심층 컨볼루션 신경망을 사용한 이미지넷 분류

알렉스 크리제프스키 토론토 대학
교 kriz@cs.utoronto.ca

일리야 수츠케버
토론토 대학교
ilya@cs.utoronto.ca

제프리 E. 힌튼 토론토 대학교
hinton@cs.utoronto.ca

Abstract

우리는 대규모 심층 컨볼루션 신경망을 훈련시켜 ImageNet LSVRC-2010 콘테스트에서 120만 개의 고해상도 이미지를 1000개의 서로 다른 클래스로 분류했습니다. 테스트 데이터에서 상위 1%와 상위 5%의 오류율은 37.5%와 17.0%로 기존 최첨단 기술보다 훨씬 개선된 결과를 얻었습니다. 6천만 개의 파라미터와 65만 개의 뉴런으로 구성된 신경망은 5개의 컨볼루션 레이어, 그 중 일부는 최대 풀링 레이어, 최종 1000-방향 소프트맥스로 완전히 연결된 3개의 레이어로 구성됩니다. 훈련 속도를 높이기 위해 비포화 뉴런과 컨볼루션 연산의 매우 효율적인 GPU 구현을 사용했습니다. 완전히 연결된 레이어에서 과적합을 줄이기 위해 최근에 개발된 "드롭아웃"이라는 정규화 방법을 사용했는데, 매우 효과적인 것으로 입증되었습니다. 또한 이 모델의 변형을 ILSVRC-2012 대회에 출품하여 2위 출품작의 26.2%에 비해 15.3%의 상위 5위 테스트 오류율을 달성했습니다.

1 소개

현재 객체 인식에 대한 접근 방식은 머신러닝 방법을 필수적으로 사용합니다. 성능을 개선하기 위해 더 큰 데이터 세트를 수집하고, 더 강력한 모델을 학습하고, 과적합을 방지하기 위한 베타 기법을 사용할 수 있습니다. 최근까지 라벨이 지정된 이미지의 데이터 세트는 수만 장 정도의 비교적 작은 규모였습니다(예: NORB [16], Caltech-101/256 [8, 9], CIFAR-10/100 [12]). 간단한 인식 작업은 이 정도 크기의 데이터 세트, 특히 레이블 보존 변환으로 보강된 경우 꽤 잘 해결할 수 있습니다. 예를 들어, MNIST 숫자 인식 작업의 현재 최고 오류율(<0.3%)은 인간의 성능에 근접합니다[4]. 그러나 실제 환경의 사물은 상당한 변동성을 보이므로 이를 인식하는 방법을 학습하려면 훨씬 더 큰 훈련 세트를 사용해야 합니다. 실제로 작은 이미지 데이터 세트의 단점은 널리 알려져 왔지만(예: Pinto 외. [21]), 수백만 개의 이미지로 라벨링된 데이터 세트를 수집하는 것은 최근에야 가능해졌습니다. 새로운 대규모 데이터 세트에는 수십만 개의 완전히 분할된 이미지로 구성된 LabelMe [23]와 22,000개 이상의 카테고리에서 1,500만 개 이상의 라벨이 지정된 고해상도 이미지로 구성된 ImageNet [6]이 있습니다.

수백만 개의 이미지에서 수천 개의 객체에 대해 학습하려면 학습 용량이 큰 모델이 필요합니다. 그러나 객체 인식 작업의 엄청난 복잡성으로 인해 ImageNet만큼 큰 데이터 세트에서도 이 확률 렘을 지정할 수 없으므로 모델에는 우리가 가지고 있지 않은 모든 데이터를 보완할 수 있는 많은 사전 지식이 있어야 합니다. 컨볼루션 신경망(CNN)은 이러한 종류의 모델 중 하나입니다[16, 11, 13, 18, 15, 22, 26]. 이 모델은 깊이와 폭을 변화시켜 용량을 조절할 수 있으며, 이미지의 특성(통계의 고정성 및 픽셀 의존성의 지역성)에 대해 강력하고 대부분 정확한 가정을 내립니다. 따라서 비슷한 크기의 레이어를 가진 표준 피드포워드 신경망에 비해 CNN은 연결과 매개변수가 훨씬 적기 때문에 훈련하기가 더 쉬운 반면, 이론적으로 가장 좋은 성능은 약간 떨어질 가능성이 높습니다.

CNN의 매력적인 특성과 로컬 아키텍처의 상대적인 효율성에도 불구하고, 고해상도 이미지에 대규모로 적용하기에는 여전히 비용이 엄청나게 비쌉니다. 다행히도 현재 GPU는 고도로 최적화된 2D 컨볼루션 구현과 결합되어 매우 큰 규모의 CNN을 쉽게 훈련할 수 있을 만큼 강력하며, ImageNet과 같은 최신 데이터 세트에는 심각한 과적합 없이 이러한 모델을 훈련하기에 충분한 라벨링된 예시가 포함되어 있습니다.

이 논문의 구체적인 기여는 다음과 같습니다: 우리는 ILSVRC-2010 및 ILSVRC-2012 대회[2]에서 사용된 ImageNet의 하위 집합에 대해 지금까지 가장 큰 규모의 컨볼루션 신경망을 훈련했으며, 이러한 데이터 세트에서 지금까지 보고된 것 중 최고의 결과를 달성했습니다. 우리는 2D 컨볼루션과 컨볼루션 신경망 훈련에 내재된 다른 모든 연산에 대해 고도로 최적화된 GPU 구현을 작성했으며, 이를 공개적으로 제공합니다[1]. 저희 네트워크에는 성능을 향상시키고 훈련 시간을 단축하는 새롭고 특이한 기능들이 많이 포함되어 있으며, 섹션 3에 자세히 설명되어 있습니다. 네트워크의 크기로 인해 120만 개의 라벨이 지정된 훈련 예시에서도 과적합이 심각한 문제가 되었기 때문에 과적합을 방지하기 위해 섹션 4에 설명된 몇 가지 효과적인 기법을 사용했습니다. 최종 네트워크에는 5개의 컨볼루션 레이어와 3개의 완전 연결 레이어가 포함되어 있으며, 이 깊이가 중요한 것으로 나타났습니다. 컨볼루션 레이어(각각 모델 파라미터의 1% 이하를 포함)를 제거하면 성능이 저하되는 것으로 나타났습니다.

결국 네트워크의 크기는 주로 현재 GPU에서 사용할 수 있는 메모리 양과 우리가 감내할 수 있는 훈련 시간에 의해 제한됩니다. 저희 네트워크는 두 개의 GTX 580 3GB GPU로 훈련하는 데 5~6일이 걸립니다. 모든 실험을 통해 더 빠른 GPU와 더 큰 데이터 세트가 제공될 때까지 기다리면 결과를 개선할 수 있음을 알 수 있습니다.

2 데이터 세트

ImageNet은 약 22,000개의 카테고리에 속하는 1,500만 개 이상의 라벨이 붙은 고해상도 이미지로 구성된 데이터 세트입니다. 이 이미지들은 웹에서 수집된 것으로, 아마존의 머신 러닝 크라우드 소싱 도구인 Mechanical Turk를 사용해 라벨러들이 라벨을 붙였습니다. 2010년부터 파스칼 비주얼 오브젝트 챌린지의 일환으로 이미지넷 대규모 시각 인식 챌린지(ILSVRC)라는 연례 대회가 열리고 있습니다. ILSVRC는 1000개의 카테고리 각각에 약 1000개의 이미지가 포함된 ImageNet의 하위 집합을 사용합니다. 총 120만 개의 훈련 이미지, 5만 개의 검증 이미지, 15만 개의 테스트 이미지가 사용됩니다.

ILSVRC-2010은 테스트 세트 레이블을 사용할 수 있는 유일한 ILSVRC 버전이므로 대부분의 실험을 이 버전에서 수행했습니다. ILSVRC-2012 대회에도 모델을 출품했기 때문에 섹션 6에서는 테스트 세트 레이블을 사용할 수 없는 이 버전의 데이터 세트에 대한 결과도 보고합니다. ImageNet에서는 상위 1%와 상위 5%의 두 가지 오류율을 보고하는 것이 일반적이며, 여기서 상위 5% 오류율은 모델에서 가장 가능성이 높은 것으로 간주되는 5개의 라벨 중 올바른 라벨이 없는 테스트 이미지의 비율을 의미합니다.

이미지넷은 가변 해상도 이미지로 구성되어 있지만, 저희 시스템은 일정한 입력 크기를 필요로 합니다. 따라서 이미지를 고정 해상도인 256× 256으로 다운샘플링했습니다. 직사각형 이미지가 주어지면 먼저 짧은 쪽의 길이가 256이 되도록 이미지의 크기를 조정한 다음, 결과 이미지에서 중앙의 256× 256 패치를 잘라냈습니다. 각 픽셀에서 학습 세트의 평균 활동량을 뺀 것 외에는 다른 방식으로 이미지를 사전 처리하지 않았습니다. 따라서 픽셀의 (중앙에 있는) 원시 RGB 값으로 네트워크를 훈련시켰습니다.

3 아키텍처

저희 네트워크의 아키텍처는 그림 2에 요약되어 있습니다. 여기에는 5개의 컨볼루션 레이어와 3개의 완전 연결 레이어 등 총 8개의 학습 레이어가 포함되어 있습니다. 아래에서는 네트워크 아키텍처의 새롭거나 특이한 몇 가지 특징에 대해 설명합니다. 섹션 3.1~3.4는 중요도 평가에 따라 가장 중요한 것부터 순서대로 정렬되어 있습니다.

[1] http://code.google.com/p/cuda-convnet/

3.1 ReLU 비선형성

뉴런의 출력 f를 입력 x의 함수로 모델링하는 표준 방법은 $f(x) = \tanh(x)$ 또는 $f(x) = (1+e^{-x})^{-1}$을 사용하는 것입니다. 경사 하강을 통한 훈련 시간 측면에서 이러한 포화 비선형성은 비포화 비선형성 $f(x) = \max(0, x)$보다 훨씬 느립니다. Nair와 Hinton[20]에 따라 이러한 비선형성을 가진 뉴런을 정류된 선형 단위(ReLU)라고 부릅니다. ReLU를 사용하는 심층 컨볼루션 신경망은 탄 단위의 동급 신경망보다 몇 배 더 빠르게 훈련합니다. 이는 특정 4계층 컨볼루션 네트워크에 대해 CIFAR-10 데이터 세트에서 25%의 훈련 오류에 도달하는 데 필요한 반복 횟수를 보여주는 그림 1에서 확인할 수 있습니다. 이 도표는 기존의 포화 뉴런 모델을 사용했다면 이 작업에 이렇게 큰 규모의 신경망을 실험할 수 없었을 것임을 보여줍니다.

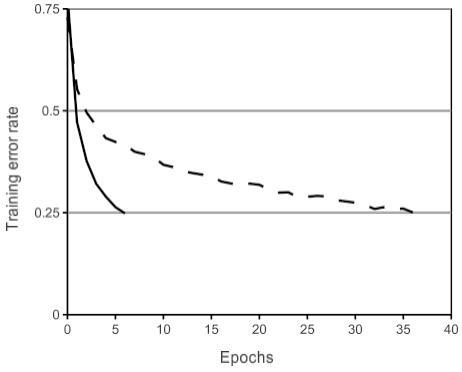

그림 1: ReLU(**실선**)를 사용하는 4계층 컨볼루션 신경망은 탄 뉴런(**점선**)을 사용하는 동급 네트워크보다 6배 더 빠르게 CIFAR-10에서 25%의 훈련 오류율에 도달합니다. 각 네트워크의 학습 속도는 가능한 한 빠르게 훈련하기 위해 독립적으로 선택되었습니다. 어떤 종류의 정규화도 사용하지 않았습니다. 여기서 보여주는 효과의 크기는 네트워크 아키텍처에 따라 다르지만, ReLU를 사용하는 네트워크는 포화 뉴런을 사용하는 동급 네트워크보다 몇 배 더 빠르게 학습합니다.

CNN에서 합성곱 뉴런 모델의 대안을 고려한 것은 우리가 처음이 아닙니다. 예를 들어, Jarrett 등[11]은 비선형성 $f(x) = |\tanh(x)|$이 Caltech-101 데이터 세트의 국부 평균 풀링에 따른 대비 노멀화 유형에서 특히 잘 작동한다고 주장합니다. 그러나 이 데이터 세트에서는 과적합을 방지하는 것이 가장 중요한 관심사이므로, 관찰한 효과는 ReLU를 사용할 때 보고하는 훈련 세트의 가속화된 적합 능력과는 다릅니다. 빠른 학습은 대규모 데이터 세트에서 학습된 대규모 모델의 성능에 큰 영향을 미칩니다.

3.2 여러 GPU에서의 훈련

단일 GTX 580 GPU에는 3GB의 메모리만 있기 때문에 학습할 수 있는 네트워크의 최대 크기가 제한됩니다. 120만 개의 훈련 예제만으로도 하나의 GPU에 담기에는 너무 큰 네트워크를 훈련하기에 충분하다는 것이 밝혀졌습니다. 따라서 두 개의 GPU에 네트워크를 분산시켰습니다. 현재 GPU는 호스트 머신 메모리를 거치지 않고 서로의 메모리를 직접 읽고 쓸 수 있기 때문에 크로스 GPU 병렬화에 특히 적합합니다. 우리가 사용하는 병렬화 방식은 기본적으로 각 GPU에 커널(또는 뉴런)의 절반을 배치하며, 한 가지 추가 트릭이 있는데, 바로 GPU가 특정 계층에서만 통신한다는 것입니다. 예를 들어, 레이어 3의 커널은 레이어 2의 모든 커널 맵에서 입력을 받습니다. 그러나 레이어 4의 커널은 동일한 GPU에 있는 레이어 3의 커널 맵으로부터만 입력을 받습니다. 연결 패턴을 선택하는 것은 교차 검증의 문제이지만, 이를 통해 계산량의 허용 가능한 일부가 될 때까지 통신량을 정밀하게 조정할 수 있습니다.

결과적인 아키텍처는 컬럼이 독립적이지 않다는 점을 제외하면 Cires¸an 외[5]가 사용한 "컬럼형" CNN과 다소 유사합니다 (그림 2 참조). 이 방식은 하나의 GPU에서 훈련된 각 컨볼루션 계층의 커널 수가 절반인 네트워크와 비교했을 때 상위 1%와 상위 5%의 오류율을 각각 1.7%와 1.2% 감소시킵니다. 2-GPU 넷은 1-GPU 넷보다 훈련에 걸리는 시간이 약간 더 짧습니다[2].

[2]1-GPU 넷은 실제로 최종 컨볼루션 레이어에서 2-GPU 넷과 동일한 수의 커널을 갖습니다. 이는 넷의 파라미터 대부분이 마지막 컨볼루션 레이어를 입력으로 사용하는 첫 번째 완전 연결 레이어에 있기 때문입니다. 따라서 두 네트워크의 파라미터 수가 거의 같도록 하기 위해 최종 컨볼루션 레이어의 크기를 절반으로 줄이지 않았습니다(그 뒤에 있는 완전 연결 레이어도 마찬가지입니다). 따라서 이 비교는 1-GPU 넷이 2-GPU 넷의 "절반 크기"보다 더 크기 때문에 1-GPU 넷에 유리하게 편향되어 있습니다.

3.3 로컬 응답 정규화

ReLU는 포화 상태를 방지하기 위해 입력 정규화가 필요하지 않다는 바람직한 특성을 가지고 있습니다. 적어도 일부 훈련 예시에서 ReLU에 양의 입력을 생성하면 해당 뉴런에서 학습이 이루어집니다. 그러나 다음과 같은 로컬 정규화 방식이 여전히 도움이 된다는 사실을 발견했습니다.

일반화. $a^i_{x,y}$ 로 나타내는 뉴런의 활동은 커널 i를 위치 (x, y)에 커널 i를 적용한 다음 ReLU 비선형성을 적용하여 계산된 뉴런의 활성도를 나타내는 반응 정규화 활성도 $b^i_{x,y}$ 는 다음과 같이 주어집니다.

로 표현됩니다.

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

여기서 합은 동일한 공간 위치에서 "인접한" 커널 맵 n개에 걸쳐 실행되며, N은 레이어에 있는 커널의 총 개수입니다. 물론 커널 맵의 순서는 훈련이 시작되기 전에 임의로 결정됩니다. 이러한 종류의 반응 정규화는 실제 뉴런에서 발견되는 유형에서 영감을 얻은 일종의 측면 억제를 구현하여 서로 다른 커널을 사용하여 계산된 뉴런 출력 간에 큰 활동을 위한 경쟁을 유발합니다. 상수 k, n, α, β는 검증 세트를 사용하여 값이 결정되는 하이퍼파라미터로, $k = 2$, $n = 5$, $\alpha = 10^{-4}$, $\beta = 0.75$를 사용했습니다. 특정 레이어에 ReLU 비선형성을 적용한 후 이 정규화를 적용했습니다(섹션 3.5 참조).

이 방식은 Jarrett 등[11]의 국소 대비 정규화 방식과 어느 정도 유사하지만, 평균 활동을 빼지 않기 때문에 "밝기 정규화"라고 부르는 것이 더 정확할 것입니다. 응답 정규화는 상위 1%와 상위 5%의 오류율을 각각 1.4%와 1.2% 감소시켰습니다. 또한, 4계층 CNN은 정규화 없이 13%의 테스트 오류율과 정규화[3]를 통해 11%의 오류율을 달성했습니다.

3.4 중복 풀링

CNN의 풀링 레이어는 동일한 커널 맵에서 인접한 뉴런 그룹의 출력을 요약합니다. 전통적으로 인접한 풀링 유닛에 의해 요약된 이웃은 겹치지 않습니다(예: [17, 11, 4]). 더 정확하게 말하면, 풀링 레이어는 풀링 유닛의 위치를 중심으로 $z \times z$ 크기의 이웃을 요약하는 풀링 유닛의 그리드로 구성된 s 픽셀 간격으로 구성된 것으로 생각할 수 있습니다. $s = z$를 설정하면 CNN에서 일반적으로 사용되는 전통적인 로컬 풀링을 얻을 수 있습니다. $s < z$를 설정하면 중복 풀링을 얻게 됩니다. 이것이 바로 네트워크 전체에서 사용되는 방식이며, $s = 2$ 및 $z = 3$입니다. 이 방식은 동일한 차원의 출력을 생성하는 비중첩 방식인 $s = 2, z = 2$에 비해 상위 1%와 상위 5%의 오류율을 각각 0.4%와 0.3%씩 줄입니다. 일반적으로 훈련 과정에서 겹치는 풀링이 있는 모델은 과적합이 약간 더 어렵다는 것을 관찰할 수 있습니다.

3.5 전체 아키텍처

이제 CNN의 전체 아키텍처를 설명할 준비가 되었습니다. 그림 2에 표시된 것처럼 이 네트워크에는 가중치가 있는 8개의 레이어가 있으며, 처음 5개는 컨볼루션이고 나머지 3개는 완전히 연결된 레이어입니다. 마지막으로 완전히 연결된 계층의 출력은 1000개의 클래스 레이블에 대한 분포를 생성하는 1000방향 소프트맥스에 공급됩니다. 이 네트워크는 다항 로지스틱 회귀 목표를 최대화하는데, 이는 예측 분포에서 올바른 레이블의 로그 확률에 대한 훈련 사례의 평균을 최대화하는 것과 같습니다.

두 번째, 네 번째, 다섯 번째 컨볼루션 계층의 커널은 동일한 GPU에 있는 이전 계층의 커널 맵에만 연결됩니다(그림 2 참조). 세 번째 컨볼루션 계층의 커널은 두 번째 계층의 모든 커널 맵에 연결됩니다. 완전히 연결된 레이어의 뉴런은 이전 레이어의 모든 뉴런에 연결됩니다. 응답 정규화 레이어는 첫 번째와 두 번째 컨볼루션 레이어를 따릅니다. 3.4절에서 설명한 종류의 최대 풀링 레이어는 응답 정규화 레이어와 다섯 번째 컨볼루션 레이어를 모두 따릅니다. ReLU 비선형성은 모든 컨볼루션 및 완전히 연결된 레이어의 출력에 적용됩니다.

첫 번째 컨볼루션 레이어는 $224 \times 224 \times 3$ 입력 이미지를 $11 \times 11 \times 3$ 크기의 96개 커널로 4픽셀 간격으로 필터링합니다(이는 인접한 수신 필드 중심 사이의 거리입니다).

[3]공간 제약으로 인해 이 네트워크를 자세히 설명할 수는 없지만, 여기에 제공된 코드와 파라미터 파일(http://code.google.com/p/cuda-convnet/).에 정확하게 명시되어 있습니다

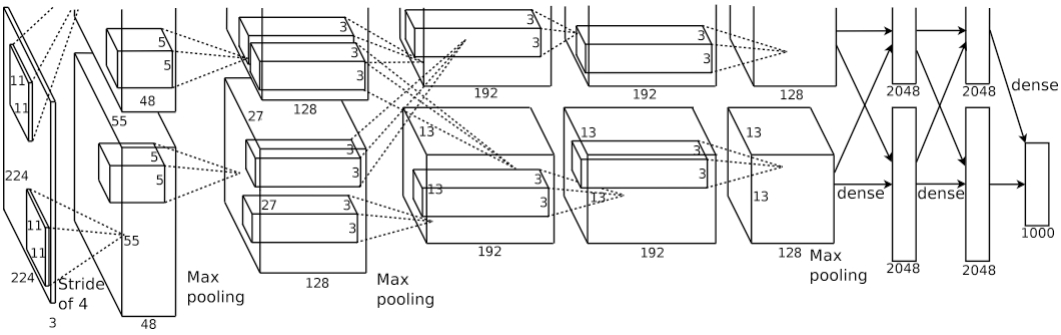

그림 2: 두 GPU 간의 책임 분담을 명시적으로 보여주는 CNN의 아키텍처 그림. 한 GPU는 그림 위쪽의 레이어 부분을 실행하고 다른 GPU는 아래쪽의 레이어 부분을 실행합니다. GPU는 특정 레이어에서만 통신합니다. 네트워크의 입력은 150,528 차원이며 네트워크의 나머지 레이어에 있는 뉴런 수는 253,440-186,624-64,896-64,896-43,264-4096-4096-1000으로 주어집니다.

커널 맵의 뉴런). 두 번째 컨볼루션 계층은 첫 번째 컨볼루션 계층의 (응답 정규화 및 풀링된) 출력을 입력으로 받아 5× 5× 48 크기의 256개의 커널로 필터링합니다. 세 번째, 네 번째, 다섯 번째 컨볼루션 레이어는 풀링 또는 정규화 레이어 없이 서로 연결됩니다. 세 번째 컨볼루션 레이어에는 3× 3× 256 크기의 384개의 커널이 두 번째 컨볼루션 레이어의 (정규화, 풀링된) 출력에 연결됩니다. 네 번째 컨볼루션 레이어에는 크기 3× 3× 192의 384개의 커널이 있고, 다섯 번째 컨볼루션 레이어에는 크기 3× 3× 192의 256개의 커널이 있습니다. 완전히 연결된 레이어에는 각각 4096개의 뉴런이 있습니다.

4 과적합 감소

저희의 신경망 아키텍처에는 6천만 개의 매개변수가 있습니다. ILSVRC의 1000개 클래스는 각 훈련 예제에서 이미지에서 레이블로 매핑하는 데 10비트의 제약 조건을 부과하지만, 이는 상당한 과적합 없이 많은 파라미터를 학습하기에 충분하지 않은 것으로 밝혀졌습니다. 아래에서는 과적합을 방지하는 두 가지 주요 방법을 설명합니다.

4.1 데이터 증강

이미지 데이터에서 과적합을 줄이는 가장 쉽고 가장 일반적인 방법은 라벨 보존 변환(예: [25, 4, 5])을 사용하여 데이터 세트를 인위적으로 확대하는 것입니다. 우리는 두 가지 다른 형태의 데이터 증강을 사용하는데, 두 가지 모두 아주 적은 계산으로 원본 이미지에서 변환된 이미지를 생성할 수 있으므로 변환된 이미지를 디스크에 저장할 필요가 없습니다. 저희의 구현에서는 변환된 이미지가 CPU에서 Python 코드로 생성되는 동안 GPU는 이전 이미지 배치에 대해 학습합니다. 따라서 이러한 데이터 증강 방식은 사실상 계산이 필요 없습니다.

데이터 증강의 첫 번째 형태는 이미지 번역과 수평 반사를 생성하는 것입니다. 256개의× 256개의 이미지에서 무작위로 224개의× 224개의 패치(및 수평 반사)를 추출하고 이 추출된 패치[4]에 대해 네트워크를 훈련하는 방식으로 수행합니다. 이렇게 하면 훈련 세트의 크기가 2048배로 증가하지만, 결과 훈련 예제는 물론 상호 의존성이 매우 높습니다. 이 방식이 없었다면 네트워크에 상당한 과적합이 발생하여 훨씬 더 작은 네트워크를 사용해야 했을 것입니다. 테스트 시 네트워크는 5개의 224개의× 224 패치(4개의 모서리 패치와 중앙 패치)와 수평 반사(따라서 총 10개의 패치)를 추출하고 네트워크의 소프트맥스 계층에서 10개의 패치에 대한 예측을 평균화하여 예측을 수행합니다.

두 번째 형태의 데이터 증강은 훈련 이미지에서 RGB 채널의 강도를 변경하는 것입니다. 구체적으로 이미지넷 훈련 세트 전체의 RGB 픽셀 값 집합에 대해 PCA를 수행합니다. 각 훈련 이미지에 발견된 주 성분의 배수를 추가합니다,

[4]이것이 그림 2의 입력 이미지가 224× 224× 3차원인 이유입니다.

해당 고유값에 비례하는 크기에 평균 0, 표준편차 0.1의 가우스에서 추출한 무작위 변수를 곱한 값으로 변환됩니다. 따라서 각 RGB 이미지 픽셀 I_{xy} = $[I^R_{xy}, I^G_{xy}, I^B_{xy}]^T$에 다음 수량을 더합니다:

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

여기서 $\mathbf{p}_i$및 λ_i는각각 RGB 픽셀 값의 3×3 공분산 행렬의 고유 벡터와 고유값이며 α_i는앞서 언급한 무작위 변수입니다. 각 α_i는특정 훈련 이미지의 모든 픽셀에 대해 한 번만 그려지며, 해당 이미지가 다시 훈련에 사용될 때까지는 다시 그려집니다. 이 방식은 자연 이미지의 중요한 속성, 즉 조명의 강도와 색상 변화에 따라 물체의 동일성이 변하지 않는다는 점을 대략적으로 포착합니다. 이 방식은 상위 1%의 오류율을 1% 이상 줄입니다.

4.2 드롭아웃

다양한 모델의 예측을 결합하는 것은 테스트 오류를 줄이는 매우 성공적인 방법이지만[1, 3], 이미 훈련하는 데 며칠이 걸리는 대규모 신경망에는 너무 많은 비용이 드는 것으로 보입니다. 그러나 훈련하는 동안 약 2분의 1의 비용만 드는 매우 효율적인 모델 조합 버전이 있습니다. 최근에 도입된 "드롭아웃"[10]이라는 기술은 각 숨겨진 뉴런의 출력을 0.5의 확률로 0으로 설정하는 것으로 구성됩니다. 이렇게 "드롭아웃"된 뉴런은 순방향 전달에 기여하지 않으며 역전파에도 참여하지 않습니다. 따라서 입력이 제시될 때마다 신경망은 다른 아키텍처를 샘플링하지만 이 모든 아키텍처는 가중치를 공유합니다. 이 기술은 뉴런이 특정 다른 뉴런의 존재에 의존할 수 없기 때문에 뉴런의 복잡한 공동 적응을 줄입니다. 따라서 다른 뉴런의 다양한 무작위 하위 집합과 함께 유용한 더 강력한 기능을 학습해야 합니다. 테스트 시에는 모든 뉴런을 사용하되, 기하급수적으로 많은 드롭아웃 네트워크에서 생성된 예측 분포의 기하평균을 취하는 데 적합한 근사치인 0.5를 출력에 곱합니다.

그림 2의 처음 두 개의 완전히 연결된 레이어에서 드롭아웃을 사용합니다. 드롭아웃이 없으면 네트워크는 상당한 과적합을 나타냅니다. 드롭아웃은 수렴에 필요한 반복 횟수를 약 두 배로 늘립니다.

5 학습의 세부 사항

그림 3: 224×224×3개의 입력 이미지에 대해 첫 번째 컨볼루션 레이어에서 학습한 11×11×3 크기의 96개의 컨볼루션 커널. 상위 48개의 커널은 GPU 1에서 학습되고
GPU에서 학습된 하위 48개 커널입니다.
2. 자세한 내용은 섹션 6.1을 참조하십시오.

배치 크기 128개, 모멘텀 0.9, 가중치 감쇠 0.0005의 확률적 경사 하강을 사용하여 모델을 훈련했습니다. 이 소량의 가중치 감쇠가 모델의 학습에 중요하다는 것을 발견했습니다. 즉, 여기서 가중치 감쇠는 단순한 정규화가 아니라 모델의 학습 오류를 줄여줍니다. 가중치 w_i에 대한 업데이트 규칙은 다음과 같습니다.

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

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

여기서 i는 반복 지수, v는 운동량 변수, ϵ는 학습 속도, $\left\langle \frac{\partial L}{\partial w_i} \right\rangle_{D_i}$는 w_i에 대한 목표 미분의 i번째 배치 D_i에 대한 평균으로, 다음에서 평가됩니다.

각 층의 가중치는 표준 편차 0.01의 제로 평균 가우스 분포에서 초기화했습니다. 두 번째, 네 번째, 다섯 번째 컨볼루션 층과 완전히 연결된 숨겨진 층의 뉴런 편향은 상수 1로 초기화했습니다. 이 초기화는 ReLU에 양의 입력을 제공함으로써 학습의 초기 단계를 가속화합니다. 나머지 레이어의 뉴런 편향은 상수 0으로 초기화했습니다.

모든 레이어에 동일한 학습률을 사용했으며, 학습을 진행하는 동안 수동으로 조정했습니다. 우리가 따랐던 휴리스틱은 유효성 검사 오류율이 현재 학습률로 개선되지 않을 때 학습률을 10으로 나누는 것이었습니다. 학습률은 0.01로 초기화되어 종료 전에

종료하기 전에 세 번 감소했습니다. 120만 개의 이미지로 구성된 훈련 세트를 통해 약 90주기 동안 네트워크를 훈련했으며, 두 대의 NVIDIA GTX 580 3GB GPU에서 5~6일이 소요되었습니다.

6 결과

ILSVRC-2010에 대한 결과는 표 1에 요약되어 있습니다. 우리 네트워크의 상위 1위 및 상위 5위 테스트 세트 오류율은 **37.5**%와 **17.0**%입니다[5]. ILSVRC-2010 대회에서 달성한 최고 성능은 서로 다른 특징에 대해 훈련된 6개의 스파스 코딩 모델에서 생성된 예측을 평균하는 접근 방식에서 47.1%와 28.2%였으며[2], 이후 발표된 최고 결과는 두 가지 유형의 고밀도 샘플링 특징에서 계산된 피셔 벡터(FV)로 훈련된 두 클래시 파이어의 예측을 평균하는 방식에서 45.7%와 25.7%입니다 [24].

또한 저희 모델을 ILSVRC-2012 컴피티션에 출품하여 표 2에 결과를 보고했습니다. ILSVRC-2012 테스트 세트 라벨은 공개되지 않았기 때문에 저희가 시도한 모든 모델에 대한 테스트 오류율을 보고할 수는 없습니다. 이 단락의 나머지 부분에서는 다음을 사용합니다.

모델	**Top-1**	**Top-5**
스파스 코딩 [2]	*47.1%*	*28.2%*
SIFT+ FV [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

표 1: ILSVRC-2010 테스트 세트의 결과 비교. *이탤릭체로* 표시된 것은 다른 CNN이 달성한 최고 결과입니다.

검증 오류율과 테스트 오류율은 경험상 0.1% 이상 차이가 나지 않기 때문에 같은 의미로 사용합니다(표 2 참조). 이 백서에서 설명하는 CNN의 상위 5개 오류율은 18.2%입니다. 예측 평균화 유사한 CNN 5개를 학습시키면 16.4%의 오류율을 보입니다. 마지막 풀링 레 이 어 위에 6번째 컨볼루션 레이어를 추가한 하나의 CNN을 훈련시켜 전체 ImageNet 2011 가을 릴리스(1,500만 이미지, 22만 카테고리)를 분류한 다음 ILSVRC-2012에서 '미세 조정'하면 16.6%의 오류율이 나옵니다. 2011년 가을 전체 릴리스에 대해 사전 학습된 두 개의 CNN과 앞서 언급한 다섯 개의 CNN의 예측을 평균하면 **15.3%의** 오차율이 나옵니다. 두 번째로 좋은 컨테스트 항목은 다양한 유형의 조밀하게 샘플링된 특징으로부터 계산된 FV로 훈련된 7개의 분류기의 예측을 평균하는 접근 방식으로 26.2%의 오류율을 달성했습니다[7].

마지막으로, 10,184개의 카테고리와 890만 개의 이미지가 포함된 2009년 가을 버전의 ImageNet에 대한 오류율도 보고합니다. 이 데이터 세트에서는 이미지의 절반을 학습용으로, 절반을 테스트용으로 사용하는 문헌의 관례를 따릅니다. 이 데이터 세트에는

Model	**Top-1 (val)**	**Top-5 (val)**	**Top-5 (테스트)**
SIFT+ FV [7]	-	-	*26.2%*
1 CNN	40.7%	18.2%	-
5개 CNN	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	-
7개 CNN*	36.7%	15.4%	**15.3%**

표 2: ILSVRC-2012 유효성 검사 및 테스트 세트의 오류율 비교. *이탤릭체로* 표시된 것은 다른 모델에서 얻은 최상의 결과입니다. 별표*가 있는 모델은 이미지넷 2011 가을 릴리스 전체를 분류하기 위해 "사전 학습"된 모델입니다. 자세한 내용은 섹션 6을 참조하십시오.

탭에 설정된 테스트 세트의 분할은 이전 작성자가 사용한 분할과 약간 다르지만 결과에 큰 영향을 미치지는 않습니다. 이 데이터 세트의 상위 1% 및 상위 5% 오류율은 **67.4%입니다**. **40.9%**, 위에서 설명한 네트워크에 마지막 풀링 레이어 위에 6번째 컨볼루션 레이어를 추가하여 얻은 결과입니다. 이 데이터 세트에 대해 발표된 최고 결과는 78.1%와 60.9%입니다[19].

6.1 정성적 평가

그림 3은 네트워크의 두 데이터 연결 레이어가 학습한 컨볼루션 커널을 보여줍니다. 네트워크는 다양한 주파수 및 방향 선택적 커널과 다양한 색상의 블롭을 학습했습니다. 섹션 3.5에서 설명한 제한된 연결성의 결과인 두 개의 GPU가 보여주는 전문화에 주목하세요. GPU 1의 커널은 대부분 색상에 구애받지 않는 반면, GPU 2의 커널은 대부분 특정 색상에 특화되어 있습니다. 이러한 종류의 전문화는 모든 실행 중에 발생하며 특정 무작위 가중치 초기화(모듈로 GPU의 번호 변경)와는 무

관합니다.

[5]섹션 4.1에서 설명한 대로 10개의 패치에 대한 예측을 평균화하지 않은 오류율은 39.0%와 18.3%입니다.

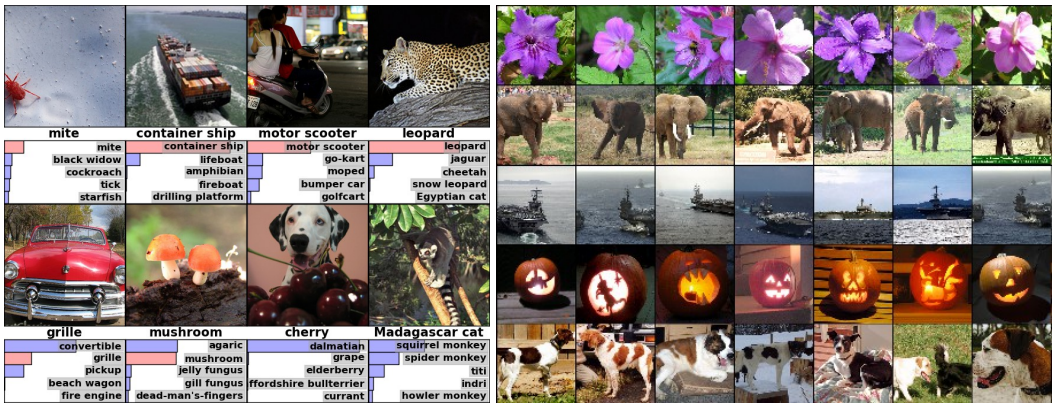

그림 4: **(왼쪽)** 8개의 ILSVRC-2010 테스트 이미지와 모델에서 가장 가능성이 높은 것으로 간주되는 5개의 레이블. 각 이미지 아래에 올바른 레이블이 표시되어 있으며, 올바른 레이블에 할당된 확률도 빨간색 막대로 표시됩니다(상위 5개에 속하는 경우). **(오른쪽)** 첫 번째 열에 있는 5개의 ILSVRC-2010 테스트 이미지. 나머지 열은 테스트 이미지의 특징 벡터와 유클리드 거리가 가장 작은 마지막 숨겨진 레이어에서 특징 벡터를 생성하는 6개의 훈련 이미지를 보여줍니다.

그림 4의 왼쪽 패널에서는 8개의 테스트 이미지에 대한 상위 5개 예측을 계산하여 네트워크가 학습한 내용을 정성적으로 평가합니다. 왼쪽 상단의 진드기처럼 중심에서 벗어난 물체도 네트워크에서 인식할 수 있음을 알 수 있습니다. 상위 5개 레이블의 대부분은 합리적인 것으로 보입니다. 예를 들어 표범의 경우 다른 종류의 고양이만 그럴듯한 레이블로 간주됩니다. 어떤 경우(그릴, 체리)에는 사진의 의도된 초점이 모호한 경우도 있습니다.

네트워크의 시각적 지식을 조사하는 또 다른 방법은 마지막 4096차원 숨겨진 레이어에서 이미지에 의해 유도된 특징 활성화를 고려하는 것입니다. 두 이미지가 유클리드 간격이 작은 특징 활성화 벡터를 생성하면 신경망의 상위 수준에서는 두 이미지가 유사하다고 간주한다고 말할 수 있습니다. 그림 4는 이 측정에 따라 테스트 세트의 이미지 5개와 훈련 세트의 이미지 6개 중 가장 유사한 이미지를 보여줍니다. 픽셀 수준에서 검색된 훈련 이미지가 일반적으로 첫 번째 열의 쿼리 이미지와 L2가 가깝지 않다는 것을 알 수 있습니다. 예를 들어, 검색된 개와 코끼리는 다양한 포즈를 취하고 있습니다. 보충 자료에서 더 많은 테스트 이미지에 대한 결과를 제시합니다.

두 개의 4096차원 실수값 벡터 사이의 유클리드 거리를 사용하여 유사도를 계산하는 것은 비효율적이지만, 이러한 벡터를 짧은 이진 코드로 압축하도록 자동 인코더를 훈련시키면 효율적으로 만들 수 있습니다. 이렇게 하면 이미지 레이블을 사용하지 않기 때문에 의미적으로 유사한지 여부와 관계없이 가장자리 패턴이 비슷한 이미지를 검색하는 경향이 있는 원시 픽셀에 자동 인코더를 적용하는 방법[14]보다 훨씬 더 나은 이미지 검색 방법을 생성할 수 있습니다.

7 토론

우리의 결과는 대규모 심층 컨볼루션 신경망이 순수 지도 학습을 사용하여 매우 까다로운 데이터 세트에서 기록적인 결과를 달성할 수 있음을 보여줍니다. 하나의 컨볼루션 계층을 제거하면 네트워크의 성능이 저하된다는 점 은 주목할 만합니다. 예를 들어, 중간 계층을 하나라도 제거하면 네트워크의 상위 1% 성능이 약 2% 저하됩니다. 따라서 결과를 얻으려면 깊이가 정말 중요합니다.

실험을 단순화하기 위해, 특히 라벨링된 데이터의 양을 늘리지 않고도 네트워크의 크기를 크게 늘릴 수 있는 충분한 계산 능력을 얻을 수 있다면 도움이 될 것으로 예상되지만 비지도 사전 학습을 사용하지 않았습니다. 지금까지 네트워크를 더 크게 만들고 더 오래 학습시키면서 결과가 개선되었지만, 인간 시각 시스템의 시공간적 경로와 일치하기 위해서는 아직 갈 길이 많이 남았습니다. 궁극적으로는 시간적 구조가 정적 이미지에서 누락되거나 훨씬 덜 분명한 매우 유용한 정보를 제공하는 비디오 시퀀스에 매우 크고 심층적인 컨볼루션 네트워크를 사용하고자 합니다.

참고 문헌

[1] R.M. Bell과 Y. Koren. 넷플릭스 프라이즈 챌린지에서 얻은 교훈. *ACM SIGKDD 탐색 뉴스레터*, 9(2):75-79, 2007.

[2] A. Berg, J. Deng, and L. Fei-Fei. 대규모 시각 인식 챌린지 2010. www.image-net.org/challenges. 2010.

[3] L. Breiman. 랜덤 포레스트. *기계 학습*, 45(1):5-32, 2001.

[4] D. Cires¸an, U. Meier, and J. Schmidhuber. 이미지 분류를 위한 다중 열 심층 신경망. *아카이브 사전 인쇄 arXiv:1202.2745*, 2012.

[5] D.C. Cires¸an, U. Meier, J. Masci, L.M. Gambardella, 및 J. Schmidhuber. 시각적 객체 분류를 위한 고성능 신경망. *아카이브 사전 인쇄 arXiv:1102.0183*, 2011.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, 및 L. Fei-Fei. ImageNet: 대규모 계층적 이미지 데이터베이스. In *CVPR09*, 2009.

[7] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. *ILSVRC-2012*, 2012. URL http://www.image-net.org/challenges/LSVRC/2012/.

[8] L. 페이 페이, R. 퍼거스, P. 페로나. 소수의 훈련 예제에서 생성적 시각 모델 학습: 101개의 객체 범주에 대해 테스트한 증분적 베이지안 접근법. *컴퓨터 비전 및 이미지 이해*, 106(1):59-70, 2007.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 객체 카테고리 데이터 세트. 기술 보고서 7694, 캘리포니아 공과 대학, 2007. URL http://authors.library.caltech.edu/7694.

[10] G.E. 힌튼, N. 스리바스타바, A. 크리제프스키, I. 수츠케버, 및 R.R. 살라쿠트디노프. 특징 검출기의 공동 적응을 방지하여 신경망 작업 개선. *arXiv 사전 인쇄물 arXiv:1207.0580*, 2012.

[11] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, 및 Y. LeCun. 물체 인식을 위한 최고의 다단계 아키텍처는 무엇인가요? *컴퓨터 비전 국제 컨퍼런스*, 2146-2153페이지. IEEE, 2009.

[12] A. 크리제프스키. 작은 이미지에서 여러 계층의 특징 학습하기. 토론토 대학교 컴퓨터 과학과 석사 학위 논문, 2009.

[13] A. 크리제프스키. cifar-10의 컨볼루션 심층 신념 네트워크. *미발표 원고*, 2010.

[14] A. 크리제프스키와 G.E. 힌튼. 콘텐츠 기반 이미지 검색을 위해 매우 심층적인 자동 인코더 사용. In *ESANN*, 2011.

[15] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel 등. 역전파 네트워크를 이용한 손글씨 숫자 인식, 역전파 네트워크를 이용한 손글씨 숫자 인식. *신경 정보 처리 시스템의 발전*, 1990.

[16] Y. LeCun, F.J. Huang, and L. Bottou. 포즈와 조명에 불변성을 갖는 일반적인 물체 인식을 위한 학습 방법. In *컴퓨터 비전 및 패턴 인식, 2004. CVPR 2004. 2004 IEEE 컴퓨터 학회 학술대회* 논문집, 2권, II-97페이지. IEEE, 2004.

[17] Y. LeCun, K. Kavukcuoglu, 및 C. Farabet. 비전에서의 컨볼루션 네트워크 및 응용. *회로 및 시스템(ISCAS), 2010 IEEE 국제 심포지엄 논문집*, 253-256페이지. IEEE, 2010.

[18] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. 계층적 표현의 확장 가능한 비지도 학습을 위한 컨볼루션 심층 신념 네트워크. *제26회 연례 기계 학습 국제 컨퍼런스 논문집*, 609-616페이지. ACM, 2009.

[19] T. 멘싱크, J. 버벡, F. 페로닌, G. 크수르카. 대규모 이미지 분류를 위한 메트릭 학습: 거의 제로에 가까운 비용으로 새로운 클래스로 일반화하기. *ECCV - 유럽 컴퓨터 비전 컨퍼런스*, 이탈리아 피렌체, 2012년 10월.

[20] V. Nair와 G. E. Hinton. 정류된 선형 단위로 제한된 볼츠만 머신 개선. *제27회 기계 학습 국제 컨퍼런스*, 2010.

[21] N. 핀토, D.D. 콕스, J.J. 디카를로. 실제 시각적 물체 인식은 왜 어려운가? *PLoS 전산 생물학*, 4(1):e27, 2008.

[22] N. 핀토, D. 두칸, J.J. 디카를로, D.D. 콕스. 생물학적으로 영감을 받은 시각적 표현의 좋은 형태를 발견하기 위한 고처리량 스크리닝 접근법. *PLoS 계산 생물학*, 5(11):e1000579, 2009.

[23] B.C. 러셀, A. 토랄바, K.P. 머피, W.T. 프리먼. Labelme: 이미지 주석을 위한 데이터베이스 및 웹 기반 도구. *국제 컴퓨터 비전 저널*, 77(1):157-173, 2008.

[24] J. 산체스 및 F. 페로닌. 대규모 이미지 분류를 위한 고차원 서명 압축. *컴퓨터 비전 및 패턴 인식(CVPR), 2011 IEEE 컨퍼런스*, 1665-1672페이지. IEEE, 2011.

[25] P.Y. Simard, D. Steinkraus, and J.C. Platt. 시각적 문서 분석에 적용된 컨볼루션 신경망의 모범 사례. *제7회 국제 문서 분석 및 인식 컨퍼런스* 논문집, 2권, 958-962쪽, 2003.

[26] S.C. Turaga, J.F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, H.S. Seung. 컨볼루션 네트워크는 이미지 분할을 위한 선호도 그래프를 생성하는 방법을 학습할 수 있습니다. *신경 계산*, 22(2):511-538, 2010.