구글넷은 구글에서 만들었고 2014년 이미지넷 인식대회에서 VGG를 이기고 우승한 알고리즘이다.

우승은 구글넷이 하였고 VGG는 준우승을 하였지만 구조의 간결함과 사용의 편이성으로 인하여 VGG가 구글넷보다 더 각광받았다.

구글넷은 신경망 내부 컴퓨터 자원의 활용을 높이기 위해 계산량을 일정하게 유지하면서 신경망의 깊이와 폭을 늘려도 연산량이 증가하지 않고 유지되도록 설계되어 있다.

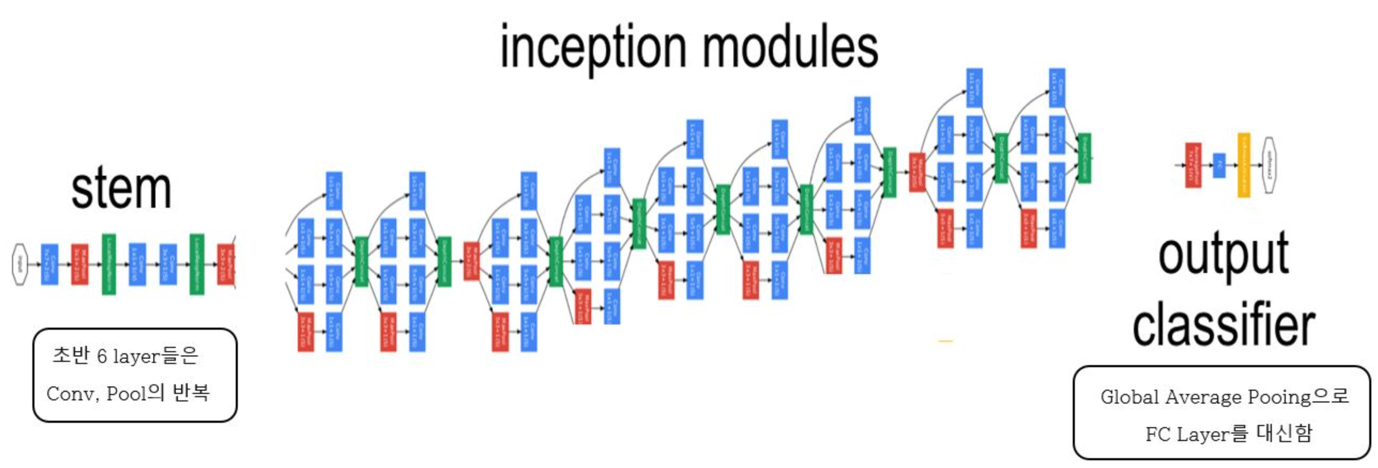
성능을 최적화하기 위해 hebbian principle과 multi-scale-processing을 적용하였다.



구글넷은 22개의 층으로 구성되어 있고 Error rate는 6.7 %이다

<inception modules>

연산을 효율적으로 수행하기 위해 inception modules을 쌓아서 아키텍처를 만든다.



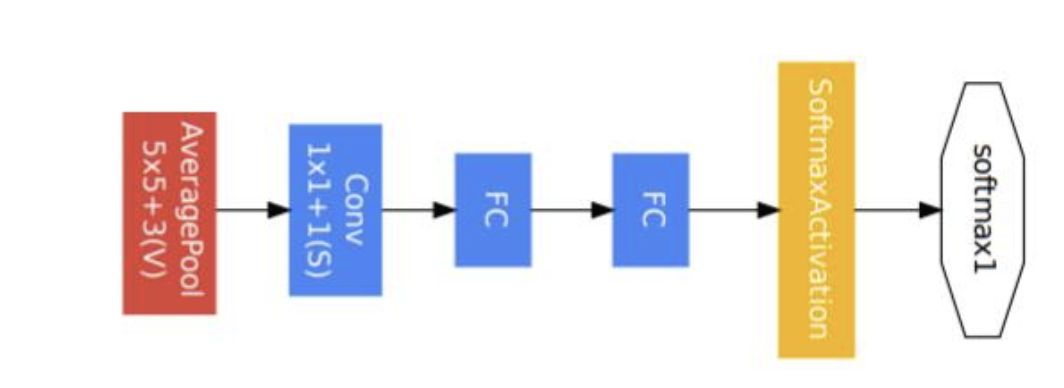
첫 번째의 6개의 layer는 conv, pool layer의 반복을 쌓고 steam이라 부른다.

두 번째는 inception module을 9개 층으로 쌓았고 마지막에 softmax로 클래스를 분류하는 output classifier 영역으로 쌓는다. output classifier 영역에 FC layer 대신 Global Average Pooling을 쓰고 학습할 parameter의 양을 줄인다.

<auxiliary classifier>









inception module들에는 보조분류기가 달려있다.

신경망이 22층으로 되어있기 때문에 기울기가 역전파되는 도중 소실이 될 수 있으므로, 신경망 중간에 예측 결과 loss를 삽입해줌으로써, 기울기 소실 문제를 해결한다.

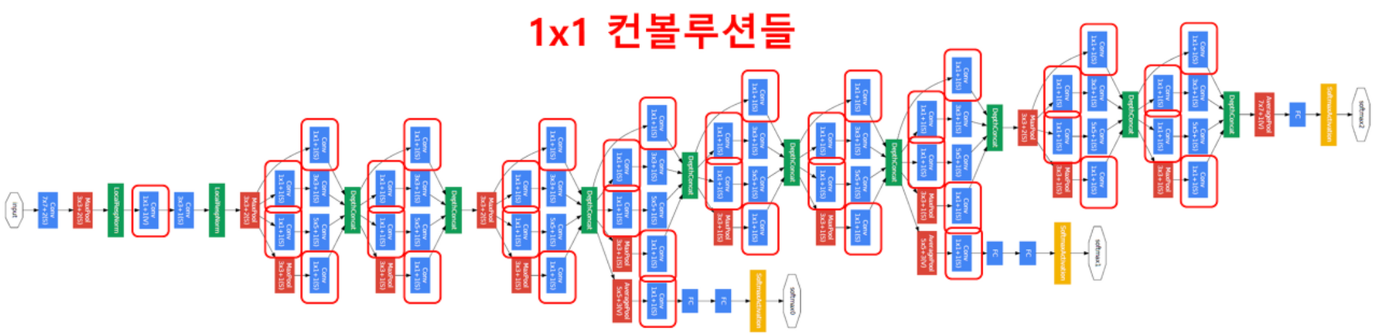
최종 loss는 중간에 전파되는 손실함수 값과 output classifier에 전파되는 손실함수의 값의

평균으로 구한다.

\*역전파란?

- 출력값에 대한 입력값의 기울기를 출력층 layer에서부터 계산하여 거꾸로 전파시키는 것이다.

<1 x 1 convolution>

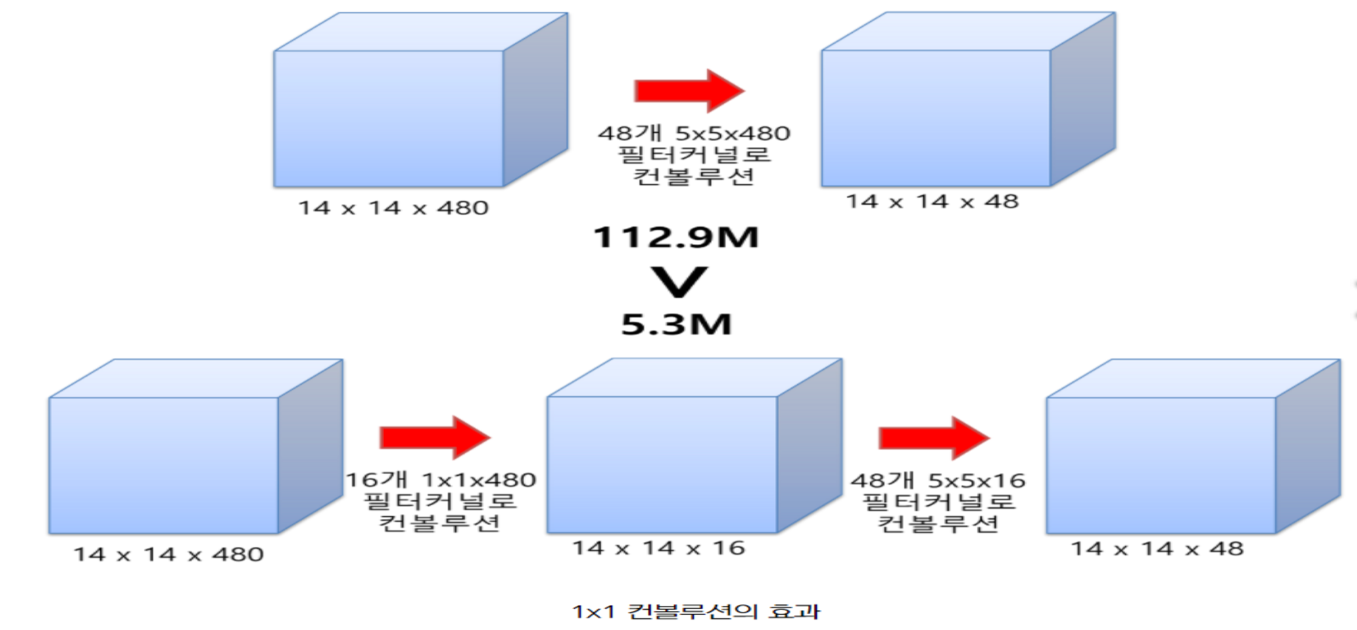


1 x 1 사이즈의 필터로 컨볼루션 해주는 것을 주목해서 봐야한다.

1 x 1 컨볼루션은 특성맵의 갯수를 줄이는 목적으로 사용이된다. 특성맵의 갯수가 줄어들면

그만큼 연산량이 줄어든다.

\* 연산량을 줄일 수 있다는 것은 네트워크를 더 깊이 만들 수 있게 도와준다.





1번

14 x 14 사이즈의 특성맵이 480장 있다고 가정한다.

48개의 5 x 5 x 480의 필터커널로 컨볼루션을 해주면 48장의 14 x 14 x 48의 특성맵이 된다.

(zero padding = 2, 컨볼루션 보폭 = 1로 설정)

이 때 필요한 연산은 ( 14 x 14 x 48 ) x (5 x 5 x 480) = 112.9M이 된다.

2번

14 x 14 사이즈의 특성맵 480장이 있다고 가정한다.

16개의 1 x 1 x 480의 필터커널로 컨볼루션을 해주면 14 x 14 x 16의 특성맵이 생성된다.

14 x 14 x 16 특성맵을 48개의 5 x 5 x 16의 필터커널로 컨볼루션을 해주면 14 x 14 x 48의

특성맵이 생성된다.

이 때 필요한 연산은 (14 x 14 x 16) x (1 x 1 x 480) + (14 x 14 x 48) x (5 x 5 x 16) = 5.3M이

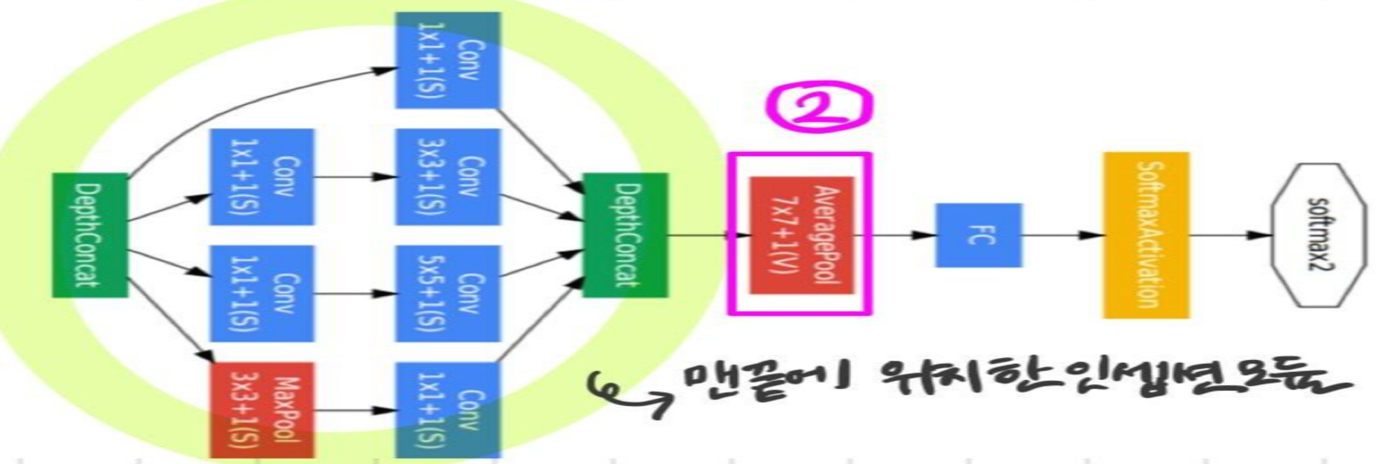
된다.

1 x 1 컨볼루션이 없을 때와 있을 때는 결과적으로 산출된 특성맵의 크기와 깊이가 같다.

2번이 1번보다 연산량이 적다는 것은 네트워크를 더 깊이 만들 수 있게 도와준다는 점에서

중요하다.

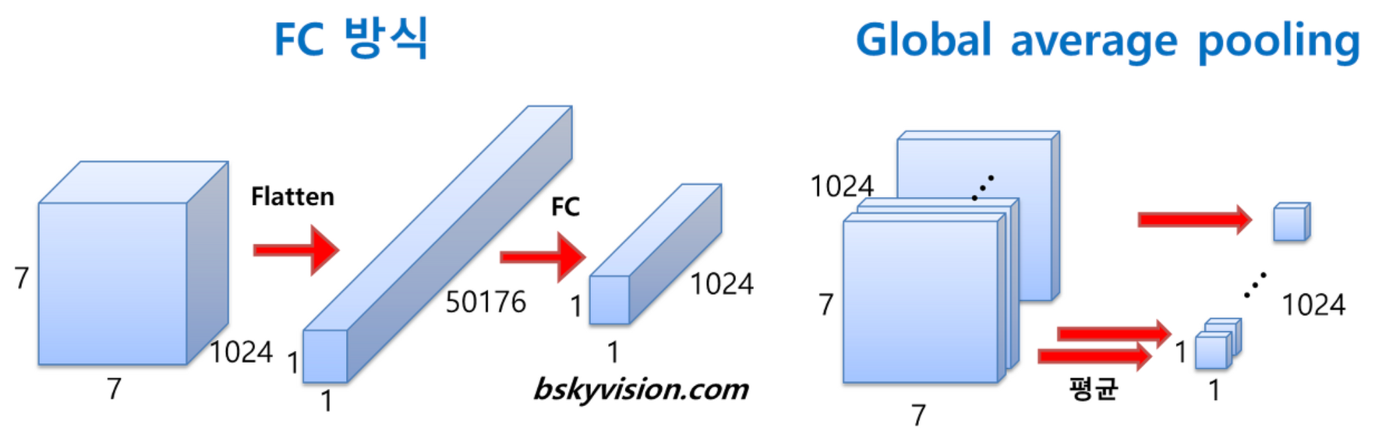
<global average pooling>



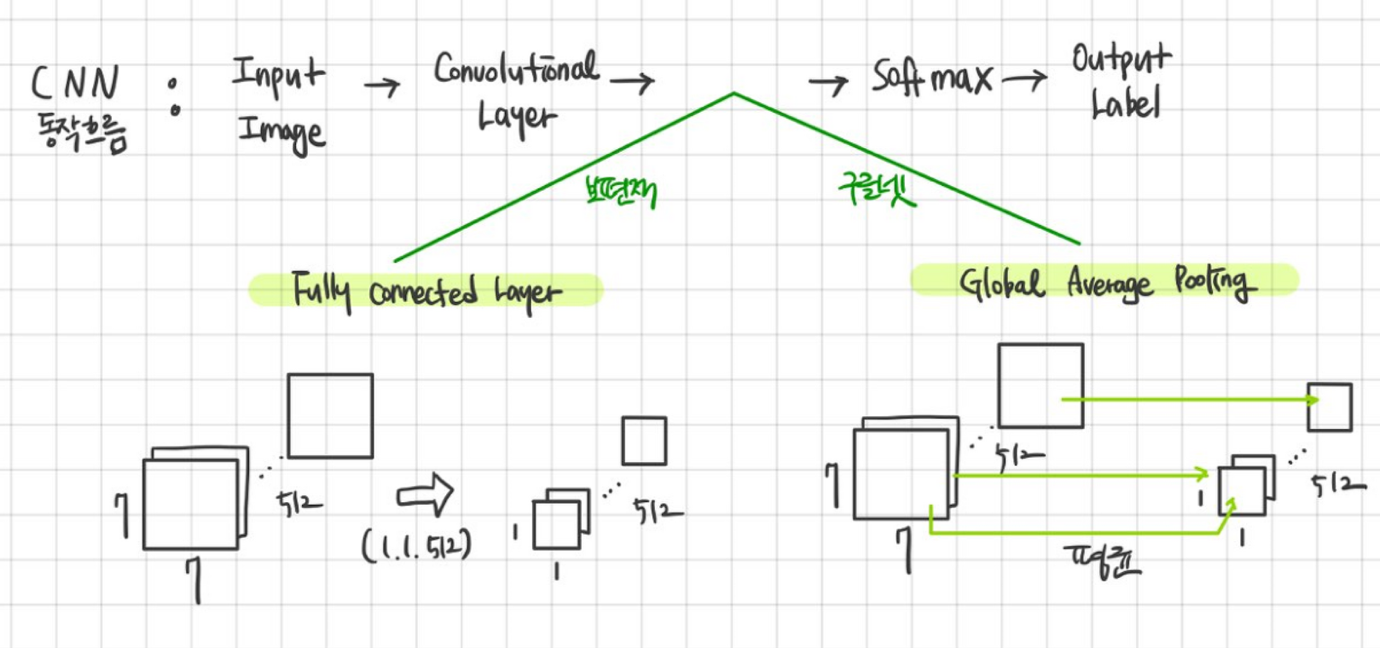
평균 풀링은 출력층에서 나타나며 분홍색으로 표시된 상자이다.

일반적으로 출력층 직전에 완전 연결 층(fully connected layer)을 배치하여 마지막 피처맵을 일렬로 펼친 뒤 softmax를 이용해서 분류를 한다.

구글넷에서는 완전 연결층 대신에 평균 폴링을 사용하여 피처맵을 벡터화 하여 softmax 처리를 한다.



평균하여 1차원 벡터로 만들면 가중치의 개수를 상당히 많이 줄여주는데 FC방식을 이용할 경우에는 가중치의 개수가 7 x 7 x 1024 x 1024 = 51.3M이지만, GAP를 사용하면 단 1개의 가중치도 필요하지 않다. 또한 GAP를 적용할 시 fine tuning을 하기 쉽게 만든다.



완전 연결층을 평균 폴링으로 대체하였을 경우 얻는 이점은 학습 매개변수가 훨씬 줄어든다는 것이다. 신경망에서 대부분의 매개변수는 완전 연결층으로 넘어가는 단계에서 생성된다. 그만큼 마지막 분류의 과정에서 연결해야 하는 가지의 수가 많다는 것인데 평균 폴링을 이용하면 입력

이미지가 무엇인지 분류하는 게 필요한 수많은 가지를 하나도 안 쓸 수 없다.   
폴링은 매개변수의 사용이 전혀 없기 때문에 학습이 필요없다.