2022.01.20

ImageNet Classification with Deep Convolutional Neural Networks

- Alex Krizhevsky, Ilya Sutsker, Geoffrey E. Hinton

Abstract

ImageNet LSVRC-2010 콘테스트에서 좋은 성능을 보인 네트워크를 학습시켰다.

6천만개의 모수와 650,000개의 뉴런을 사용했고 5개의 convolutional layers인데 이는 1000-way softmax와 함께 max-pooling layers, 3개의 fully-connected layers로 구성되어 있다. 더 빠르게 하기 위해 non-saturating 뉴런과 GPU를 사용했다. 과적합을 막기 위해 '드롭아웃'이라는 정규화 방법을 사용하기도 했다.

1. Introduction

객체를 감지하는 것은 머신러닝 기법에서 중요해지고 있다. 이 성능을 향상시키기 위해선 많은 데이터, 강력한 모델을 구축하고 과적합을 방지하기 위한 나은 기술을 사용하는 것이 중요하다.

대부분 데이터를 많이 갖게 됨으로써 많은 문제들이 해결하고 데이터가 없기에 발생하는 문제점들은 널리 알려져 있다. 간단한 인식 문제는 적은 사이즈로도 가능하나 현실에서는 사물들이 상당한 변동을 갖고 있고 최근에 와서야 수백만 개의 이미지와 함께 라벨링된 데이터셋을 모으는 게 가능해졌다.

수백만개의 이미지를 학습하기 위해선 높은 능력이 필요하고 우리가 갖고 있지 않은 데이터 모두에 대해 보충하기 위해선 사전 지식이 필요하다. CNN은 depth와 breadth를 달리함으로써 통제가 되고 이미지 본질에 대해 더욱 정확한 가정이 가능해진다.

비슷한 층을 가진 feedforward 뉴런 네트워크와 비교하였을 때 CNN은 더 적은 연결과 모수로도 비슷한 성능을 낼 수 있다.

이러한 장점에도 불구하고 고화질 사진에 대해 비용이 많이 든다는 단점이 있다. 다행히 최근에는 GPU 성능도 좋아지고 ImageNet과 같은 최신 데이터셋이 라벨과 함께 나와 과적합 없이 모형을 훈련시킬 수 있다.

이 논문의 contribution은 다음과 같다.

가장 큰 CNN 중 의 하나를 ImageNet의 일부에 대해 학습시켰다. 2D convolution의 최대 최적화

GPU 실행에 대해 작성하였다. 모형 성능을 향상시키고 훈련 시간을 절약시켜주는 흔치않은 feature들을 사용하였다. 과적합을 방지하기 위해 몇몇 효과적인 기술들을 사용했다.

2. The Datasets

ImageNet : 22,000개의 카테고리에 속하는 1500만 개의 라벨이 달린 고화질 이미지 보유, 웹에서 얻어진 이미지들, 사람들이 라벨을 달았다, 2010년부터 ILSVRC가 매년 개최되었다. 120만개의 training images, 50,000개의 validation images, 150,000개의 testing images가 있다.

ILSVRC-2010은 ILSVRC 중 유일하게 테스트 셋 라벨이 사용가능한 버전이다. 2012 버전은 이용할 수 없다.

이미지마다 화질이 다르다. 차원은 같다. 따라서 이지미를 256 x 256 resolution으로 down-sample 하였다. 각 픽셀에서부터 평균 activity를 빼는 것을 제외하고는 어떤 전처리도 하지 않았다. 픽셀의 raw RGB values에 기반에 네트워크를 학습시킨다.

3. The Architecture

Figure2가 구조를 간단히 보여준다.

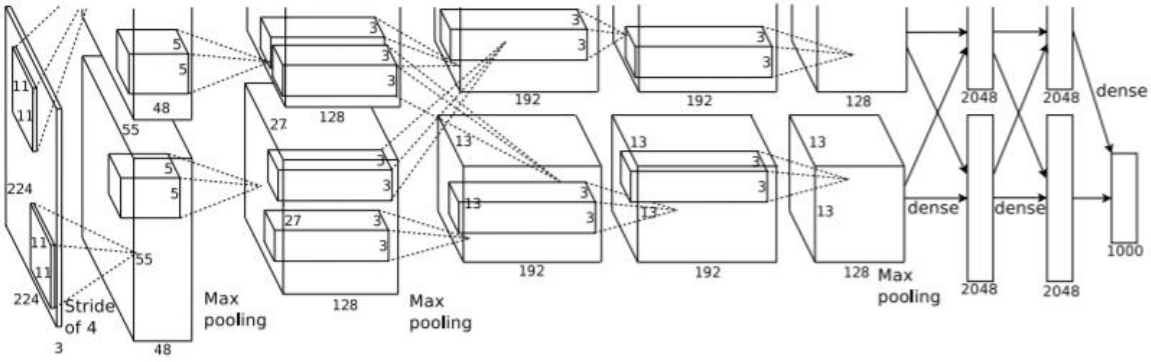

Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

3.1 ReLU Nonlinearity

보편적인 방법은 tanh를 사용하는 것이다. 하지만 이러한 saturating 비선형은 non-saturating 비선형(max(0,x))보다 느리다. 이것을 Nair와 Hinton에 따라 Rectified Linear Units라고 부르도록 하겠다. 그리고 이 ReLU는 tanh와 비교하였을 때 deep convolutional neural network를 더 빠르게 학습한다. 다음 figure 1에서 확인할 수 있다.

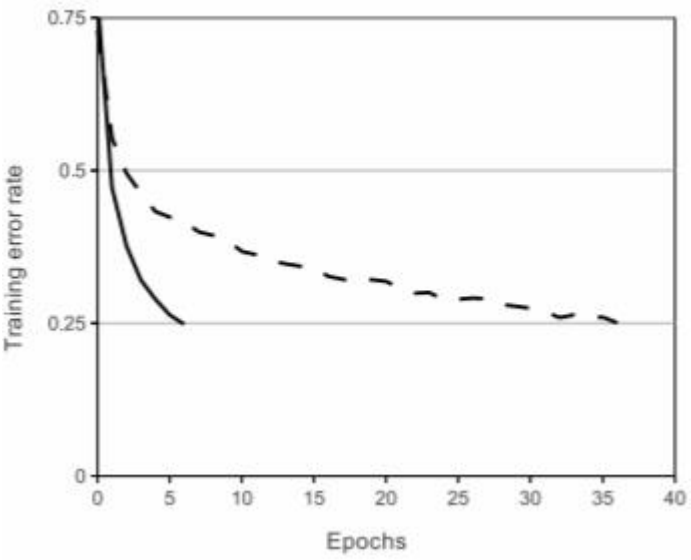

Figure 1: A four-layer convolutional neural network with ReLUs (**solid line**) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons (**dashed line**). The learning rates for each network were chosen independently to make training as fast as possible. No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons.

이전에도 Jarrett이 |tanh(x)|를 주장하였다. 하지만 이것이 적용된 Caltech-101 dataset의 주요 관심은 과적합을 방지한느 것이었고 학습 속도를 빠르게 하는 것과는 달랐다. 빠른 학습은 큰 데이터 셋에 대해 큰 모델의 성능에 효과적은 영향을 미친다.

3.2 Training on Multiple GPUs

하나의 GTX 580 GPU는 3GB 메모리만 갖고 있고 이는 훈련할 수 있는 네트워크의 최대 사이즈를 제한한다. 120만 개의 training example이면 네트워크를 훈련시킬 수 있기에 충분한데 하나의 GPU에 적합시키기엔 너무 크다. 그래서 넷을 두 개의 GPU로 퍼뜨렸다. 현재 GPU들은 cross-GPU parallelization(병렬화)에 적합하다. Host machine memory를 통과할 필요없이 다른 메모리로부터 직접 읽고 쓰는 게 가능하기 때문이다. Paralleization 계획은 kernel(뉴런)의 절반을 각각의 GPU로 할당하는 것이다. 여기서 하나의 트릭은 GPU는 특정 layers에서만 소통한다는 것이다. 예를 들어 layer 3의 커널은 layer 2의 모든 커널 맵의 input을 가지지만 layer 4의 커널들은 같은 GPU에 존

재하는 특정 커널 맵의 input만을 가진다. 연결의 패턴을 고르는 것은 cross-validation의 문제이지만 이것은 수용할 수 있는 계산의 양만큼 소통의 양을 조정하도록 한다.

3.3 Local Response Normalization

ReLU들은 saturating을 방지하기 위해 input을 정규화해야 하지 않아도 된다는 바람직한 특성이 있다. 하나의 훈련 샘플이라도 ReLU에 양의 input 값을 만들어낸다면 학습은 뉴런에서 발생할 것이다. 하지만, 일반화를 도와주는 지역적 정규화가 있다.

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

n-같은 공간 위치에서 인정한 커널 맵

N-층의 모든 커널 수

커널 맵의 순서는 임의적이고 훈련이 시작되기 전에 결정된다. 이러한 종류의 정규화는 실제 뉴런에서 발견되는 타입에 영감을 받아 측면의 억제(lateral inhibition, 신경생물학에서 활성화된 뉴런이 주변 이웃 뉴런들을 억누르는 현상)를 시행한다. 이는 다른 커널들을 사용해 계산된 뉴런 output들 중 가장 큰 activities를 위해 경쟁을 만든다(강하게 활성화된 뉴런의 주변 이웃들에 대해 normalization을 실행한다.). 주변에 비해 어떤 뉴런이 비교적 강하게 활성화된다면 그 뉴런의 반응은 더 돋보일 것이고 강하게 활성화된 뉴런 주변도 모두 강하게 활성화된다면 모두 값이 작아질 것이다. K, n, alpha, beta는 모두 하이퍼-파라미터이다. 이러한 것은 Jarrett의 loak contrast normalization 계획과 많이 닮았다. 하지만 이 논문은 mean activity를 빼지 않았기에 brightness normailization이라고 부르는 것이 더욱 정확하다.

3.4 Overlapping Pooling

CNN의 pooling layers는 같은 커널 맵에서 이웃하는 뉴런의 집단들의 output을 요약한다. 전통적으로 인정하는 pooling unit에 의해 요약되는 이웃들은 overlap되지 않는다. 더 정확히 말하면, pooling layer는 s 픽셀들 떨어진 pooling unit들의 격자로 구성되어 있다고 볼 수 있다. 그리고 각각은 z*z 사이즈의 이웃을 요약한다. 이것은 기존의 maxpooling에 픽셀을 겹치게 하여 과적합을 예방한다.

3.5 Overall Architecture

가중치가 달린 8개의 layers가 있고 처음 다섯 개는 convolutional이고 나머지 세 개는 fully-connected layers이다. 마지막 fully-connected 층에서의 아웃풋은 1000-way softmax로 주어지고 1000개의 클레스 라벨에 대해 분포를 제공한다. 네트워크 목표는 로지스틱 회귀 목표를 최대화하려 한다. 이는 예측 분포하에서 정확한 라벨의 로그 확률의 훈련 cases들의 평균을 최대화하려는 것과 같다. 두 번째, 네 번째 그리고 다섯 번째 convolutional layers들의 커널들은 같은 GPU에 존재하는 이전 층의 커널 맵에 오직 연결된다. 세 번째 convolutional 층의 이전 층의 뉴런에 모두 연결된다. Fully-connected layers들의 뉴런은 이전 층의 모든 뉴런에 연결된다. Response-normalization layers는 처음과 두 번째의 convolutional layers를 따른다. Max-pooling layers는 response-normalization 층 모두와 다섯 번째 convolutional layer를 따른다. ReLU 비선형 함수는 모든 convolutional layer와 fully-connected layer의 output에 적용된다.

처음 convolutional 층은 224x225x3의 입력 이미지를 11x11x3 사이즈의 stride가 4인 픽셀 96개로 스캔한다. 두 번째 convolutional 층은 첫 번째 convolutional 층과 256개 커널들의 output을 input으로 한다. 세 번째, 네 번째와 다섯 번째는 pooling layer이나 normalization layer 없이 서로 연결된다. 세 번째 convolutional layer는 3x3x256 사이즈의 커널 384개를 갖는데 이는 정규화되고 pool된 두 번째 층의 output과 연결된다. 네 번째 convolutional layer는 3x3x192 사이즈의 커널 256개를 갖고 fully-connected layer들은 4096개의 뉴런을 각각 갖는다.

4. Reducing Overfitting

신경망 구조는 6000만 개의 파라미터가 있다. ILSVRC의 1000개의 클래스가 각 훈련 샘플들에게 이미지에서 라벨로 mapping하는 데 10 bits 제약조건을 가하지만 과적합을 당연히 고려를 안 할 수 없다. 과적합을 방지하기 위한 방법 두 가지를 말해본다.

4.1 Data Augmentation

과적합을 피하기 위한 가장 쉽고 흔한 방법은 인위적으로 label-preserving 변환을 통해 데이터셋을 여러 개 갖는 것이다. 원래의 그림에서 변형된 이미지를 생성하는 두 가지 방법을 사용했는데 이렇게 변환된 이미지는 디스크에 저장될 필요가 없다. 이 논문에서는 변환된 사진은 파이썬 코드를 통해 CPU에서 생성되고 그 와중에 GPU는 그전 이미지 배치들을 학습 중이다. 따라서 이러한 이미지 증강 계획은 계산적으로 free하다.

첫 번째 방법은 image translation을 생성하고 horizontal reflections를 포함한다. 기존의 256x256 이미지에서 랜덤한 224x224 패치들을 추출하여 추출된 패치들에 대해 네트워크를 학습시킨다. 학

습용 데이터를 비록 서로 연관이 있지만 크기를 많이 증가시킨다. 이런 계획이 없다면 상당한 과적합으로 인해 힘들 것이다. 그리고 이것은 더 작은 네트워크를 사용하도록 한다. 테스트 시간에는, 네트워크는 5개(코너 4개, 센터 1개)의 패치들을 224x224뿐만 아니라 horizontal reflections로부터 추출하여 10개를 뽑은 후 이것에 대해 소프트 맥스 층에서 만들어진 예측 값들을 평균 낸다.

두 번째 방법은 훈련용 이미지에서 RGB 채널의 강도를 바꾸는 것이다. 특히 RGB 픽셀에 대해 PCA를 수행한다. 각각의 훈련용 이미지에 발견된 주요 속성들을 더한다. 이러한 계획은 근사하게 원본 이미지의 주요한 특성들을 포착한다. 즉 물체 인식은 intensiry의 변화나 illumination의 색깔 변화에 변하지 않는다는 것을 의미한다.

4.2 Dropout

여러 모델로부터 예측치들을 조합하는 것은 오차를 줄이는 좋은 방법이다. 하지만 훈련하는 데 수일이 걸리는 큰 신경망의 경우에는 너무 비용이 크다. 하지만 드롭아웃이라는 비용도 크게 안 들고 효과적인 방법이 있다. 드롭아웃은 0.5의 확률로 몇몇 은닉 뉴런의 output을 0으로 만든다. 뉴런이 드롭아웃 되었다는 것은 forward pass하지 않고 역전파 과정에 참여하지 않는다는 것을 의미한다. 그래서 매번 input이 등장한다고 해도 신경망은 다른 구조를 추출한다. 하지만 이 모든 구조는 같은 가중치를 공유한다. 이러한 기법은 뉴런이 특정한 다른 뉴런들의 존재에 의존할 수 없기에 뉴런의 복잡한 co-adaptions을 줄여준다. 따라서 더 robust한 특성들을 학습하도록 강요되고 뉴런들의 다른 랜덤한 일부들과의 인접에 유용하다. Test 시에는, 뉴런의 output에 -0.5를 곱한 후 사용하는데 예측 분포들의 기하적인 평균으로 근사하는 타당한 수치이다.

5. Details of learning

128의 배치 사이즈, 0.9의 momentum, 0.005의 weight decay와 함께 확률적 경사하강법을 사용했다. 작은 크기의 weight decay가 학습에 중요하고 단순히 정규화하기 위한 용도가 아님을 의미한다. Training error를 감소시킨다. 업데이트 되는 방식은 다음과 같다.

$$
\begin{aligned}
v_{i+1} &:= 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i} \\
w_{i+1} &:= w_i + v_{i+1}
\end{aligned}
$$

I는 반복 inex, v는 momentum 변수, epsilon은 학습율, 미분식은 i번째 배치에 대한 w에 관한 미분식의 평균

평균은 0이고 표준편차는 0.01인 가우시안 분포를 통해 각 층의 가중치를 초기화했다. 2번째, 4번째, 다섯 번째 convolutional layers뿐만 아니라 fully-connected 은닉층의 절편은 1로 초기화한다.

이러한 초기화는 ReLU에 양의 값을 줌으로써 초반에 학습 속도를 가속하게 한다. 남아있는 층에 대해서는 절편을 0으로 초기화한다. 모든 층에 대해서 같은 학습율을 사용할 것이고 훈련하는 동안 알맞게 조정될 것이다. 0.01로 초기화를 했고 종료 이전에 3배 감소한다. 훈련을 거의 90순환 하였고 NVIDIA GTX 580 3GB GPUs에서 5일에서 6일 정도 걸렸다.

6. Results

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

6.1 Qualitative Evaluations

Figure 3는 네트워크의 두 개의 데이터가 연결된 layer에 의해 학습되는 convolutional 커널들을 보여준다. 네트워크는 frequency-kernels 와 orientation-selective kernels의 다양성을 학습한다. 두 개의 GPU에 의해 보여지는 전문성이 있다. GPU 1에 있는 커널들은 크게 color-agnostic이고 GPU 2에 있는 커널들은 color-specific이다. 이러한 전문성은 매 실행마다 발생하며 어떠한 랜덤 가중치 초기화와 무관하다.

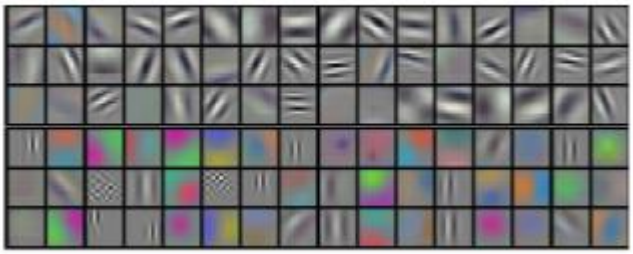

Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

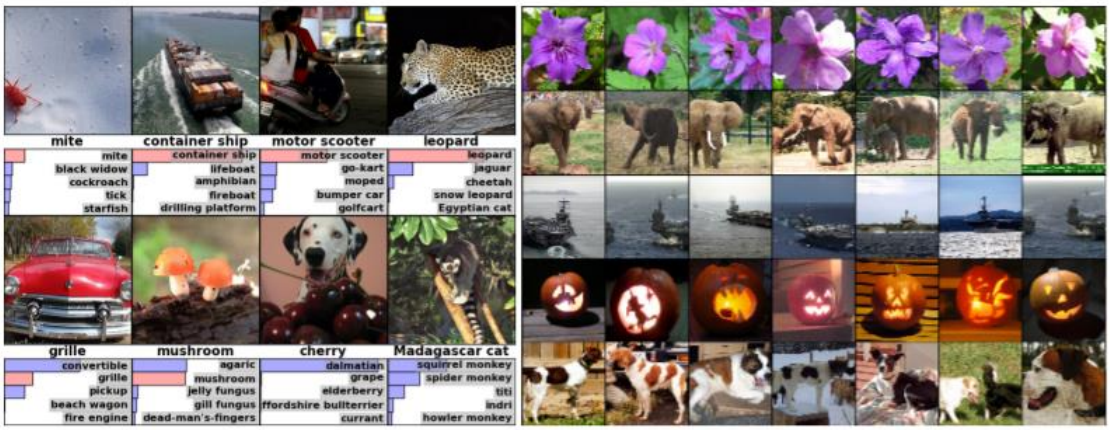

Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

7. Discussion

결과들은 크고 깊은 CNN이 순수한 지도 학습을 사용해 상당히 도저전적인 데이터셋에 기록될만한 기록을 남겼다. 만일 한 층의 합성곱 층이 제거되었다면 성능은 저하됐을 것이다.

단순히 말하면 비지도 학습으로 미리 훈련된 모형이 도움이 될 것이라고 생각했지만 사용하지 않았다. 후에 video sequences에 도전해보고 싶다고 한다.

url : https://s3.us-west-2.amazonaws.com/secure.notion-static.com/18ad7bd1-c2e3-46b5-8cb2-ee805ea18564/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Credential=AKIAT73L2G45EIPT3X45%2F20220120%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20220120T072704Z&X-Amz-Expires=86400&X-Amz-Signature=adffacd54ce20aedb4121b330c508b09fd0ce416e7cb4178801c288ae73030a0&X-Amz-SignedHeaders=host&response-content-disposition=filename%20%3D%224824-imagenet-classification-with-deep-convolutional-neural-networks.pdf%22&x-id=GetObject