Very Deep Convolutional Networks for Large-Scale Image Recognition

Karen Simonyan & Andrew Zisserman

Abstract

Convolution Network depth의 효과에 대해 조사를 한다. 작은 합성곱 필터를 사용하여 네트워크의 깊이를 늘렸을 때의 철저한 평가를 하는 것이다. 작은 합성곱 필터들은 16-19 가중치 layers 만으로도 이전 configurations의 상당한 효과를 달성할 수 있음을 보여준다.

1. Introduction

합성곱 network들은 최근에 이미지와 비디오 recognition에서 상당한 성공을 거뒀다. 그리고 이는 ImageNet과 같은 큰 공공 이미지 저장소와 고성능의 연산 시스템 덕분에 가능해졌다. 특히 ILSVRC 덕에 가능해졌는데 이는 몇몇 큰 규모의 이미지 분류 시스템에서 testbed로 활용되고 있다.

이러한 배경에서 기존 구조의 성능을 향상시키기 위한 시도들이 잇따르고 있다. 이 논문에서 ConvNet 구조의 깊이의 중요성에 주목하면서 진행해본다. 구조의 다른 파라미터들을 고정시키고 네트워크의 깊이를 증가시키면서 살펴보겠다. 그리고 이는 매우 작은 합성곱 필터들을 모든 layer에서 사용하기에 가능하다.

결과적으로, 더 나은 구조를 만들어냈다. 정확성을 더 확보한 것뿐만 아니라 다른 데이터들에 대해서도 활용이 가능했다.

2. ConvNet Configurations

2.1 Arichitecture

training시에는 인풋 이미지의 크기는 224x224이다. 유일한 전처리는 RGB 값의 평균을 빼는 것이다. 전처리 된 이미지는 합성곱 layers들을 통과하는데 3x3 사이즈의 작은 필터들을 사용한다. Stride는 1이고 spatial padding을 이용했다. Spatial pooling은 다섯 개의 max-pooling layers를 사용했는데 max-pooling은 stride 2로 2x2 픽셀로 진행하였다.

Convolution layers의 stack은 세 개의 FC layers를 사용하였다. 처음 두 개는 각각 4096개의 채널을 가지고 있고 세번째 층은 1000개의 ILSVRC 분류를 해야하므로 1000개의 채널을 갖고 있다. 마지막 층은 soft-max layer이다.

모든 은닉층은 ReLU 비선형을 갖고 있다. 어떠한 network도 local response normalization 정규화를 하지 않았다. 성능 향상을 이끌지는 않지만 메모리 증가와 계산 시간 증가를 야기한다.

2.2 Configurations

Table 1에 ConvNet configuration이 나와 있다. 모든 것들은 깊이에서만 다르다. Table 2에서는 모수의 수를 보여주는데 깊이에도 불구하고 필요한 가중치는 큰 차이가 없었다.

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

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					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

2.3 Discussion

다른 architecture와의 큰 차이는 7x7 필터 대신에 3x3 필터 3개를 사용한 것이다.

이에 대한 효과는?

첫 번째로, 3개의 비선형 rectification을 층을 활용할 수 있으니 결정 함수를 더욱 확실하게 만들 수 있다. 두 번째로, 필요한 파라미터 수를 줄일 수 있다. Table 1에 C와 같이 1x1 합성곱 layer을 넣는 것은 필터에 영향을 주지 않으면서 결정 함수의 비선형성을 높일 수 있는 방식이다. 비록

1x1 합성곱이 같은 차원 공간에 선형 projection일지라도 추가적인 비선형성은 rectification function에 의해 가능하다.

3. Classification Framework

3.1 Training

훈련은 momentum을 활용한 미니-배치 경사 하강법을 이용한 다중 로지스틱 회귀를 이ㅛㅇ해 최적화시켰다. 처음 두 개의 FC layers에 대해서는 Weight decay와 drop out을 통해 규제되었다. Learning rate는 0.01 수준으로 설정되었고 정확도가 오르는 것이 감소하였을 때는 a factor 10으로 학습률이 감소하였다. 비록 필요한 파라미터가 많고 network의 깊이도 깊었지만 더 깊은 깊이와 작은 필터 사이즈에 내포된 규제와 특정 층에 대한 사전 초기화 덕분에 epoch를 더 적게 요구하였다.

가중치의 초기화는 중요하다. 옳지 않은 초기화는 깊은 망에서 기울기의 안정성 때문에 학습을 안 하려 하기 때문이다. 이러한 문제를 벗어나기 위해 랜덤 초기화로 충분히 얕게 훈련시키기 시작했다. 그 이후 더 깊은 구조를 훈련시킬 때, 첫 번째 네 개의 합성곱 층과 마지막 세 개의 FC 층들을 A의 층들로 함께 초기화시켰다. 중간층들은 아직도 랜덤하게 초기화된 상태이다. 미리 초기화된 layer에 대해서 학습률을 감소시키지 않았고 훈련하는 동안 바뀌도록 허용하였다. 랜덤한 초기화는 평균이 0이고 분산이 0.01인 정규분포에서 샘플링하였다. bias들은 0으로 초기화하였다.

고정된 224x224 인풋 이미지를 얻기 위해 rescale된 훈련 이미지에서 랜덤하게 crop해왔다. 더 증강하기 위해서 crop들은 random horizontal flipping과 random RGB colour shift를 하였다.

S를 isotropically-rescaled training image의 가장 작은 면이라고 하자. Crop 사이즈가 224x224로 고정되어 있는 와중에 원측 적으로 S는 224보다 작지 않은 어떤 값을 가져도 된다. 224인 경우엔 crop은 whole-image statistics를 포착할 것이다. 224보다 큰 경우에는 crop은 이미지의 작은 부분에 대응할 것이다.

S를 세팅하기 위한 두 가지 방식이 있다. 첫 번째는 S를 고정시키는 것으로 single-scale training에 상응한다. 위 실험에서는 256과 384로 고정시킨 두 가지 방식을 이용하여 보았다. 두 번째 접근은 S를 multi-scale training으로 세팅하는 것이다. 각 훈련 이미지는 랜덤하게 샘플링한 S에서 각각 rescale 되어진다. 이미지에서 물체들이 다른 사이즈를 가질 수 있으므로 이걸 고려하는 것이 낫다. Scale jittering을 통해 훈련 셋들을 증강시키는 것으로 볼 수 있다.

3.2 Testing

Test time에서 훈련된 ConvNet과 입력 이미지가 주어졌을 때 다음과 같은 방식으로 분류되었다. 첫 번째 Q라고 칭해지는 미리 정의된 가장 작인 이미지면에 isotropically rescale된 것이다. Q는 S

와 꼭 같을 필요는 없다. FC layer들이 합성곱 층으로 처음 변환된다. 그 결과 net은 전체 이미지에 적용된다. 결과는 class 개수와 동일한 channel의 class score map과 인풋 이미지 크기에 의존하는 variable spatial resolution이다. 마지막으로, 이미지에 대해 고정된 크기의 벡터를 얻기 위해 class score map은 spatially average된다. 우리는 또한 test 셋의 데이터를 증강하였고 원래 이미지의 점수와 flip된 이미지의 점수를 평균내어서 얻었다.

합성곱 네트워크가 모든 이미지에 적용되기 때문에 테스트 타임에서는 각 crop마다 network 재계산을 요구하기에 다수의 crop들을 샘플링 할 필요 없다. 동시에 많은 crop 셋을 활용하여 향상된 정확성을 이끌 수 있다. 또한 다른 convolution boundary condition 때문에 multi-crop 평가는 dense 평가와 상호 보완적이다. ConvNet을 crop에 적용할 때, convolve 된 feature map은 0으로 padding 된다. 반면에 dense evaluation 경우네는 같은 crop에 대한 padding이 이미지의 neighbouring part로부터 온다. 이는 서서히 overall network receptive filed를 증가시켜 더 많은 context가 포착된다. 실제로 다수의 crop들의 증가된 계산 시간이 정확성에서 향상을 보여준다고 정당화할 수 없다.

3.3 Implementation Details

Implementation은 공공히 가능한 C++ Caffe toolbos로부터 온다. 하지만 많은 중요한 수정을 하였고 훈련과 평가를 다수의 GPU에서 할 수 있도록 해준다. 또한 full-size에 대한 훈련과 평가를 multiple scale에서 가능하게 한다. Multi-GPU 훈련은 data parallelism을 exploit시키고 훈련 데이터들의 batch들을 몇몇 GPU 배치로 분할함으로써 행해진다. GPU batch 기울기가 계산되고 나면, full batch의 기울기를 얻기 위해 평균화 된다. 기울기 계산은 GPU상에서 synchronous 다순 GPU 상에서 훈련했을 때와 결과가 같다.

다수의 sophisticated 방법들이 최근에 제안되고 있지만 더 빠른 걸 찾는 건 쉽지 않다.

4. Classification Experiments

Dataset

Dataset은 1000개의 class를 가진 이미지들로 구성되는데 training, validation, testing 세 개로 나눈다. 분류 성능은 top-1과 top-5 오류 두 가지 방법을 이용해 평가된다. 전자는 multi-class 분류 오류를 말하고 후자는 주로 사용되는 분류 기준이다.

4.1 Single Scale Evaluation

Local response normalisation이 그렇게 큰 효과를 보이지 못한다는 것을 알았다. 또한 depth가 깊어질수록 분류 에러가 감소한다는 것을 알았다. 비선형 층을 추가하는 것도 의미가 없다는 것을 알아챘다. Non-trivial receptive filed에 합성곱 필터를 사용하는 것은 spatial context를 포착하는 데

중요하다. 깊고 작은 필터를 사용하는 것이 얕고 큰 필터를 사용하는 network보다도 성능이 좋다. 고정된 scale을 사용하는 것보다 scale jittering을 하는 것이 더욱 성능이 좋았다.

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

4.2 Multi-Scale Evaluation

훈련 scale과 test scale의 큰 차이가 오히려 모델 성능을 감소시킨다는 것을 고려하였을 때, 고정된 S가 평가시에 사용된다. 결과는 scale jittering이 더 나은 성능을 보였다.

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

4.3 Multi-crop evaluation

Multiple crop을 이용하는 것은 dense evaluation보다 성능이 좋았고 그 둘의 방법은 상호보완적이다.

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

4.4 ConvNet Fusion

이 실험 부분에서는 몇몇 모델들의 output들을 조합할 것이다. 이는 모델의 상호보완성 덕분에 성능이 향상될 것이다.

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

4.5 Comparison with the State of the Art

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

5. Conclusion

큰 규모의 이미지 분류에 매우 깊은 합성곱 신경망을 평가하였다. Representation depth가 분류 정확성에 의미가 있음을 말하며 원래 구조에 깊이를 더 더함으로써 더 나은 성능을 보였다.