AlexNet

핵심 요약

- 기존 머신러닝 모델을 제치고 딥러닝 모델이 더 우수한 성능을 보일 수 있음을 증명한 최초의 모델입니다.
- ReLU 활성화 함수와 Dropout 의 유용함, Data Augmentation 기법을 제시하였습니다.
- 2012년 ImageNet 대회 ILSVRC 에서 우승을 차지한 모델입니다.

Introduction

AlexNet 이전의 객체 인식 모델은 대부분 고전적인 ML 모델로, 수만개 정도의 작은 데이터셋(NORB, Caltech-101/256, CIFAR-10/100)을 사용했습니다.

그러나 이후, 수십만 개의 완전 분할 된 이미지로 구성된 LabelMe 와 1500 만 개 이상의 고해상도 이미지로 구성된 ImageNet 이 등장합니다.

간단한 인식 문제는 augmentation으로 해결가능 ex) MNIST error rate <0.3% <human performance, 반면 현실적인 물체는 상당한 가변성을 가지고 있어서 dataset이 더 많이 필요.

이런 데이터셋을 처리하기 위해서는 높은 학습 역량을 가진 모델이 필요합니다. 또한, 학습 과정에 사용되지 않은 수많은 데이터에 대해서도 추론을 할 수 있는 방대한 사전 지식을 담아내야합니다. (GPU + ImageNet의 large dataset의 등장)

이에 논문은 컨볼루션 신경망(CNN) 모델을 기반으로 하는 AlexNet 을 제시합니다.

CNN 은 FFNN(feed-forward NN)에 비해 더 적은 매개 변수를 가지므로 훈련이 용이합니다. 이를 통해 ILSVRC-2010, ILSVRC-2012 대회에 사용된 ImageNet subset에서 최고의 성능을 달성했습니다.

또한, 네트워크 성능 향상과 훈련시간 감소를 위한 여러가지 방법을 제시합니다.

ALexNet 은 2개의 GTX 580 3GB GPU에서 5-6 일동안 훈련을 수행하였습니다.

Top 1 error : prediction 상위 class 1개를 보고 정답을 맞추면 0%

Top 5 error : prediction 상위 class 5개를 보고 정답을 맞추면 0%

The Dataset

지금은 대부분의 딥러닝 모델에서 기본적으로 사용하는 ImageNet 에 대한 소개입니다.

- 22,000 개 범주로 구분되는 1,500 만개 고해상도 이미지
- 웹에서 수집한 이미지를 Amazon 의 Mechanical Turk 크라우드 소싱 도구로 라벨링

2010 년부터 Pascal Visual Object Challenge의 일환으로 ImageNet 대규모 시각 인식 도전 (ILSVRC = ImageNet Large-Scale Visual Recognition Challenge)이라는 연례 대회가 열렸습니다.

ILSVRC는 1000 개의 카테고리 각각에 약 1000 개의 이미지가있는 ImageNet의 하위 집합을 사용합니다. 이는 약 120 만 개의 훈련 이미지, 50,000 개의 검증 이미지, 150,000 개의 테스트 이미지로 구성됩니다.

대부분의 실험 결과는 테스트 이미지가 공개된 ILSVRC-2010 를 사용합니다. 별도로, AlexNet 이 참가했던 ILSVRC-2012 실험 결과 또한 제시합니다.

ImageNet 데이터셋 성능 지표로는 Top-1/Top-5 Accuracy 를 사용합니다.

가변 해상도로 구성된 ImageNet 을 처리하기 위해 256 × 256의 고정 해상도로 다운 샘플링을 수행합니다. 직사각형 이미지는 scaling 후 중앙 256x256 패치를 잘라냅니다. 이외의 전처리는 수행하지 않습니다.

The Architecture

ReLU Nonlinearity

논문 발표 당시 일반적으로 사용된 perceptron 의 activation 함수는 tanh 혹은 sigmoid 입니다. 이들은 출력값이 무한대로 발산하지 않고 특정한 영역으로 제한되는 saturating 함수입니다. 반면 ReLU(Recitified Linear Unit) activation 은 출력값이 0 에서 무한대까지 발산할 수 있는 non-saturating 함수입니다.

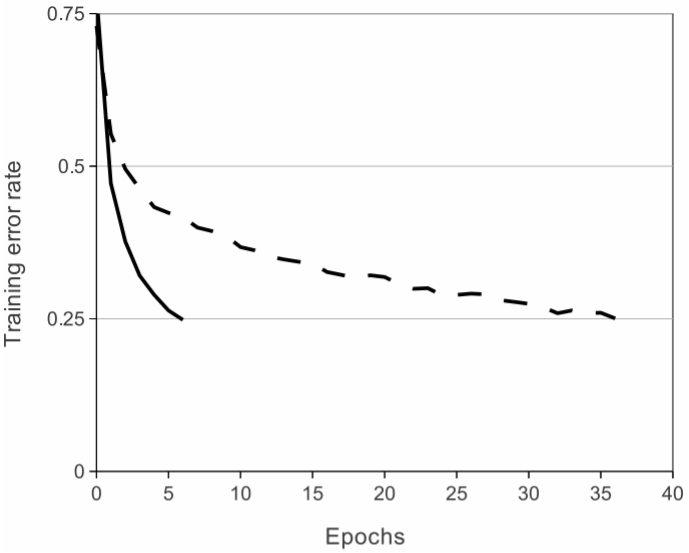


Figure 1: A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**. The learning rates for each network were chosen independently to make training as fast as possible. 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.

논문은 4 layer CNN 으로 CIFAR-10 데이터셋을 사용하여 학습하였을 때, ReLU 가 6배 빠른 학습 속도를 보여주었음을 제시합니다. 이를 통해, non-saturating 한 함수가 gradient 를 더 빠르게 update 할 수 있음을 제시합니다.

Saturating function : x가 무한대로 갈 때 함수 값이 어떤 범위 내에 존재하는 함수.

Training on Multiple GPUs

GPU 메모리 제한과 느린 학습 속도를 개선할 수 있는 병렬학습 방법을 제시합니다.

기본 골자는 네트워크를 분할(커널, 뉴런 등)하여 서로 다른 GPU 에서 병렬적으로 연산을 수행하는 것입니다. 이 때, 메모리의 한계 및 병목 현상을 고려하여, 특정한 레이어에서만 연산 결과를 교환합니다.

논문은 이를 통해 half-size kernel 을 사용한 단일 GPU 모델보다 Top-1/Top-5 accuracy 를 1.7% / 1.2% 감소시켰음을 제시합니다.

Local Response Normalization

ReLU 활성 함수는 입력을 normalization 하지 않아도 saturation 이 발생하지 않습니다. 그러나 positive value 를 그대로 출력하는 ReLU 함수의 특성으로 인해 CNN 의 일부 구역에서 강한 신호가 생성될 수 있습니다. 이에 논문은 아래와 같은 local response normalization 방법을 제시합니다.

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

Activation value를 억제, How? i번째 kernel의 특정 kernel의 앞에서부터 n개를 본다.

요약하면, CNN 에서 인접한 필터를 사용하여 normalization 을 진행한 것으로, 논문에서는 Top-1/Top-5 Accuracy 를 1.4%, 1.2% 개선할 수 있었음을 제시합니다. 또한, CIFAR-10 으로 학습을 수행하였을 때도 2% 의 오차율 감소를 보였음을 제시합니다.
(논문 당시에는 Batch Normalization 이 소개되지 않았습니다.)

5. Local Response Normalization

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

$k = 2, n = 5, \alpha = 10^{-4}$, and $\beta = 0.75$.

K = bias

i = i번째 kernel

ai(x,y) : i번째 kernel의 (x,y) feature의 activation 값

n : 내가 몇 개의 kernel을 참고할 것인가

N : 실제 총 kernel 개수

Positive value를 그대로 출력하는 relu의 특성으로 인해, 한 노드의 값이 커질 수 있다는 문제가 생길 수 있다. => activation value 억제

I 번째 activation 값이 주변 kernel의 activation값에 의해 규제된다.

Overlapping Pooling

CNN의 풀링 레이어는 같은 채널에 존재하는 인접한 뉴런의 출력을 요약해줍니다. 논문 이전에는 pooling 을 수행하는 영역이 겹치지 않도록 구성하여 사용하는 것이 일반적이었습니다.

논문은 풀링을 수행하는 영역이 이동하는 거리를 조절하여 풀링 영역이 겹치도록 한 결과, Top-1/Top-5 Accuracy 를 0.4 %/0.3 % 감소했다고 합니다. 또한 이를 통해 모델의 과적합 가능성을 줄일 수 있었다고 합니다.

Overall Architecture

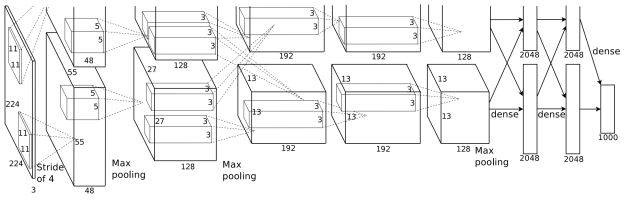
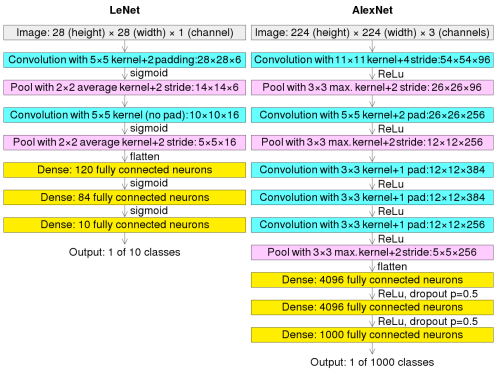


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input 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.



AlexNet 의 전체 구조도 입니다. 2GPU 로 병렬학습을 수행하기 위해 두 갈래로 나뉘어 표현되어 있습니다. 총 5개의 convolution layer 와 3개의 max pooling layer, 3개의 dense layer 로 구성되어 있으며, 필요한 경우에만 GPU 연산 결과를 공유합니다. 또한 convolution/dense layer 의 활성함수는 ReLU 를 사용합니다.

- Input : 224 x 224 x 3 = 150,528
- Convolution 1 : 11x11 kernel, 4 stride : 54x54x96
- Max pooling 1 : 3x3 kernel, 2 stride : 26x26x96
- Convolution 2 : 5x5 kernel, 2 padding : 26x26x256
- Max pooling 2 : 3x3 kernel, 2 stride : 12x12x256
- Convolution 3 : 3x3 kernel, 1 padding : 12x12x384
- Convolution 4 : 3x3 kernel, 1 padding : 12x12x384
- Convolution 5 : 3x3 kernel, 1 padding : 12x12x384
- Max pooling 3 : 3x3 kernel, 2 stride : 5x5x256
- Dense 1 : 4096
- Dense 2 : 4096

- Dense 3 : 1000

Reducing Overfitting

6천만개의 파라미터로 구성된 모델의 과적합을 막기 위해 사용한 방법을 소개합니다.

Data Augmentation

학습 데이터를 인위적으로 변환하여 훈련 데이터를 증가시키는 방법입니다. 변환된 이미지를 저장하지 않고 GPU 학습시에 CPU에서 계산하도록 하여, 추가적인 계산 비용을 소모하지 않았다고 합니다. 주요 방법은 두가지로 요약됩니다.

- 256 × 256 이미지에서 224 × 224 패치를 추출하고, 수평 방향으로 뒤집기
  - 기존 데이터 셋의 2048 배 확장 가능 (256 - 224)*(256 - 224) * 2 = 2024
  - 실제 : 5 개의 224 × 224 패치 (4 개의 코너 패치 및 중앙 패치)와 수평 반사를 수행한 10개의 패치 사용 (10배?)
- RGB 채널 강도 조정
  - 학습 데이터셋의 픽셀값으로 PCA 를 수행
  - PCA eigenvector 에 N(0,0.1) 인 정규분포에 추출한 랜덤값을 곱해 색상을 조정
  - Top-1 오차율을 1% 감소할 수 있었음

7. Data augmentation

256 x 256 이미지를 227 x 227로 patch 추출 : kernel 처럼 움직여서 뽑아낸다.

그럼, (256 – 227) * (256 – 227) 가지의 경우의 수가 나옴

+ 수평 방향으로 뒤집어서 (256 – 227) * (256 – 227) *2

실제로는, 기존 data보다 10배 늘려서 학습 (1200만장)

Dropout

Dense Layer 의 Output 에 Dropout rate 0.5 를 사용한 Dropout layer 를 추가합니다. 학습에 필요한 Epoch 를 2배 정도 늘렸으나, 과적합을 성공적으로 방지했음을 제시합니다.

Details of learning

모델 학습의 세부내용입니다.

- Batch size : 128
- SGD (momentum 0.9, weight decay 0.0005)

weight decay 가 모델을 정규화 할 뿐만 아니라 직접적으로 모델의 학습 오차를 줄였음을 제시합니다. 가중치 업데이트 과정은 아래와 같습니다.

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

$$w_{i+1} := w_i + v_{i+1}$$

the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

where i is the iteration index, v is the momentum variable, ϵ is the learning rate, and $\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$ is the average over the ith batch D_i of the derivative of the objective with respect to w, evaluated at w_i.

가중치는 평균이 0, 표준 편차가 0.01인 정규 분포를 따르도록 초기화합니다. 2/4/5 번째 convolution 과 dense layer의 bias 는 1로 초기화하여, 학습을 가속할 수 있었음을 제시합니다.

학습률은 모든 layer 에 대해서 동일하되, 훈련을 수행하면서 메뉴얼하게 조정합니다. 학습률 0.01 에서 시작하여, 학습이 개선되지 않을 때 학습률을 10으로 나누는 방식으로 수행합니다.

RESULT

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.

ILSVRC-2010 데이터에 대해서 기존 모델의 결과를 압도적으로 상회하는 결과를 제시하였습니다.

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.

AlexNet 이 직접 참가했던 ILSVRC-2012 에서도 다른 최고 성능의 모델에 비해 압도적인 결과를 보였음을 확인할 수 있습니다. 또한, CNN Layer 갯수를 추가할 때마다 성능이 상승함을 제시합니다.

Qualitative Evaluations

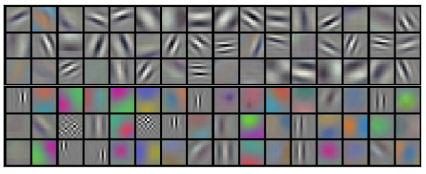


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

CNN kernel 을 시각화한 그림을 제시하면서, 각 커널이 이미지의 다양한 Feature 를 효과적으로 추출해냈음을 제시합니다.

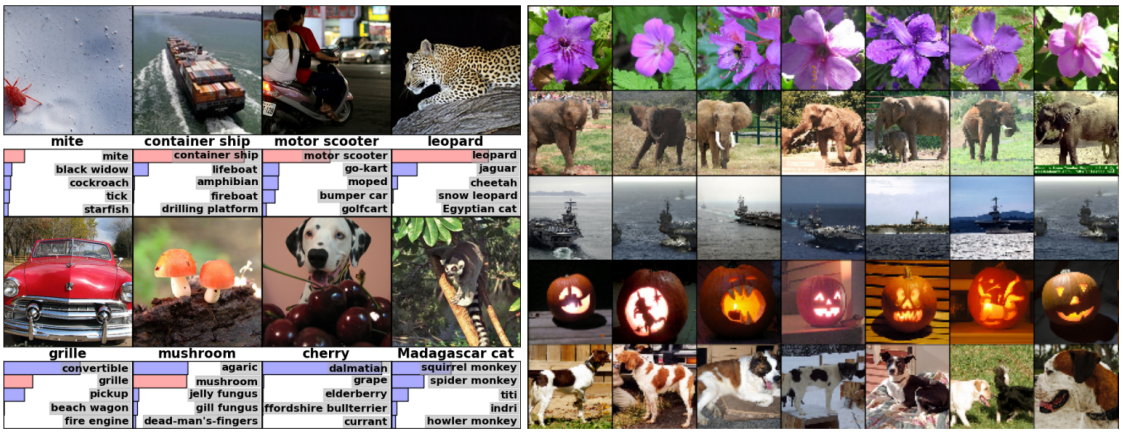


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 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.

AlexNet 은 중앙을 벗어나는 데이터도 효과적으로 분류해냈습니다. 또한, Top-5 예측이 대부분 유사한 범주인 것으로 보아 합리적인 예측을 수행하고 있음을 제시합니다.

또한, 자세가 서로 다른 코끼리의 사례와 같이, Pixel 차원에서 완전히 다른 데이터임에도 유사한 범주로 분류할 수 있는 결과를 보여줍니다.

Discussion

"깊은" CNN 이 효과적으로 작동하였음을 제시합니다. Convolution layer 를 제거할 때마다 Top-1 Accuracy 가 2%씩 감소하는 점에 미루어, "깊이"의 중요성을 다시 한번 제시합니다.