

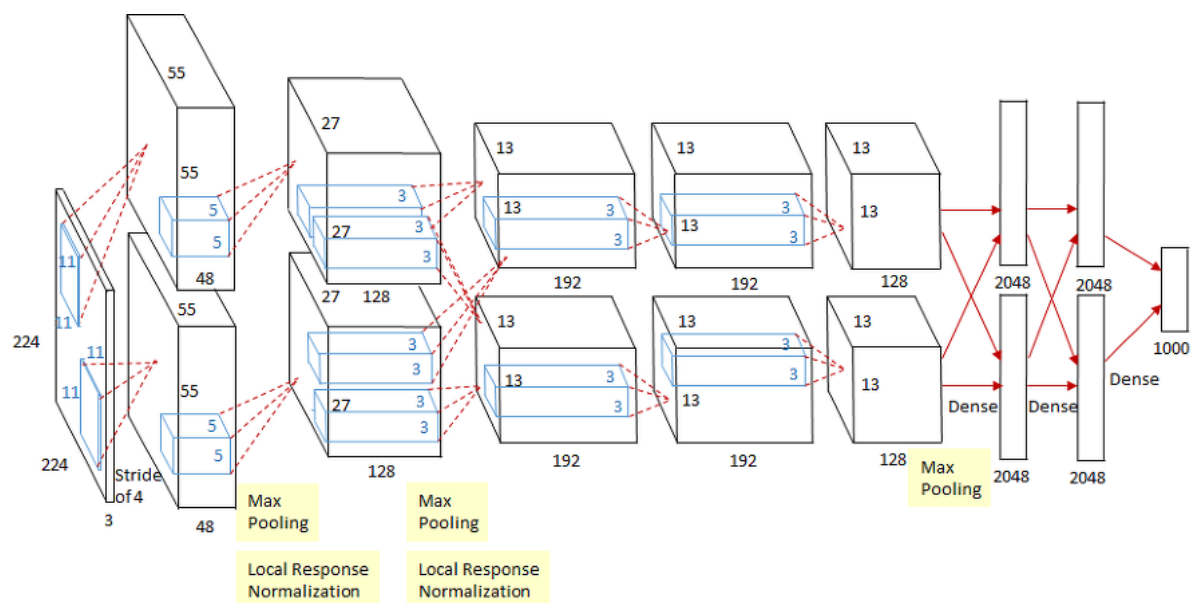
# 차량지능기초\_2

20173416 이민지

## 모델의 간단한 특징을 이해 및 조사

### AlexNet

AlexNet은 2012년에 개최된 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 대회에 우승을 차지한 컨볼루션 신경망(CNN) 구조입니다. 기존의 다른 어떤 방식보다 이미지 분류 (image classification) 분야에서 당시 오차율 16.4%로 압도적인 성능을 보여주었습니다.



(처음 224 \* 224 \* 3 의 이미지로 표시되어 있지만 227 \* 227 \* 3 입니다 )

구조를 보겠습니다.

구조를 보게 되면 총 8개의 레이어로 구성되어 있음을 알 수 있습니다. 5개의 컨볼루션 레이어와 3개의 fc레이어로 구성되어 있습니다. 2,4,5번째 컨볼루션 레이어는 전 단계와 같은

채널의 특성맵들과 연결되어있는 반면 3번째 컨볼루션 레이어는 전 단계의 두 채널의 특성맵들과 모두 연결되어 있습니다. 또한 풀링은 max, average pooling 중 최대값을 찾는 max pooling을 이용하였습니다. ReLU 활성화 함수를 적용하기 전에 Normalization을 적용하여 함수의 결과값에서 더 좋은 일반화 결과를 도출했습니다.

특징을 한번 살펴보겠습니다.

1. 우선 위의 그림에서 볼 수 있듯이 병렬적인 구조로 설계되었습니다. 2개의 GPU로 병렬 연산을 수행하기 위해서입니다.

2. 활성화 함수로 ReLU 함수를 사용하였습니다.

sigmoid를 필두로 Tanh 함수 사용하던 이전과는 달리 ReLU 함수를 사용하여 정확도를 유지하면서 약 6배의 빠른 성능을 가지게 되었습니다.

3. **fc레이어 (Fully Connected Layer)**에서 오버피팅을 막기 위해 드롭아웃을 사용하였습니다.

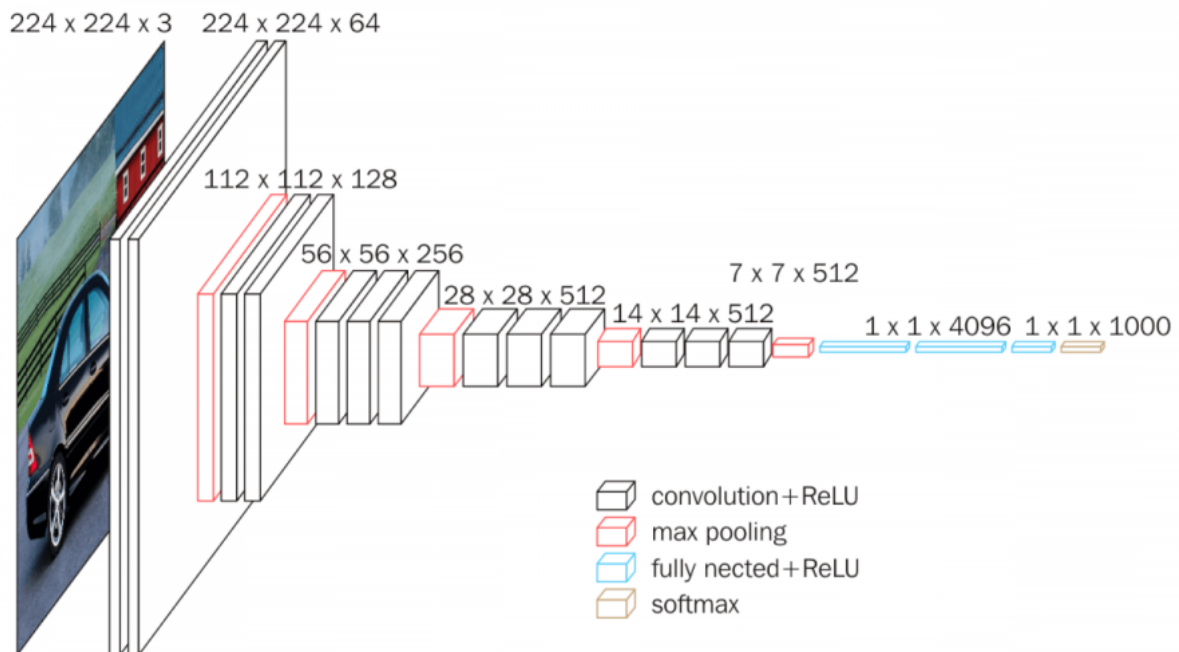
fc레이어를 학습하게 되면 크게 4가지의 문제점들이 발생하게 됩니다. 이미지의 높아지는 해상도에 따른 파라미터 수의 급증, 공간적 특성의 무시, 입력 데이터 변형의 취약, 변형된 이미지의 학습데이터의 과대 요구 등입니다. 따라서 학습 데이터에 지나치게 집중을 함으로 오히려 결과가 더 나쁘게 나올 수 있는 오버피팅이 발생할 수 있게 됩니다. 또한 복수개의 망을 학습시키는 것은 매우 힘든 작업이며 다양한 모델을 시행시킬 때 연산 시간을 잡아먹어 빠른 응답을 요구하는 데에 어려움이 있습니다. 따라서 네트워크의 일부를 무작위로 생략하여 학습을 방해함으로 학습이 학습용 데이터에 치우치는 현상을 막아 오버피팅을 막았습니다.

4. **Data augmentation** 기법을 적용하였습니다.

적은 양의 훈련 데이터에 인위적인 변화를 가해 새로운 훈련 데이터를 대량 확보하는 방법입니다. 예를 들어 이미지를 상하좌우로 뒤집거나 자르는 방식으로 새로운 이미지 데이터를 확보하였습니다.

## VGG16

VGGNet은 옥스포드 대학의 연구팀 VGG에 의해 개발된 모델로써 2014년 이미지넷 이미지 인식 대회에서 준우승을 한 모델입니다. 16개의 층으로 구성된 VGGNet 모델을 VGG16 이라고 합니다.



구조를 살펴보겠습니다.

input으로 224 \* 224 \* 3 의 이미지를 입력받았습니다.

15층까지는 ReLu 활성화 함수를 사용하였고 마지막 16층의 경우 출력값들을 softmax 함수로 활성화해주었습니다.

다음으로 특징입니다.

### 1. 컨볼루션 필터커널의 사이즈를 가장 작은 3 \* 3 으로 고정했습니다.

작은 사이즈의 필터를 사용하여 가중치/파라미터의 개수를 줄이고 망을 깊게 만들었습니다. 가중치가 작다는 것은 그만큼 훈련시켜야할 것의 개수가 작다는 것으로 학습의 속도를 향상시킬 수 있다는 것입니다.

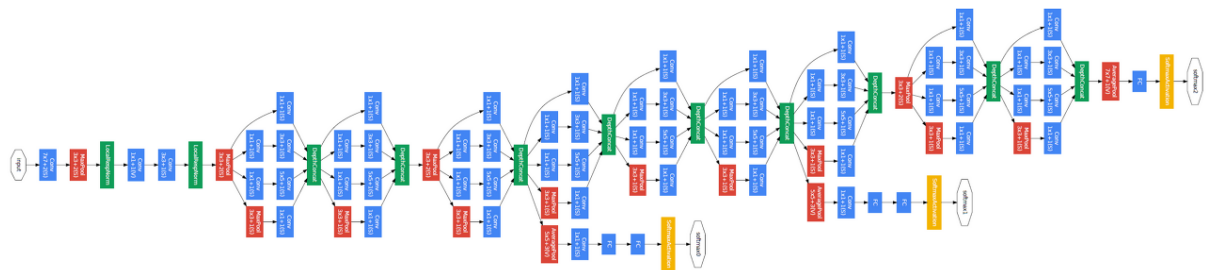
## 2. 더 많은 비선형성을 확보할 수 있었습니다.

3 \* 3 필터를 사용함으로써 깊은 모델을 만들 수 있게 되므로 더 많은 ReLu가 활성화 함수를 사용할 수 있게 되었습니다. 이 함수가 많아지면서 비선형성이 적용되어 학습의 효과를 증폭시킬 수 있었습니다.

## GoogLeNet

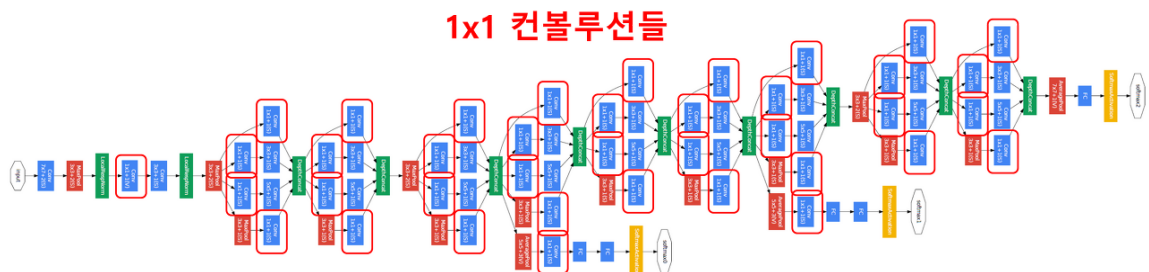
GoogLeNet은 2014년 ILSVRC에서 우승을 차지한 모델입니다. GoogLeNet은 22층으로 구성되었습니다.

.



특징을 살펴보겠습니다.

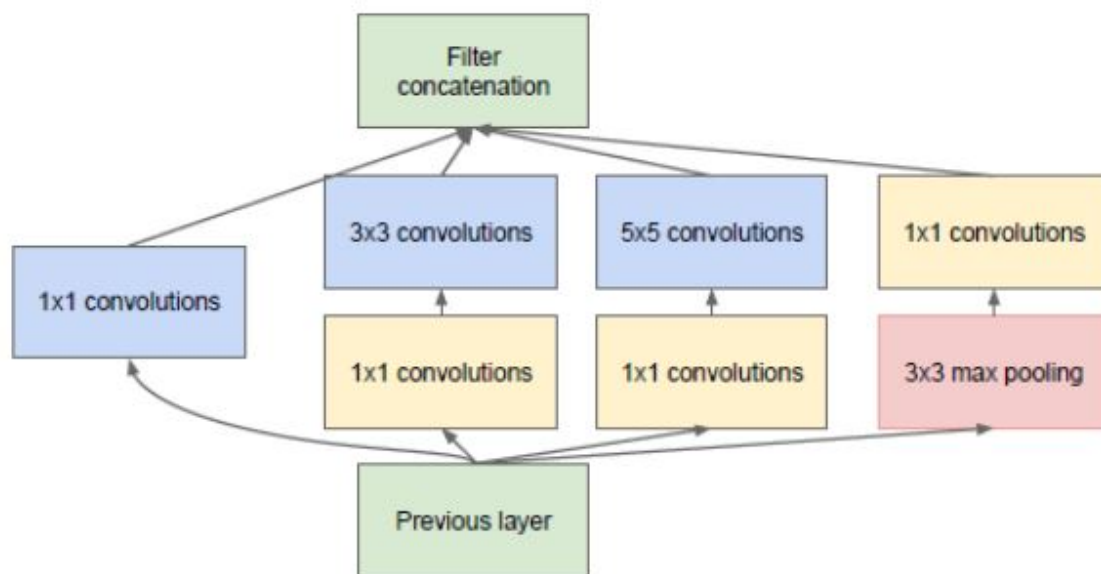
### 1. 1 \* 1 사이즈의 컨볼루션 필터커널을 사용하였습니다.



1 \* 1 사이즈의 필터를 사용함으로써 차원을 줄이고 망이 깊어졌을 때 연산량이 늘어나는 문제를 해결하였습니다.

### 2. Inception 모듈을 사용하였습니다.

GoogLeNet은 총 9개의 인셉션 모델을 포함하고 있으며 구조는 아래와 같습니다.



(b) Inception module with dimensionality reduction

1 \* 1 / 3 \* 3 / 5 \* 5 컨볼루션, 3 \* 3 최대풀링을 해준 결과 얻은 특성맵들을 모두 함께 쌓았고 결과적으로 좀 더 다양한 종류의 특성이 도출되었습니다.

### 3. global average pooling을 사용하였습니다.

global average pooling 방식은 전 층에서 산출된 특성맵들을 각각 평균낸 것을 이어서 1차원 벡터로 만들어주는 것입니다. 이를 통해 가중치의 개수를 상당히 줄일 수 있었습니다.

### 4. 보조 분류기(auxiliary classifier)를 사용하였습니다.

가중치를 훈련하는 과정에서 주로 역전파법을 활용합니다. 이 과정에서 가중치를 업데이트하는데 사용되는 gradient가 0이 되버려 가중치들이 제대로 훈련되지 않을 수 있습니다. 이러한 문제를 vanishing gradient 문제라고 합니다. 이 문제를 극복하기 위해 두 개의 보조 분류기를 사용하였습니다. 이 보조 분류기들은 훈련시에만 활용할 뿐 사용할 땐 제거되어집니다.

## ResNet18

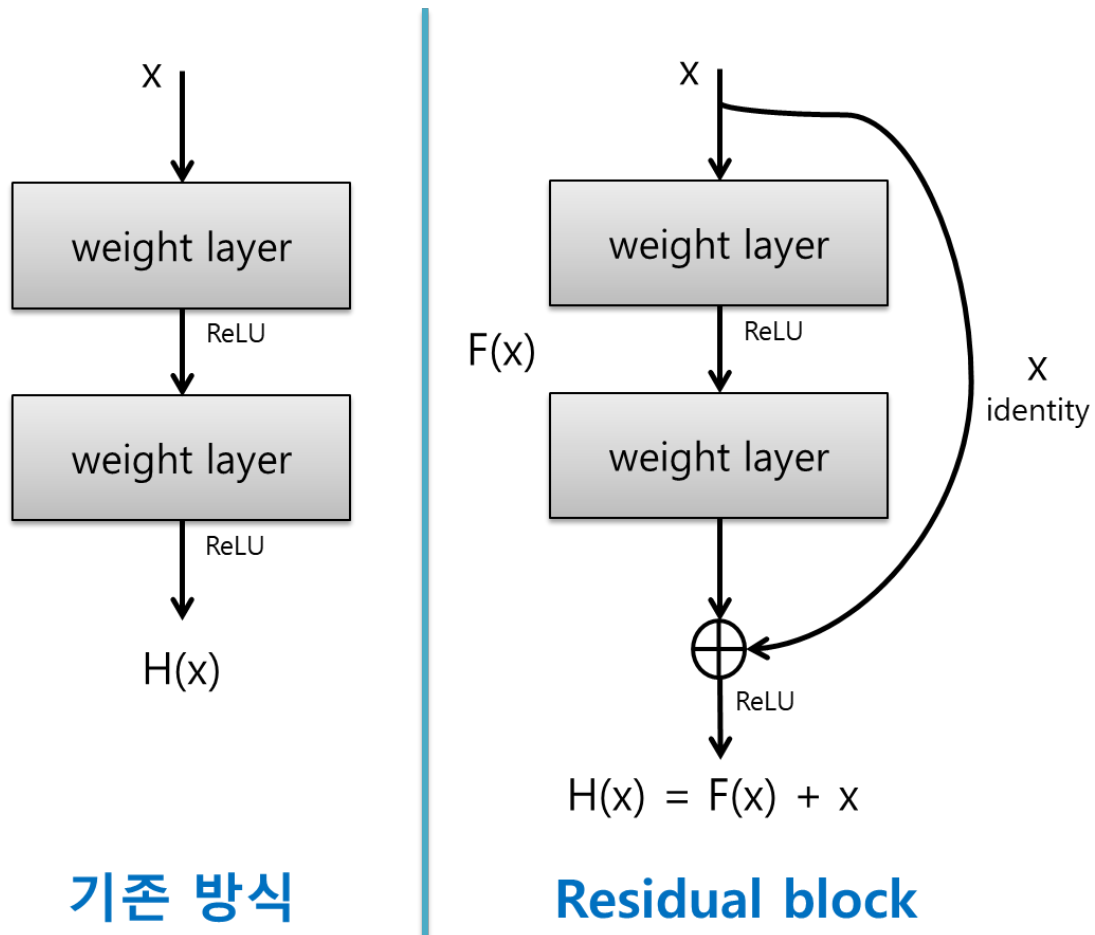
2015년 ILSVRC에서 우승을 차지한 마이크로소프트에서 개발한 알고리즘입니다.

ResNet18은 18개층으로 구성된 컨벌루션 신경망입니다.

Resnet 연구진들은 망을 무조건 깊게 한다고 성능이 좋아지지 않는다는 것을 확인하였고 새로운 방법을 고안해내었습니다.

특징을 살펴보겠습니다.

### 1. Residual Block을 사용하였습니다.



위의 그림과 같이 입력값을 출력값에 더해줄 수 있는 지름길을 하나 만들어주었습니다.  $F(x) + x$ 를 최소화하는 것을 목적으로 잔차인  $H(x) - x$ 를 최소로 해주는 것에 초점을 맞추었습니다.

## 2. 구조는 VGG19의 구조를 뼈대로 하였습니다.

처음의 층은 7 \* 7 사이즈의 컨볼루션 필터를 사용했고 나머지의 층은 3 \* 3 사이즈의 컨볼루션 필터를 사용하였습니다.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

## 각 모델 성능 비교

### AlexNet, VGG16, GoogLeNet, ResNet18

AlexNet top-1 percentage : 56.52% VGG16 top-1 percentage : 71.59%

AlexNet

VGG16

GoogLeNet top-1 percentage : 69.78% ResNet18 top-1 percentage : 69.76%

GoogLeNet

ResNet18

top-1 정확도를 보았을 때 VGG16 > ResNet18 > GoogLeNet > AlexNet 순으로 정확도가 높은 것을 확인할 수 있습니다.

torchvision을 이용하여 imagenet의 pretrained model을 classification 해볼 수 있었습니다.