대표적인 합성곱 신경망 모델의 비교

권하연, 김소정, 배인우, 이강은

Comparision of representive Convolution Neural Network models

KwonHayeon, KimSojeong, BaeInwoo, LeeKangeun

요 약

본 보고서에서는 널리 알려진 합성곱 신경망 모델 AlexNet, VGG, GoogLeNet, ResNet을 종합적으로 비교한다. 우선 각 모델들의 구조와 특징을 소개한 후 Pytorch의 모델 라이브러리를 이용하여 CIFAR-10 데이터 셋의 분류 성능을 실험한다. 실험에서는 모델의 훈련과 테스트 정확도, 소요 시간을 포함하여 모델의 주요 특성과 성능을 분석하고 평가한다. 여전히 많은 분야에서 Backbone Network등으로 활용되고 있는 모델[1]들을 소개 및 평가하며 CNN 아키텍쳐의 발전에 대한 통찰력을 제공하고 현재 이루어지는 컴퓨터 비전 연구를 이해하는 데에 도움을 주고자 한다.

Abstract

In this report, we comprehensively compare well-known convolutional neural network models AlexNet, VGG, GoogLeNet, and ResNet. First, after introducing the structure and characteristics of each model, we experiment with the classification performance of the CIFAR-10 dataset using Pytorch's model library. In the experiment, the key characteristics and performance of the model are analyzed and evaluated, including the training and test accuracy of the model, and the time required. It introduces and evaluates models that are still used as Backbone Networks in many fields[1], provides insight into the development of CNN architecture, and helps to understand the current computer vision research.

Key words

convolutional neural network, CNN model, deep learning, image classification, architecture

Ⅰ. 서 론

Convolution Neural Network(CNN)가 발표된 이후 모델의 성능을 향상시키려는 시도가 계속되었다. CNN모델의 기초인 LeNet[2] 발표 후 14년만에 나온 AlexNet[3]은 GPU의 병렬화와 활성화 함수 ReLU의 사용으로 큰 변화를 가져다 주었다. 이후 GPU의 발전과 데이터 셋의 크기의 증가로 더 높은 성능의 모델 연구가 진행되었다. 그 결과 CNN 모델은 VGG[4], GoogLenet[5], Resnet[6] 등 다양한 방법으로 진화하였다. 이들은 모두 합성곱 레이어와 완전 연결 계층(Fully Connected Layer)을 갖춘 CNN을 기본구조로 하며 계층의 깊이와 커널의 크기를 달리하고 국소적 정규화나 배치 정규화, dropout 등이 추가되는 등의 차이를 가진다.

해당 모델들은 모두 ImageNet 데이터셋에서 우

수한 성능을 보였으며 ImageNet Large-Scale Visual Recognition Challenge(ILSVRC)에서 높은 성적을 거두었다. 본 보고서에서는 이미지 분류를 위해 제안되었던 대표적인 네 가지 모델을 소개하고 CIFAR-10 데이터셋 분류를 실행한다. 실험 환경으로는 구글의 CoLab을 사용하였으며 Pytorch의 라이브러리에 있는 사전학습된 모델을 이용하였다.

Ⅱ. 관련 연구

1. Alexnet[3]

AlexNet은 컴퓨터 비전 분야를 발전시키는 데 중추적인 역할을 한 CNN 모델이다. 이전 방법보다 훨씬 뛰어난 성능을 보여주며 ILSVRC-2012에서 우승하였다.

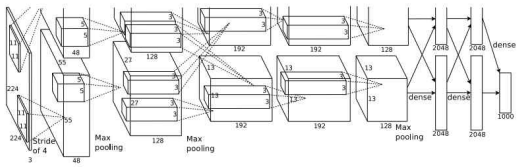

그림 1 AlexNet 구조

AlexNet은 8개의 레이어, 5개의 컨볼루션 레이어 및 3개의 완전히 연결된 레이어로 구성된다. 기존에 활성화 함수로 tanhx를 사용했던 것과 달리 ReLU를 사용했다. 이를 통해 AlexNet은 더 빨리 낮은 에러율에 도달할 수 있었다. 정규화를 위해 국소적 정규화를 사용하였으며 과적합(Overfitting)을 방지하기 위해 의도적으로 은닉층 일부를 무시하는 Dropout을 사용하였다. 최적화 방법으로는 모멘텀이 있는 확률적 경사하강법을 이용하였다.

AlexNet은 심층 신경망 훈련을 위해 그래픽 처리 장치(GPU)를 최초로 활용하기도 했다. GPU의 병렬화는 training 시간을 크게 단축하여 딥러닝에 대한 접근성을 높였다.

AlexNet은 ILSVRC-2010에서 top-5 error 17.0%와 top-1 error 37.5%를 각각 기록하였다. 이때 top-5 error란 가장 확률이 높은 분류 5개 중에 정답이 없을 확률이며 top-1 error는 가장 확률이 높은 라벨이 정답이 아닐 확률을 의미한다. 이는 기존까지의 다른 분류 방법보다 훨씬 높은 정확도였다.

2. VGG[4]

VGG(Visual Geometry Group)는 옥스퍼드 대학교의 연구 그룹이 개발한 CNN 모델이다. VGG는 일관성 있는 아키텍처의 장점을 극대화한 모델로서, 모든 레이어에서 동일한 작은 필터 크기(3x3)와 동일한 보폭(stride, 1)을 사용한다. VGG-16과 VGG-19라는 두 가지 주요 버전이 있는데, 이 숫자는 네트워크의 깊이(층의 수)를 나타낸다.

VGG의 핵심 차별점은 일관된 아키텍처와 깊이에 존재한다. 이전의 CNN 모델, 예를 들어 AlexNet은 여러 크기의 필터(11x11, 5x5, 3x3)를 사용했으나 VGG는 작은 3x3 필터만 사용한다. 이런 일관성 있는 구조는 네트워크 구조를 단순화하고 이해하기 쉽게 만든다.

VGG의 가장 큰 특징은 작은 필터를 여러 층에 걸쳐 사용한 것이다. 이로 인해 더 많은 비선형성이 증가한다. 예컨대, 1-layer 7x7 필터링의 경우 한 번의 비선형 함수가 적용되는 반면 3-layer 3x3 필터링은 세 번의 비선형 함수가 적용되기 때문이다. 결과적으로 레이어가 증가함에 따라 비선형성이 증가하게 되고 이것은 모델의 특징 식별성 증가로 이어진다.

작은 필터를 사용하는 것의 또 다른 장점은 파라미터 수를 크게 줄일 수 있다는 점이다. 예컨대, 7x7필터 1개에 대한 학습 파라미터 수는 49이고 3x3 필터 3개에 대한 학습 파라미터 수는 27(3x3x3)이 된다. 다만 이때 특징 맵들이 여러 레이어를 거쳐 만들어지기 때문에 더 추상적인 정보를 담게 된다는 점을 유의해야 한다. 모델의 사용 목적에 따라 요구되는 특징맵이 다르기 때문에 보완이 필요할 수도 있다.

한편 VGG중에서도 VGG-16과 VGG-19를 결합한 모델은 top-5 error 6.8%이라는 특히 높은 정확도를 보였다. 그러나 깊이가 깊어질수록 계산 복잡성이 증가하고 과적합 문제가 생긴다는 단점이 있다.

3. GoogLenet[5]

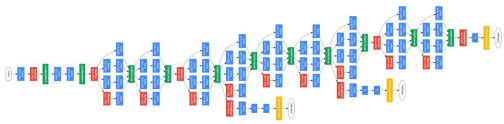

그림 2 GoogLeNet 구조

GoogLeNet은2014년에 발표된 CNN 모델이다. GoogLeNet은 Inception이라는 개념을 도입하여 효율적인 네트워크 구조를 구현한다. Inception 모듈은 다양한 커널 크기와 다른 깊이의 컨볼루션 레이어를 동시에 사용하여 입력 데이터의 다양한 특징을 추출하는 것을 목표로 한다. 이 구조를 활용하여 AlexNet보다 더 깊지만 파라미터 수는 1/12인 효율적인 구조를 설계하였다.

GoogLeNet의 아키텍처는 깊은 신경망으로 구성되어 있으며, 전체적으로는 Inception 모듈과 pooling layer, 완전 연결 계층으로 구성된다. Inception 모듈은 여러 개의 병렬 컨볼루션 레이어로 구성되어 있으며, 각 레이어는 세 개의 서로 다른 크기의 커널(1x1, 3x3, 5x5)을 사용하여 특징을 추출한다. 이러한 다양한 크기의 커널을 통해 모델은 입력 이미지의 다양한 스케일에서 특징을 탐지할 수 있다.

GoogLeNet은 pooling layer를 사용하여 공간 크기를 줄이고 추상화된 특징을 인코딩한다. 이후에는 전연결층이 이 추상화된 특징을 기반으로 입력 이미지의 클래스를 분류한다. GoogLeNet은 마지막으로 소프트맥스(softmax) 활성화 함수를 사용하여 각 클래스에 대한 확률을 출력한다.

GoogLeNet은 대규모 이미지 데이터셋인 ImageNet에서 학습되었으며, 다양한 컴퓨터 비전 작업에서 높은 성능을 보여주는 기본 아키텍처로 사용되어 왔다. 그러나 이전 모델에 비해 훨씬 깊고 복잡한 구조를 가지고 있어 학습 및 추론 시에 더 많은 계산 리소스와 처리 시간을 요구하는 복잡성의 문제가 있으며, ImageNet과 같은 대규모 데이터셋에서는 높은 성능을 보이지만, 작은 규모의 데이터셋에서는 과적합을 보이는 경향이 있다.

GoogLeNet은 CNN 구조에 대해 깊게 고민하여 설계된 효율적이고 높은 성능을 갖는 네트워크다. GoogLeNet은 모델 구조 등을 지속적으로 개선해가며 현재까지 GoogLeNet-V1, GoogLeNet-V2, GoogLeNet-V3, GoogLeNet-V4, 총 4가지의 버전이 발표되었다. GoogLeNet은 다양한 컴퓨터 비전 작업에서 기본 아키텍처로 사용되어 왔으며, 심층 신경망의 발전에 큰 기여를 한 모델 중 하나이다.

4. ResNet[6]

ResNet은 Residual Network의 줄임말로 마이크로소프트팀이 2015년에 제안한 모델이다. 이 모델은 매우 깊은 모델에서 발생하는 vanishing gradient에 대한 문제를 해결하기 위해 제안되었다.

많은 레이어를 쌓는 것은 분류 성능을 향상시키는데 도움을 주었지만 vanishing/exploding gradient와 Degradation의 발생으로 정확도가 오히려 떨어지는 부작용을 초래했다. ResNet에서는 이에 대한 해결 방안으로 네트워크가 하나 이상의 계층을 건너뛰고 해당 연결에 잔여 블록을 추가하는 방법을 도입했다.

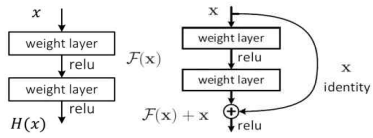

그림 3 잔차의 이용

기존 네트워크는 x를 입력받고 H(x)를 출력함으로써 mapping 함수 H(x)를 얻는 것을 목적으로 했다. 그러나 ResNet은 예측함수 H(x)와 입력값 x의 차이인 F(x) = H(x) - x의 값을 얻는 것을 목적으로 한다. 예측이 정확해질수록 차이는 줄어들며, 가장 이상적인 출력값은 0이다. 즉, 최종적으로 H(x)=x로 mapping 하는 것이 학습의 목표가 된다. 알지 못하는 H(x)를 학습시키는 문제에서 H(x)=x라는 목표값이 제공되는 문제가 되며 학습이 더 쉬워진 것이다. 이렇게 F(x)에 x를 더하는 shortcut connection은 네트워크 구조를 크게 변경하지 않아도 될 뿐만 아니라 덧셈을 제외하면 연산량 증가도 없다는 장점이 있다.

ResNet은 ResNet-18, ResNet-50, ResNet-101, ResNet-152와 같이 다양한 깊이의 아키텍처가 있다. 이 숫자들은 네트워크의 총 계층의 수를 나타내며 기존 VGG보다 최대 8배 깊은 층을 가진다. 그럼에도 깊은 층이 가진 vanishing gradient 등과

같은 정확도 저하 문제 없이 ILSVRC-2015에서 top-5 error 3.57%를 기록했다.

Ⅲ. 실험 과정

1. 실험 환경

본 연구에서는 PyTorch를 사용하여 딥러닝 환경을 구축하였으며, 동일한 환경에서 4개의 실험을 병렬적으로 진행하였다. 실험에서는 AlexNet, VGG, GoogLeNet, ResNet 등 4개의 모델을 사용하였다. 이때 모델 간의 차이 외의 다른 변수는 철저하게 통제하였다. 4개의 모델은 모두 ImageNet을 기반으로 사전에 학습된 가중치로 초기화한 상태에서 시작하였다.

실험에서는 높은 성능의 그래픽 처리 장치(GPU)를 사용하여 연산 성능을 향상시키기 위해 NVIDIA GeForce RTX 3060을 사용하였다. 이를 통해 딥러닝 모델의 학습과 추론 속도를 효율적으로 개선하였으며, 실험을 위해 32GB의 시스템 메모리와 i7-12700 CPU를 갖춘 컴퓨터 환경에서 동일하게 작업하였다.

실험에는 상기한 바와 같이 PyTorch 2.0.1 버전을 사용하였으며, CUDA Toolkit 12.0.1과 cuDNN 8.8.1을 이용하여 GPU 가속을 진행하였다. 이를 통해 딥러닝 모델의 학습과 추론 과정에서 최적의 성능을 설정할 수 있었으며, 오픈 소스 라이브러리 및 패키지인 torch, torchvision과 진행 과정을 파악하기 위한 tqdm 라이브러리를 사용하여 실험을 진행하였다.

2. 데이터셋 소개 & 데이터셋 전처리

본 연구에서는 CIFAR-10 데이터셋을 활용하여 실험을 진행하였다. CIFAR-10은 컴퓨터 비전 분야에서 널리 사용되는 대표적인 이미지 분류 데이터셋으로, 10개의 클래스로 구성되어 있다. 각 클래스는 다양한 객체와 장면을 대표하는 컬러 이미지를 포함하고 있으며, 각 클래스당 6,000장의 해상도가 32x32인 이미지로 구성되어 있다. 이 데이터셋은 다양한 객체 인식과 분류 문제에 적합한 특징을 가지고 있기 때문에 이번 실험에서의 데이터셋으로 사용하였다.

실험을 위해 CIFAR-10 데이터셋을 로드한 후, 다음과 같은 전처리 과정을 통해 데이터를 준비하였다. 우선, transforms.ToTensor() 함수를 사용하여 이미지를 텐서 형식으로 변환하였다. 이 과정을 통해 이미지 데이터를 모델의 입력 형식에 맞게 조정하였다. 다음으로, transforms.Normalize() 함수를 활용하여 이미지의 각 채널을 정규화하였다. 이를 위해 평균과 표준편차를 활용하여 각 채널별로 평균 0.5, 표준편차 0.5로 정규화를 수행하였다. 이를 통해 입력 데이터의 범위를 일정하게 조정하여 모델의 학습 과정의 안정화와 성능 개선을 주도하였다.

3. 학습 진행

본 연구에서는 PyTorch를 기반으로 딥러닝 환경을 구성하였고, CIFAR-10 데이터셋을 활용하여 다양한 딥러닝 모델의 학습을 진행하였다. 실험에서는 AlexNet, ResNet18, GoogLeNet, VGG16 등의 유명한 모델을 사용하였으며, 모델의 차이를 제외한 다른 변수들을 철저하게 통제하였다. 각 모델은 ImageNet 데이터셋을 기반으로 사전 학습된 가중치를 초기값으로 설정하였다.

학습을 시작하기 전에, 선택한 모델에 따라 적절한 모델 인스턴스를 생성하였다. 사용자 입력에 따라 AlexNet, ResNet18, GoogLeNet, VGG16 중 하나의 모델을 선택하도록 구현하였으며, 선택된 모델은 PyTorch의 모델 클래스에 의해 ImageNet으로 학습된 가중치를 포함하여 생성되었다. 모델은 CUDA가 사용 가능한 경우 GPU로 이동시켜 학습을 가속화하였다.

학습에는 CrossEntropyLoss를 손실 함수로 사용하였고, Adam 옵티마이저를 사용하여 모델의 파라미터를 최적화하였다. 학습 과정은 주어진 epoch 수인 총 10회만큼 반복하여 진행하였다. 각 epoch마다 모델을 훈련 상태로 설정하고, 훈련 데이터셋을 미니배치로 나누어 훈련을 진행하였다. 또한 손실 함수를 이용하여 손실을 계산하고, 옵티마이저를 통해 역전파된 그래디언트를 이용하여 가중치를 업데이트하였다.

각 epoch의 학습 과정에서는 훈련 손실과 정확

도를 계산하여 출력하였다. 훈련 손실은 미니배치 손실의 평균으로 계산되었으며, 훈련 정확도는 예측과 실제 레이블을 비교하여 계산하였다. 이를 통해 각 epoch의 훈련 상태를 모니터링하고 모델의 학습 진행을 파악할 수 있었다.

훈련이 완료된 후에는 모델을 평가 상태로 설정하여 테스트 데이터셋에 대한 예측을 수행하였다. 예측 결과인 테스트 정확도를 계산하여 출력하였으며, 이는 모델의 일반화 성능을 평가하는데 사용되었다.

Ⅳ. 결 론

각 모델을 학습시킨 결과는 다음과 같다.

	소요 시간(초)	정확도(%)
AlexNet	17	70.40
VGG16	129	70.66
GoogLeNet	26	82.46
ResNet18	17	79.14

1. 실험 결과

AlexNet의 정확도가 70.40%로 비교적 낮게 나타난 이유는 AlexNet이 최근의 신경망 모델에 비해 구조적으로 단순하고 깊이가 낮기 때문이다. 반면 그러한 이유로 소요 시간은 짧게 나타났다. CIFAR-10 데이터셋과 같은 복잡한 이미지 분류 문제에는 상대적으로 복잡한 모델이 더 좋은 성능을 보이는 경향이 있기 때문에 AlexNet은 다른 모델과 비교했을 때 정확도가 낮게 나타난 것으로 예상된다.

VGG16의 정확도가 70.66%로 비교적 낮게 나타난 이유는 VGG16의 깊고 많은 층으로 인해 모델이 복잡해지고 계산 비용이 증가하기 때문이다. 이 모델은 16개의 합성곱 층과 3개의 완전 연결층으로 구성되어 있고 작은 필터 크기인 3x3을 여러 번 적용하여 깊은 구조를 형성한다. 이러한 이유로 이미지의 다양한 특징을 추출할 수 있으나, CIFAR-10 데이터셋과 같은 작은 크기의 이미지에서는 파라미터 수가 많아서 과적합의 가능성이 있고, 계산 비용도 높아져 정확도가 상대적으로 낮아지고 소요 시간이 늘어난 것으로 예상된다. 특히, VGG16은 깊고 연속된 층으로 구성되어 있어 병렬 처리가 어려워지기 때문에 학습 및 추론 시간이 상대적으로 더 오래 걸리게 되었다.

GoogLeNet의 정확도가 82.46%로 높게 나타난 이유는 인셉션 모듈의 활용과 경량화된 네트워크 구조 때문이다. 인셉션 모듈은 다양한 필터 크기를 사용하여 객체 인식과 분류에 효과적인 특징을 추출할 수 있다. 또한 GoogLeNet은 비교적 더 작은 파라미터 수를 갖고 있으며, 이는 학습과정에서 더 효율적인 계산을 가능하게 한다. 따라서 CIFAR-10 데이터셋에서도 높은 정확도를 보일 수 있었다.

ResNet18의 정확도가 79.14%로 상대적으로 높게 나타난 이유는 Residual Network의 특성 때문이다. Residual Network는 스킵 연결을 통해 그래디언트 소실 문제를 해결하고, 네트워크가 깊어질수록 더 나은 성능을 발휘할 수 있는 구조를 갖추고 있다. 따라서 CIFAR-10 데이터셋의 작은 크기인 32x32와 같은 경우에도 높은 정확도를 보일 수 있었다. 또한, 이러한 특징은 전파가 원활하게 이루어지게 하므로 병렬 처리에 유리하여 시간을 단축하는 데에 효과적이다. ResNet18은 깊은 네트워크 구조와 함께 작은 필터 크기를 사용하여 지역적인 특징을 잘 포착할 수 있기 때문에, 작은 크기의 이미지에서도 객체 인식과 분류에 효과적인 특징을 추출할 수 있었을 것으로 예상된다.

2. 최종 결론

이렇게 본 연구를 통해 AlexNet, ResNet18, GoogLeNet, VGG16 네 가지 딥러닝 모델을 CIFAR-10 데이터셋을 활용하여 평가하였다. 각 모델은 동일한 환경에서 학습되었고 모두 ImageNet 데이터셋으로부터 사전 학습된 상태에서 시작하였다. 이 실험을 통해 얻은 결과로부터 다음과 같은 결론을 도출하였다.

ResNet18과 GoogLeNet은 높은 정확도와 비교적 짧은 학습 시간을 통해 CIFAR-10 분류 작업에 적합한 모델로 판단하였다. ResNet18의 잔차 연결 구조와 그레디언트 전파의 효율성이, 그리고

GoogLeNet의 인셉션 모듈과 다양한 구조적 요소들이 해당 데이터셋을 분류하는 데에 핵심적인 역할을 하였다.

반면 AlexNet과 VGG16은 낮은 정확도를 가졌고 VGG16은 특히 매우 긴 학습 시간이 소요되었다. 이는 VGG16의 깊은 층 구조와 많은 파라미터 수로 인해 발생한 것으로 추정되며 작은 이미지 데이터셋에서 과적합이 일어나 성능이 저하되었다. 반면 AlexNet은 구조적으로 단순하고 깊이가 낮은 모델이기 때문에 정확도가 비교적 낮게 나타났다.

결론적으로, 본 연구에서는 GoogLeNet과 ResNet18을 선택하는 것이 CIFAR-10 분류 작업에 있어서 가장 적합한 모델로 판단된다. 두 모델은 높은 정확도와 비교적 짧은 학습 시간을 통해 효율적인 분류 모델로서의 잠재력과 적합성을 보였다.

Ⅴ. 부 록

1. 오픈 소스 라이브러리 출처

https://pytorch.kr/hub/

2. 데이터셋 CIFAR-10 출처

https://www.cs.toronto.edu/~kriz/cifar.html

참 고 문 헌

[1] Elharrouss, Omar et al. “Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches.” ArXiv abs/2206.08016, 2022.

[2] Yann LeCun Leon Bottou Yoshua Bengio and Patrick Haffner, “GradientBased Learning Applied to Document Recognition”, IEEE, Nov. 1998.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6, 2017.

[4] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large—scale Image Recognition”, ICLR, 2015.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions”, IEEE, 2015

[6] Kaiming He, Xiangyu Zhang, Shaoquing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, IEEE, 2016