Network In Network

Table of Contents

Table of Contents
- Abstract
- 1. Introduction
- 2. Convolutional Neural Networks
- 3. Network In Network
  - 3.1 MLP Convolution Layers
  - 3.2 Global Average Pooling
  - 3.3 Network In Network Structure
- 4. Experiments
  - 4.1 Overview
  - 4.2 CIFAR-10
  - 4.3 CIFAR-100
  - 4.4 Street View House Numbers
  - 4.5 MNIST
  - 4.6 Global Average Pooling as a Regularizer
  - 4.7 Visualization of NIN
- 5. Conclusions

Reference

> Network In Network
> Min Lin, Qiang Chen, Shuicheng Yan

Abstract

NIN이라는 새로운 심층 네트워크 구조를 제안.

기존의 CNN은 선형 필터를 사용한 후 비선형 활성화 함수를 사용하여 입력을 스캔하지만,

NIN은 더 복잡한 구조를 가진 마이크로 신경망을 구축하며, 다층 퍼셉트론을 사용하여 마이크로 신경망을 인스턴스화 한다.

NIN은 분류 계층에서 기능 맵에 대한 글로벌 평균 폴링(Global Average Pooling)을 활용하며, 이를 사용해서 기존의 Fully-connected보다 해석하기 쉽고 과적합하기 쉽다.

해당 논문의 검증은 CIFAR-10 및 CIFAR-100에서 SVHN 및 MNIST Dataset에서 성능을 검증했다.

1. Introduction

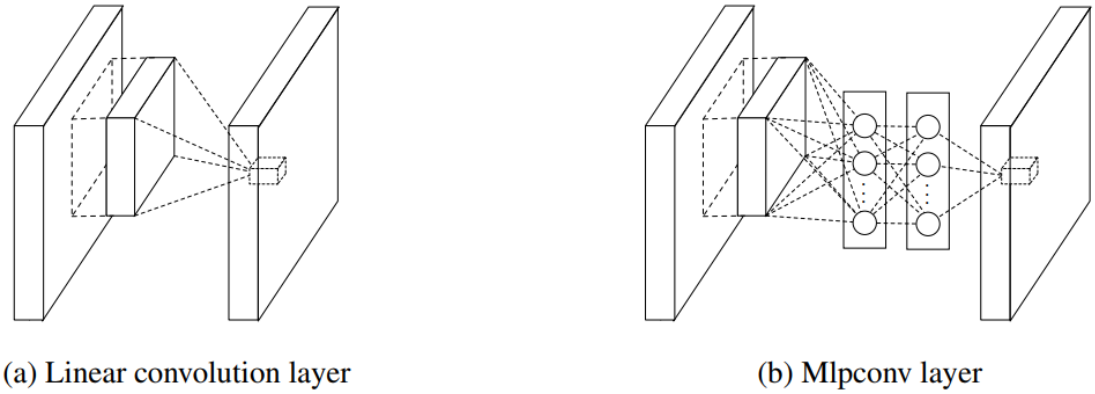


그림1 : CNN와 MLP의 비교.
CNN은 선형 필터를 포함하며, MLP는 마이크로 네트워크를 포함한다.(Multilayer Perceptron)

기존의 CNN은 선형 필터와 기본 수용 필드의 내부 곱을 취한 후 입력의 모든 로컬 부분에서 비선형 활성화 함수를 취한 후, 나온 결과물로 피쳐 맵을 사용한다.

CNN의 convolution filter는 일반화된 선형 모델(GLM)을 사용하며, 해당 논문에서는 GLM의 추상화 수준이 낮다고 주장하고 있으며, 추상화로 인해 동일한 개념의 변형이 불변함을 의미하기 때문에 NIN에서는 GLM보다 강력한 비선형 함수 근사치로 대체하여 모델의 추상화 능력을 향상시킬 수 있다고 주장한다.

GLM은 잠재 개념의 샘플이 선형적으로 분리될 수 있을 때, 예를 들어 yes or no 처럼 하나의 구분선을 두고 온전히 분리가 가능할 때에 상당한 정도의 추상화를 달성할 수 있다.

따라서 기존의 CNN은 잠재 개념이 선형적으로 분리 가능하다는 전제를 놓고 암시적으로 만든다.

하지만, 동일한 개념에 대한 데이터는 종종 비선형 manifold에 존재 한다.

NIN에서의 GLM은 일반적인 비선형 함수 근사치인 "마이크로 네트워크" 구조로 대체된다.

해당 논문에서는 마이크로 네트워크의 인스턴스화로 Multilayer Perceptron을 선택하며, 이는 범용 함수 근사치이며 역전파에 의해 훈련될 수 있는 신경망이다.

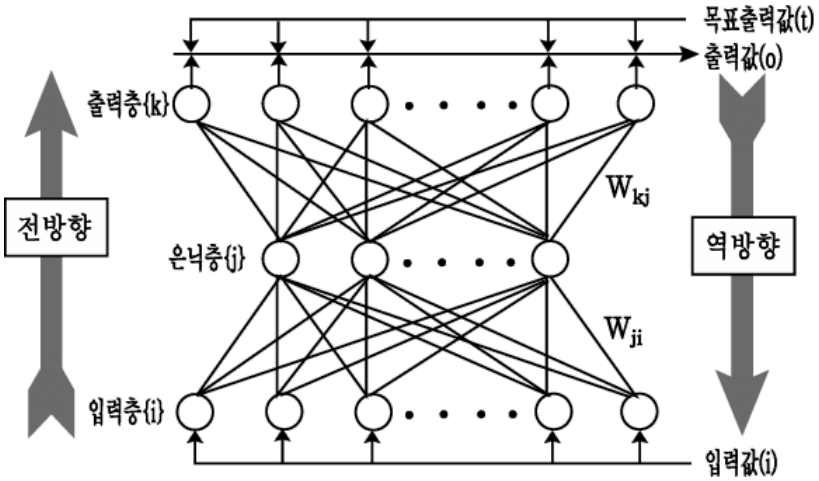


그림2: Multilayer Perceptron의 구조

Linear convolution layer과 mlpconv layer 모두 출력 특징 벡터에 매핑한다.

mlpconv는 비선형 활성화 기능을 가진 여러 개의 Fully-connected layer로 구성된 Multilayer perceptron(MLP)을 사용하여 입력 로컬 패치를 출력 피쳐 벡터에 매핑한다.

MLP는 모든 로컬 수신 필드 간의 공유된다.

기능 맵은 CNN과 유사한 방식으로 진행되며, NIN은 여러 mlpconv layer의 적층이다.

전체 심층 네트워크의 요소를 구성하는 MLP가 mlpconv layer 내에 있기 때문에 "NIN"이라고 불리며,
CNN의 마지막의 Fully-connected layer을 사용하는 대신 마지막 mlpconv layer의 공간 평균을 c의 신뢰로 직접 출력한다.

기존 CNN에서는 블랙박스 역할을 하는 Fully-connected layer때문에 목표 비용 계층의 범주 수준 정보가 이전 컨볼루션 계층으로 다시 전달되는 방식을 해석하기 어렵지만, 전역 평균 폴링은 마이크로 네트워크를 사용한 강력한 로컬 모델링으로 인해 가능해진 특징 맵과 범주 간의 대응을 시행하기 때문에 더 의미 있게 해석이 가능하다.

또한 완전히 연결된 계층은 과적합되기 쉽고 드롭아웃 정규화에 크게 의존하는 반면, 전역 평균 풀링은 전체 구조에 대한 과적합을 기본적으로 방지하는 구조 정규화 그 자체이다.

요약 : 기존 CNN에서는 Filter를 이용하여 Stride만큼 이동하면서 Convolution으로 Feature를 추출했다. NIN에서도 유사하게 진행되는데 Filter 대신에 MLP를 쓰는 부분만 다르다고 할 수 있고, 이를 Mlpconv layer라고 부른다.

MLP를 이용했을 때 장점은 Filter를 사용할 때보다 Non-linear한 Activation function을 더 추가하여 Non-linear한 성질로 인해 더 좋은 Feature를 추출 할 수 있다는 점이다. 또한, 1×1 Convolution 을 통해 Feature map 개수를 줄임으로써 Parameter 수를 줄일 수 있었다 (아래에서 다시 설명).

2. Convolutional Neural Networks

Linear convolution은 잠재 개념의 인스턴스들이 선형적으로 분리될 때 추상화에 충분하지만, 우수한 추상화를 달성하는 표현은 일반적으로 입력 데이터의 비선형 함수가 높다.

개별 선형 필터는 동일한 개념의 다른 변화를 감지하는 방법을 학습할 수 있자만, 단일 개념에 대한 필터가 너무 많으면 이전 계층으로부터의 모든 변동 조합을 고려해야 하는 다음 계층에는 추가 부담이 발생한다.

각 로컬 패치를 더 높은 수준의 개념으로 결합하기 전에 더 나은 추상화를 수행하는 것이 유익할 것이라고 본 논문에서는 주장하며, 그 이유는 아래 계층에서 하위 수준 개념을 결합하여 더 나은 상위 수준 개념을 생성할 수 있기 때문이다.

모든 볼록 함수를 근사화할 수 있는 조각별 선형 근사치를 만들며, 선형 분리를 수행하는 기존의 convolution layer에 비해, 최대 출력 네트워크는 볼록 세트 내에 있는 개념을 분리할 수 있기 때문에 더 강력하다.

잠재된 개념의 분포가 더 복잡한 경우 더 일반적인 함수 근사치를 사용할 필요가 있을 것이다.
우리는 로컬 패치에 대한 보다 추상적인 특징을 계산하기 위해 각 컨볼루션 계층 내에 마이크로 네트워크가 도입되는 새로운 "네트워크 내 네트워크" 구조를 도입하여 이를 달성하고자 한다.

3. Network In Network

3.1절은 MLP convolution layer계층과 3.2절은 Global Average Pooling을 강조하며,

3.3절에서 전체적인 NIN의 구조를 설명합니다.

3.1 MLP Convolution Layers

잠재 개념의 더 추상적인 표현을 근사화할 수 있기 때문에 로컬 패치의 특징 추출을 위해 범용 함수 근사치를 사용하는 것이 바람직하다.

방사형 기본 네트워크와 다층 퍼셉트론은 잘 알려진 두 개의 범용 함수 근사치이다.

우리는 두 가지 이유로 이 연구에서 다층 퍼셉트론을 선택한다.

첫째, 다층 퍼셉트론은 역 전파를 사용하여 훈련된 컨볼루션 신경망 구조와 호환된다.

둘째, 다층 퍼셉트론은 기능 재사용 정신과 일치하는 심층 모델 그 자체일 수 있다.

새로운 유형의 레이어는 본 논문에서 MLP가 GLM을 대체하여 입력을 융합하는 mlpconv라고 한다.

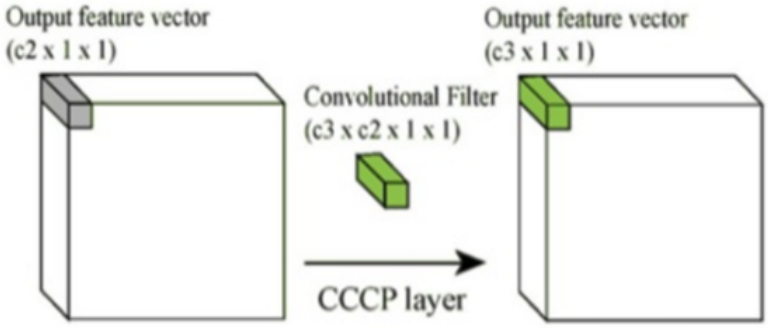
$$
\begin{aligned}
f_{i,j,k_1}^{1} &= \max(w_{k_1}^{1\,T} x_{i,j} + b_{k_1}, 0). \\
&\vdots \\
f_{i,j,k_n}^{n} &= \max(w_{k_n}^{n\,T} f_{i,j}^{n-1} + b_{k_n}, 0). \qquad (2)
\end{aligned}
$$

그림3: MLPconv layer에 의해 수행되는 계산

교차 채널 매개 변수 풀링 계층은 또한 1×1 컨볼루션 커널을 가진 컨볼루션 계층과 동일하다.
이 해석은 NIN의 구조를 이해하기에 직설적이다.

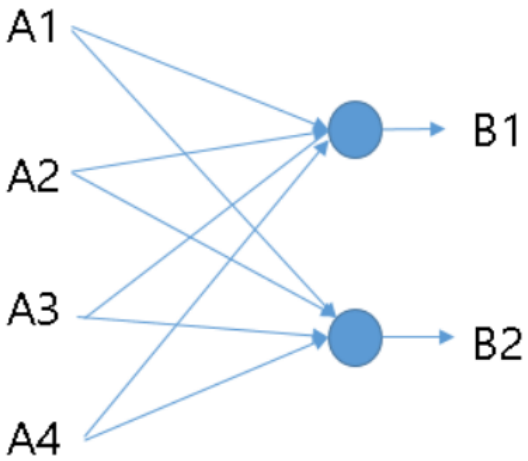
1×1 Convolution을 사용하는 가장 큰 이유는 차원을 줄이는 것 이다

1×1 Convolution을 사용하면 여러 개의 Feature map에서 비슷한 성질을 추출하여 Feature map 크기를 줄일 수 있다. Feature map 크기가 줄어들면 연산량이 줄어 Network를 더 깊게 만들 수 있다. 아래 그림으로 1×1 Convolution이 어떻게 동작하는지 알 수 있다.



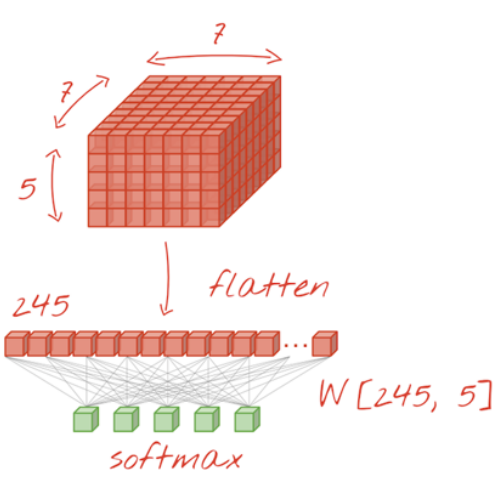
위 그림처럼 “c2 > c3”의 관계를 만들면 함축적인 의미를 가지는 더 작은 Feature map을 얻을 수 있고, 이는 차원의 감소로 이어진다.

1×1 Convolution을 조금 더 직관적으로 이해해보자. 1×1 Convolution은 1-layer fully connected neural network 라고도 하는데 그 이유는 같은 원리로 동작하기 때문이다. 아래에 예를 통해 살펴보자.

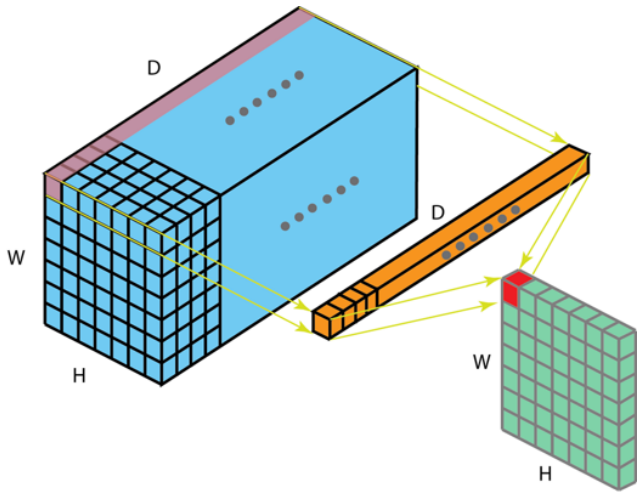


위 그림은 A1, A2, A3, A4 Feature map을 B1, B2 Feature map으로 1×1 Convolution한 예시이다.
이는 마치 4개의 Neuron을 2개의 Neuron으로 Fully connected한 경우와 유사한 형태를 띈다.
NIN에서는 이러한 과정을 통해 차원을 감소시키고 연산량을 줄일 수 있었다.
여기에 활성함수 (예: ReLU)를 추가하면 Non-linear한 성질도 추가 로 얻을 수 있다.

이번에는 실제 Feature map을 1×1 Convolution 하는 과정을 살펴보면서 이해도를 높여보자.



Fully connected layer



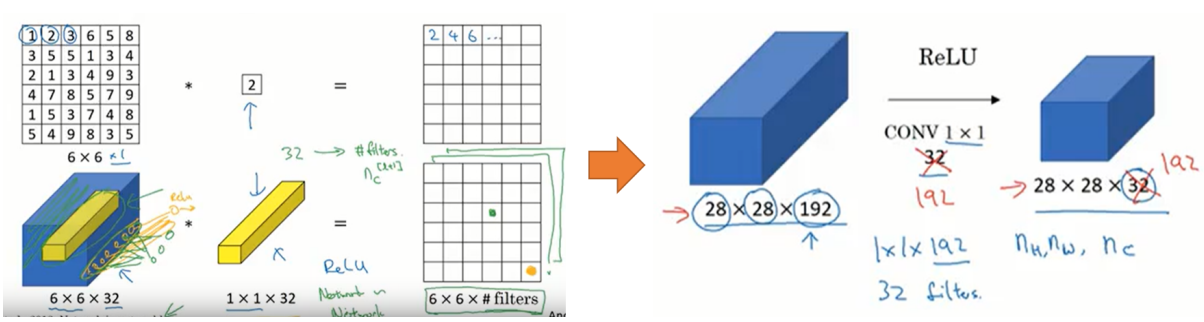
1x1 Convolution

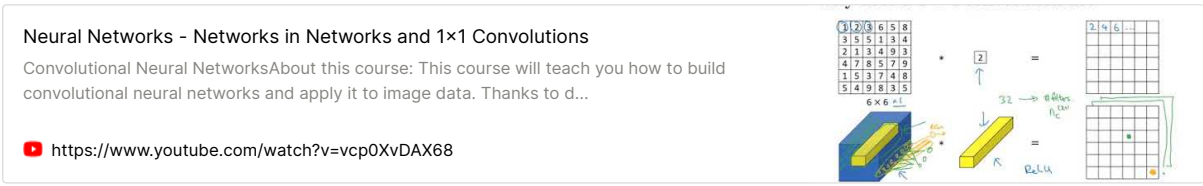
먼저 위 좌측 그림은 Feature map을 Flatten 한 후, FC layer의 모든 노드와 연결한 그림이다.
이 때, Feature map의 모든 값 하나하나가 각기 다른 Weight 값과 곱해져 FC layer로 전달된다.
이러한 과정 자체가 Fully connected 하게 연결되는 상황이고, 그래서 Fully connected layer라고 부른다.

1×1 Convolution도 그 과정이 유사하다. 위 우측 그림은 Feature map의 좌측 상단 값들 하나하나와 1×1 Convolution의 각 Filter 값 (FC layer의 Weight에 해당) 하나하나가 곱해져 새로운 Feature map이 만들어지는 과정이다.

결국 **Feature map의 모든 값들이 1×1 Convolution 내 각 값들과 곱해져 새로운 Feature map으로 전달되는 과정이 Fully connected 하게 연결** 되므로 FC layer가 생성되는 과정과 유사하다고 할 수 있다.

deeplearning.ai에서는 아래와 같이 설명한다.





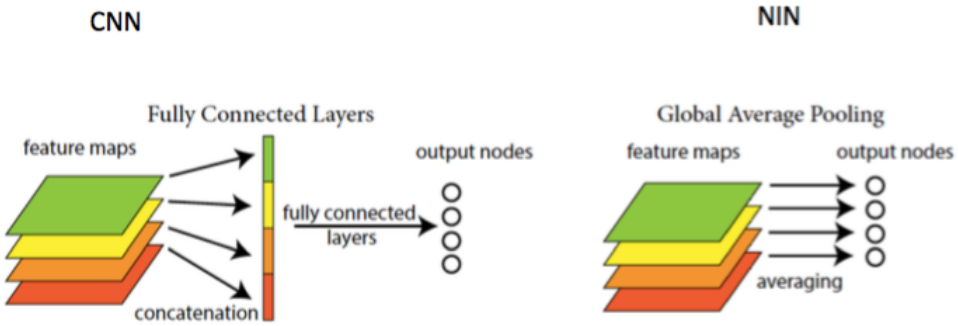
여기서도 마찬가지로 1×1 Convolution의 각 값들이 이전 Feature map의 값들과 곱해져 동일한 크기의 새로운 Feature map을 생성한다고 설명하고 있다.
결국 1×1 Convolution을 n개 사용하면 이전 Feature map의 크기와 동일한 Feature map을 n개 생성한므로, Feature map의 개수를 늘리거나 줄이는 활용할 수 있게 된다.

3.2 Global Average Pooling

NIN과 기존 CNN의 또 다른 점은 Fully connected layer (FC layer)가 없다는 점이다. NIN에서는 FC layer 대신에 Global Average Pooling (GAP)을 사용 했다. Lin은 앞에서 충분히 효과적인 Feature를 추출했고, Average pooling만으로 Classifier 역할을 할 수 있다고 주장했다. 또한 이를 통해 Overfitting과 연산량을 줄이는 효과가 있다.

실제로 FC layer의 Parameter 수는 전체의 90%에 가깝기 때문에 Overfitting 문제가 발생 하기 쉽다. 그래서 보통 Dropout을 사용하는데, NIN에서는 Average pooling 결과로 Classification을 할 수 있기 때문에 이러한 문제를 해결할 수 있게 된다

(GoogLeNet도 GAP 사용).



- **마지막 mlpconv 레이어에서 classification 해야할 만큼의 feature map 생성.**
- **feature map에서 average를 계산해 바로 soft layer에 전달.**
- **파라미터수가 감소하며, 오버피팅 방지.**

3.3 Network In Network Structure

NIN의 전체적인 구조는 mlpconv 레이어의 스택이며, 그 위에 글로벌 평균 풀링과 목표 비용 레이어가 있다.

그림 2는 세 개의 mlp conv 레이어가 있는 NIN을 보여준다.
각 mlpconv 계층에는 3계층 퍼셉트론이 있다.
NIN과 마이크로 네트워크 모두에서 계층 수는 유동적이며 특정 작업에 맞게 조정될 수 있다.

4. Experiments

4.1 Overview

CIFAR-10[12], CIFAR-100[12], SVHN[13] 및 MNIST[1]의 네 가지 벤치마크 데이터 세트에서 NIN을 평가한다.

데이터 세트에 사용되는 네트워크는 모두 3개의 스택형 mlpconv 레이어로 구성되며, 모든 실험에서 mlpconv 레이어는 입력 이미지를 2배로 다운 샘플링하는 공간 최대 풀링 레이어로 이어진다.

네트워크는 128 크기의 미니 배치를 사용하여 훈련된다.
훈련 과정은 초기 가중치와 학습 속도에서 시작하여 훈련 세트의 정확도가 향상되는 것을 멈출 때까지 계속되며, 그 다음 학습 속도가 10의 척도로 낮아진다.

4.2 CIFAR-10

CIFAR-10 데이터 세트[12]는 총 50,000개의 훈련 영상과 10,000개의 테스트 영상으로 구성된 10개 클래스의 자연 영상으로 구성된다.

각 이미지는 32×32 크기의 RGB 이미지입니다.
이 데이터 세트의 경우 Goodfellow 등이 사용한 것과 동일한 글로벌 대비 정규화와 ZCA 화이트닝을 적용한다.
최대 출력 네트워크[8]에 있습니다.
우리는 교육 세트의 마지막 10,000개의 이미지를 검증 데이터로 사용한다.
이 실험에서 각 mlpconv 계층에 대한 형상 맵의 수는 해당 maxout 네트워크에서와 동일한 수로 설정된다.
두 개의 하이퍼 파라미터는 유효성 검사 세트를 사용하여 조정됩니다.

우리는 이 데이터 세트에서 10.41%의 테스트 오류를 얻는데, 이는 최첨단 데이터 세트에 비해 1% 이상 향상된다.
이전 방법과의 비교는 표 1과 같다.

Method	Test Error
Stochastic Pooling [11]	15.13%
CNN + Spearmint [14]	14.98%
Conv. maxout + Dropout [8]	11.68%
NIN + Dropout	**10.41%**
CNN + Spearmint + Data Augmentation [14]	9.50%
Conv. maxout + Dropout + Data Augmentation [8]	9.38%
DropConnect + 12 networks + Data Augmentation [15]	9.32%
NIN + Dropout + Data Augmentation	**8.81%**

표 1: 다양한 방법의 CIFAR-10에 대한 오류율을 테스트합니다.

4.3 CIFAR-100

CIFAR-100 데이터 세트[12]는 CIFAR-10 데이터 세트와 크기와 형식이 동일하지만 100개의 클래스를 포함하고 있다. 따라서 각 클래스의 이미지 수는 CIFAR-10 데이터 세트의 10분의 1에 불과하다.

CIFAR-100의 경우 하이퍼 파라미터를 튜닝하지 않고 CIFAR-10 데이터 세트와 동일한 설정을 사용한다.
유일한 차이점은 마지막 mlpconv 계층이 100개의 기능 맵을 출력한다는 것이다.

Method	Test Error
Learned Pooling [16]	43.71%
Stochastic Pooling [11]	42.51%
Conv. maxout + Dropout [8]	38.57%
Tree based priors [17]	36.85%
NIN + Dropout	**35.68%**

표 2: 다양한 방법의 CIFAR-100에 대한 오류율을 테스트합니다.

4.4 Street View House Numbers

SVHN 데이터 세트[13]는 630,420 32×32 컬러 이미지로 구성되며, 훈련 세트, 테스트 세트 및 추가 세트로 나뉜다.
이 데이터 세트의 작업은 각 이미지의 중앙에 위치한 숫자를 분류하는 것입니다.

SVHN에서 사용되는 구조와 매개변수는 CIFAR-10에 사용되는 것과 유사하며,
이는 3개의 mlp conv 레이어와 글로벌 평균 풀링으로 구성된다.

Method	Test Error
Stochastic Pooling [11]	2.80%
Rectifier + Dropout [18]	2.78%
Rectifier + Dropout + Synthetic Translation [18]	2.68%
Conv. maxout + Dropout [8]	2.47%
NIN + Dropout	2.35%
Multi-digit Number Recognition [19]	2.16%
DropConnect [15]	**1.94%**

표 3: 다양한 방법의 SVHN에 대한 오류율을 테스트합니다.

4.5 MNIST

MNIST [1] 데이터 세트는 크기가 28×28인 손으로 쓴 숫자 0-9로 구성된다. 총 60,000개의 훈련 이미지와 10,000개의 테스트 이미지가 있습니다.
이 데이터 세트의 경우 CIFAR-10에 사용된 것과 동일한 네트워크 구조가 채택된다.

그러나 각 mlpconv 계층에서 생성된 기능 맵의 수는 감소한다.
MNIST는 CIFAR-10에 비해 더 간단한 데이터 세트이기 때문에, 더 적은 수의 파라미터가 필요하다.
우리는 데이터 확대 없이 이 데이터 세트에서 우리의 방법을 테스트한다.

Method	Test Error
2-Layer CNN + 2-Layer NN [11]	0.53%
Stochastic Pooling [11]	0.47%
NIN + Dropout	0.47%
Conv. maxout + Dropout [8]	**0.45%**

표 4: 다양한 방법의 MNIST에 대한 오류율을 테스트합니다.

MNIST가 매우 낮은 오류율로 조정되었기 때문에 현재 최고(0.45%)와 비교할 수는 있지만 더 나은 성능(0.47%)을 달성한다.

4.6 Global Average Pooling as a Regularizer

완전히 연결된 계층은 고밀도 변환 매트릭스를 가질 수 있으며 값은 역 전파 최적화의 영향을 받는다.
글로벌 평균 풀링의 정규화 효과를 연구하기 위해, 우리는 모델의 다른 부분은 그대로 유지하면서 글로벌 평균 풀링 계층을 완전히 연결된 계층으로 교체한다.
우리는 완전히 연결된 선형 레이어 이전의 드롭아웃과 없이 이 모델을 평가했다.

Method	Testing Error
mlpconv + Fully Connected	11.59%
mlpconv + Fully Connected + Dropout	10.88%
mlpconv + Global Average Pooling	10.41%

표 5: 완전히 연결된 계층과 비교한 글로벌 평균 풀링.

표 5에서 보듯이, 드롭아웃 정규화 없이 완전히 연결된 계층은 최악의 성능을 보였다(11.59%)

이는 정규화기가 적용되지 않는 경우 완전히 연결된 계층이 교육 데이터에 오버핏될 것으로 예상된다.
완전히 연결된 계층 이전에 드롭아웃을 추가하면 테스트 오류(10.88%)가 감소했습니다.
글로벌 평균 풀링은 세 가지 테스트 오류 중 가장 낮은 테스트 오류(10.41%)를 달성했습니다.

성능은 CIFAR-10 데이터 세트에서 테스트되었다.

비교를 공정하게 하기 위해, 우리는 글로벌 평균 풀링 체계의 각 범주에 대해 하나의 피쳐 맵만 허용되기 때문에 로컬 연결 계층의 피쳐 맵 수를 16개에서 10개로 줄인다.
그런 다음 드롭아웃 + 완전히 연결된 계층을 글로벌 평균 풀링으로 대체하여 글로벌 평균 풀링을 가진 동등한 네트워크를 생성한다.

비교를 공정하게 하기 위해, 우리는 글로벌 평균 풀링 체계의 각 범주에 대해 하나의 피쳐 맵만 허용되기 때문에 로컬 연결 계층의 피쳐 맵 수를 16개에서 10개로 줄인다.
그런 다음 드롭아웃 + 완전히 연결된 계층을 글로벌 평균 풀링으로 대체하여 글로벌 평균 풀링을 가진 동등한 네트워크를 생성한다.

4.7 Visualization of NIN

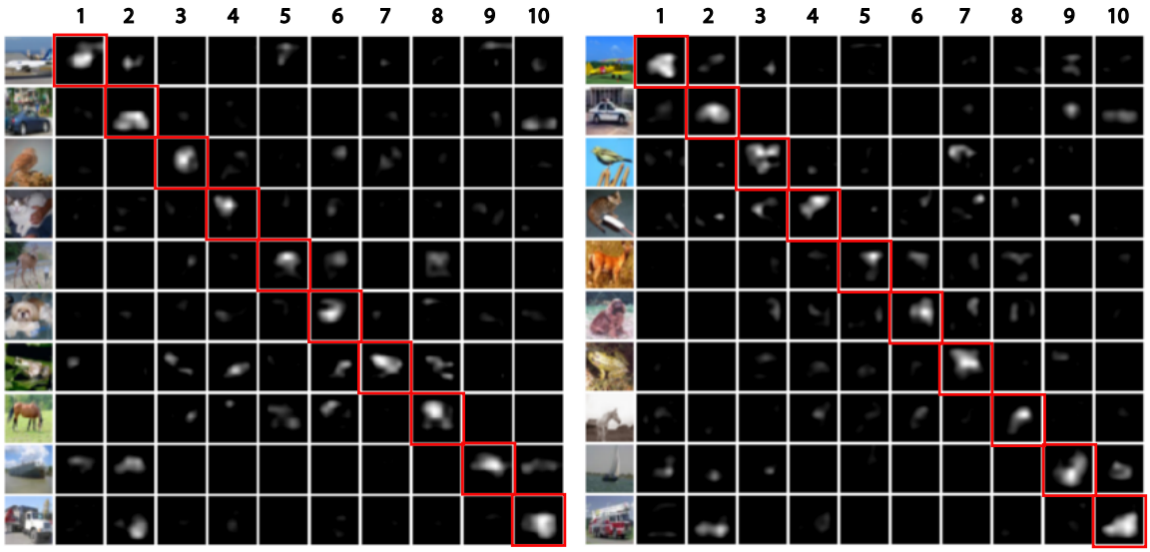


그림4

CIFAR-10에 대해 훈련된 모델의 마지막 mlpconv 계층에서 기능 맵을 추출하고 직접 시각화한다.

가장 강력한 활성화는 원본 이미지에서 객체의 동일한 영역에서 대략적으로 나타나는 것을 관찰할 수 있다.

두 번째 열에 있는 자동차와 같은 구조화된 물체에 특히 해당된다.

그런 다음 글로벌 평균 풀링은 범주 수준 피쳐 맵 학습을 시행합니다. 일반적인 물체 탐지에 대해 추가 탐사를 할 수 있다.
파라벳 등의 장면 라벨링 작업에서와 동일한 맛의 범주 수준 특징 맵을 기반으로 탐지 결과를 얻을 수 있다. [20].

5. Conclusions

우리는 분류 작업을 위해 "네트워크 내 네트워크"(NIN)라는 새로운 심층 네트워크를 제안했다.
이 새로운 구조는 기존의 CNN에서 완전히 연결된 계층을 대체하기 위해 다층 퍼셉트론을 사용하는 mlpconv 레이어와 글로벌 평균 풀링 레이어로 구성된다.

Mlpconv 레이어는 로컬 패치를 더 잘 모델링하며, 글로벌 평균 풀링은 전체적으로 과적합을 방지하는 구조적 정규화 역할을 한다.

우리는 NIN의 마지막 mlpconv 계층의 기능 맵이 범주의 신뢰 맵이라는 것을 입증했고, 이는 NIN을 통해 객체 탐지를 수행할 수 있는 가능성을 유발한다.

Reference

7.3. Network in Network (NiN) - Dive into Deep Learning 0.16.2 documentation

LeNet, AlexNet, and VGG all share a common design pattern: extract features exploiting spatial structure via a sequence of convolution and pooling layers and then post-process the representations via fully-connected layers. The improvements upon LeNet by AlexNet and VGG mainly lie in how these later networks widen and deepen these two modules.

https://d2l.ai/chapter_convolutional-modern/nin.html

Network in Network - Organize everything I know documentation

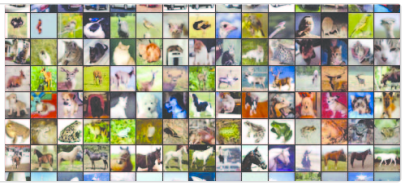
Network in Network (NIN)은 싱가포르 국립대학의 Min Lin이 2013년에 발표한 모델이다. Lin은 CNN의 Convolution layer가 Local receptive field에서 Feature를 추출할 때 Filter로 계산하여 Linear한 문제 를 해결하려고 했다. 기존에는 Feature map을 늘려서 이러한 문제를 극복하려고 했지만 Filter가 늘어남에 따라 연산량이 늘어나는 문제가 있었다.

https://oi.readthedocs.io/en/latest/computer_vision/cnn/nin.html

Review: NIN - Network In Network (Image Classification)

In this story, Network In Network (NIN), by Graduate School for Integrative Sciences and Engineering and National University of Singapore, is briefly reviewed. Micro neural networks with more than...

https://towardsdatascience.com/review-nin-network-in-network-image-classification-69e2

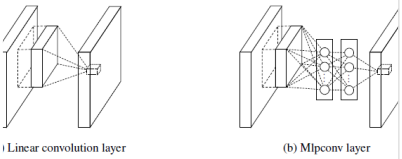


71e499ee

Network In Network

GoogLeNet의 기초가된 모델. 개요 Convolution Layer 대신 MLP Convolution Layer 사용. Fully Connected 대신 Global Average Pooling 사용. * MLP Conv 사용으로 기존의 CNN 보다 성능이 뛰어나다. 장점 Convol..

https://mjdeeplearning.tistory.com/9



Network In Network and 1×1 Convolutions

gaussian37's blog

https://gaussian37.github.io/dl-dlai-network_in_network/