의미로 해석 될 수 있다.
- 1x1 conv 은 연산량을 줄이는 데에 사용된다.
- Max Pooling layer 는 계속 커지는 차원을 줄여주는 역할을 하는 layer이다.

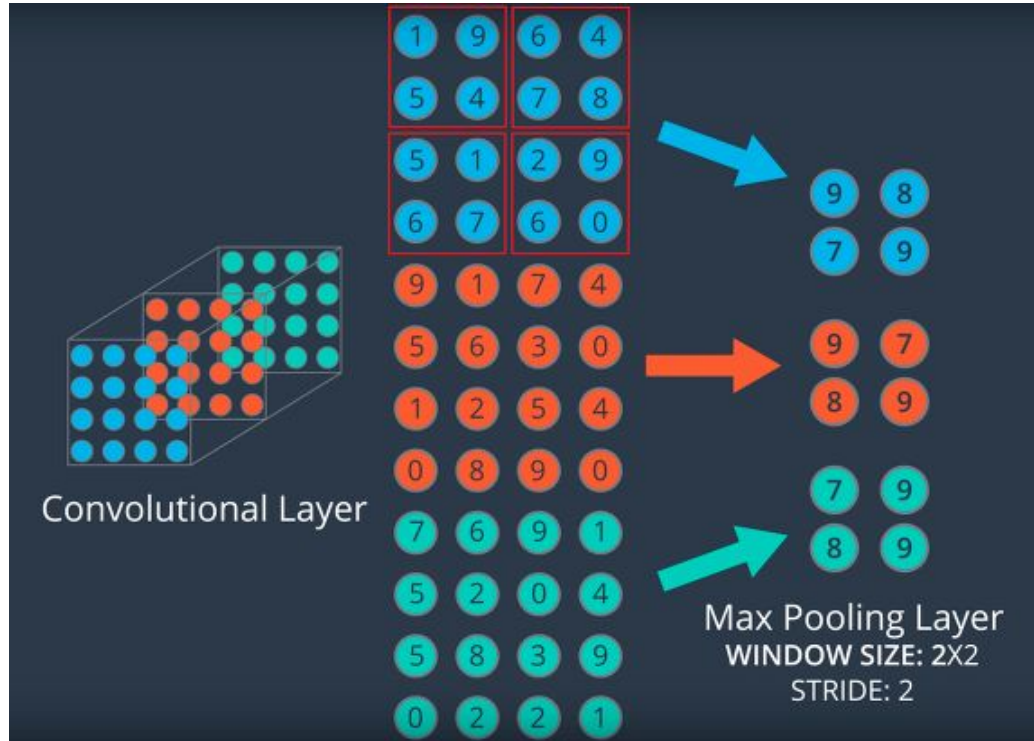
3x3 conv 동작 원리



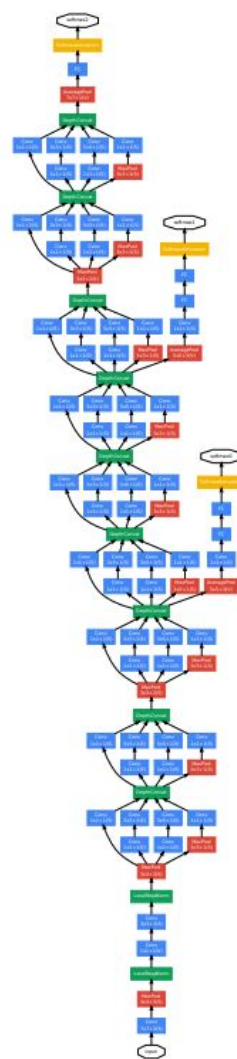
1x1 conv 동작 원리



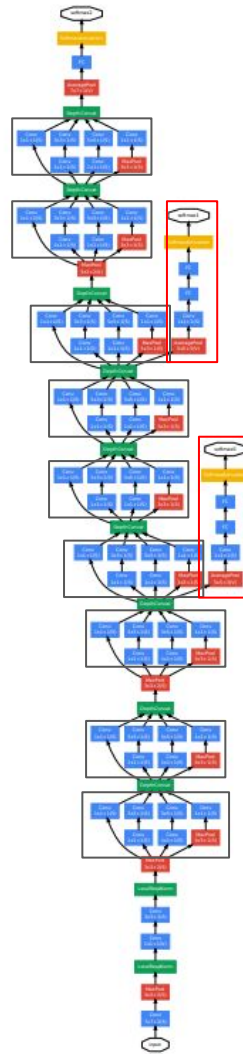
Max Pooling Layer



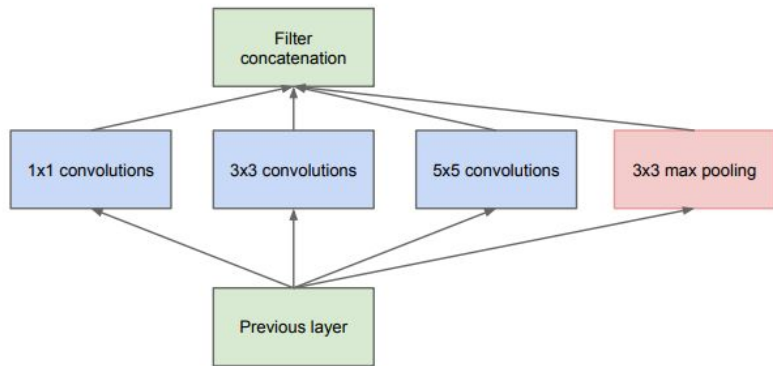
Architecture



Architecture

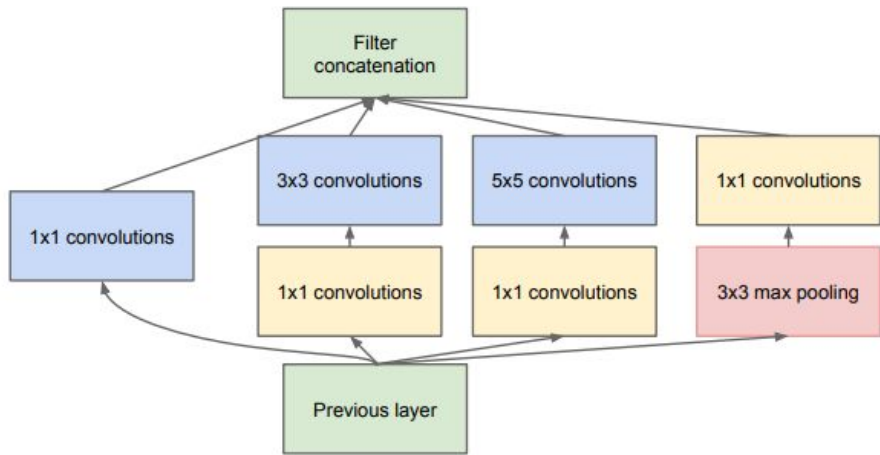


inception modules



(a) Inception module, naïve version

<기존의 inception 모듈의 모습>

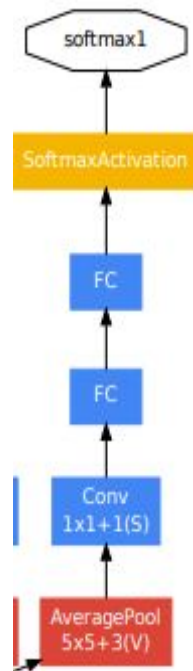


(b) Inception module with dimensionality reduction

<1x1 conv를 추가해 연산량을 줄여준 모듈의 모습>

auxiliary classifier

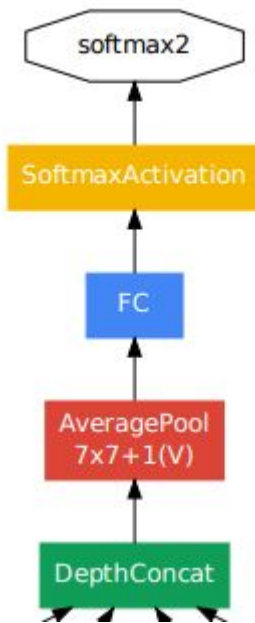
layer 중간중간마다 softmax를 뽑아서 back propagation을 해줌으로서 **overfitting**을 막고 모델의 성능을 향상시켰다.



Final Classification part

AveragePool(Global Average Pooling)

fc layer 에서 공간 정보가 없어지는 것을 막을 수 있고, 훨씬 공간적 정보를 잘 담을 수 있다고 한다.



GoogLeNet Architecture Details

reduce 가 붙은 것은
앞에 있는 모델의
channel 수를 줄여주는
1x1 filter이다.

| type | patch size/ stride | output size | depth | #1×1 | #3×3 reduce | #3×3 | #5×5 reduce | #5×5 | pool proj | params | ops |
|----------------|-----------------------|----------------|-------|------|----------------|------|----------------|------|--------------|--------|------|
| convolution | 7×7/2 | 112×112×64 | 1 | | | | | | | 2.7K | 34M |
| max pool | 3×3/2 | 56×56×64 | 0 | | | | | | | | |
| convolution | 3×3/1 | 56×56×192 | 2 | | 64 | 192 | | | | 112K | 360M |
| max pool | 3×3/2 | 28×28×192 | 0 | | | | | | | | |
| inception (3a) | | 28×28×256 | 2 | 64 | 96 | 128 | 16 | 32 | 32 | 159K | 128M |
| inception (3b) | | 28×28×480 | 2 | 128 | 128 | 192 | 32 | 96 | 64 | 380K | 304M |
| max pool | 3×3/2 | 14×14×480 | 0 | | | | | | | | |
| inception (4a) | | 14×14×512 | 2 | 192 | 96 | 208 | 16 | 48 | 64 | 364K | 73M |
| inception (4b) | | 14×14×512 | 2 | 160 | 112 | 224 | 24 | 64 | 64 | 437K | 88M |
| inception (4c) | | 14×14×512 | 2 | 128 | 128 | 256 | 24 | 64 | 64 | 463K | 100M |
| inception (4d) | | 14×14×528 | 2 | 112 | 144 | 288 | 32 | 64 | 64 | 580K | 119M |
| inception (4e) | | 14×14×832 | 2 | 256 | 160 | 320 | 32 | 128 | 128 | 840K | 170M |
| max pool | 3×3/2 | 7×7×832 | 0 | | | | | | | | |
| inception (5a) | | 7×7×832 | 2 | 256 | 160 | 320 | 32 | 128 | 128 | 1072K | 54M |
| inception (5b) | | 7×7×1024 | 2 | 384 | 192 | 384 | 48 | 128 | 128 | 1388K | 71M |
| avg pool | 7×7/1 | 1×1×1024 | 0 | | | | | | | | |
| dropout (40%) | | 1×1×1024 | 0 | | | | | | | | |
| linear | | 1×1×1000 | 1 | | | | | | | 1000K | 1M |
| softmax | | 1×1×1000 | 0 | | | | | | | | |

Table 1: GoogLeNet incarnation of the Inception architecture.