

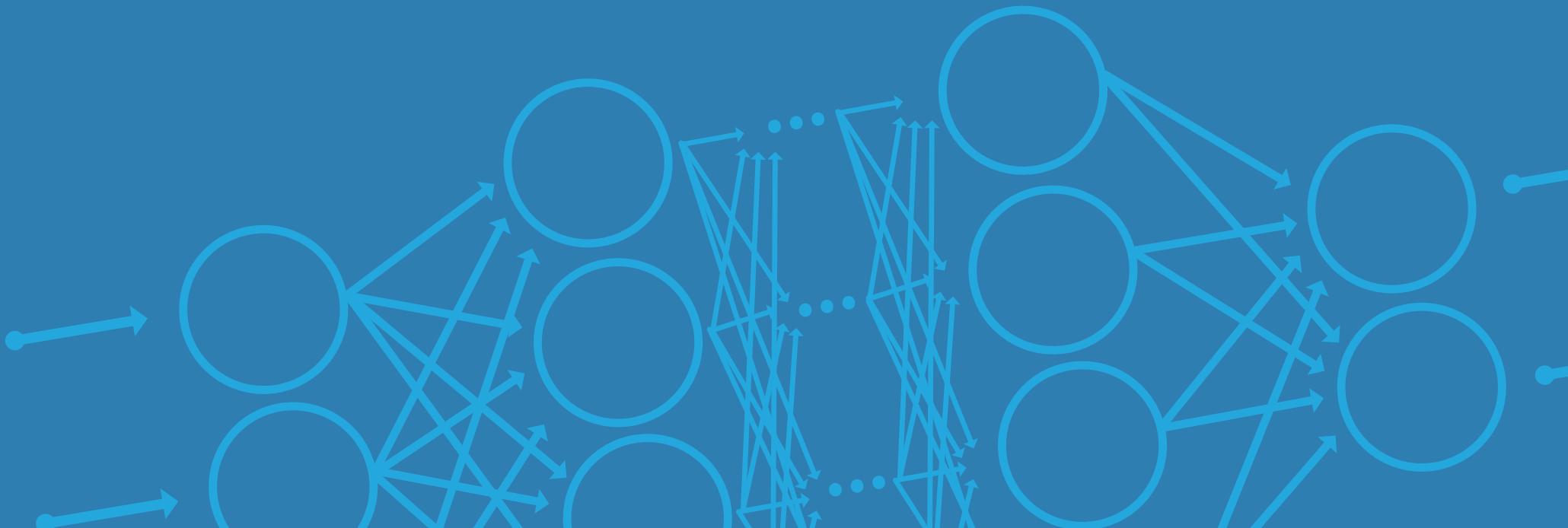


MATLAB을 활용한 딥러닝



딥러닝이란?

딥러닝은 머신 러닝의 한 유형으로, 모델이 직접 이미지, 텍스트 또는 사운드로부터 분류 작업을 학습합니다. 일반적으로 딥러닝은 신경망 구조를 사용하여 구현됩니다. “딥(deep)”이라는 용어는 네트워크의 계층 수를 나타냅니다. 계층이 많을수록 네트워크가 깊어집니다. 기존의 신경망에는 2계층 또는 3계층만이 포함되었지만 심층 신경망에는 수백 개의 계층이 있을 수 있습니다.



딥러닝 응용프로그램

다음의 몇 가지 딥러닝 예제를 살펴보겠습니다.

- 자율주행 자동차는 보행자가 횡단보도를 건너는 것을 감지하여 속도를 줄입니다.
- ATM이 위치 지폐를 인식하여 거부합니다.
- 스마트폰 앱을 사용하여 외국어로 된 표지판을 즉시 번역할 수 있습니다.

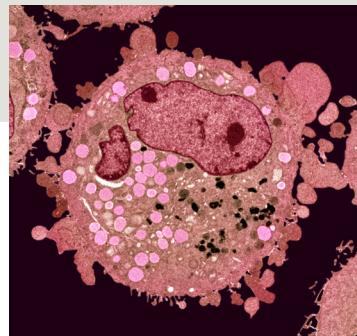
딥러닝은 특히 얼굴 인식, 텍스트 변환, 음성 인식과 차선 분류 및 교통 표지 인식을 포함한 고급 운전자 지원 시스템 (ADAS) 등의 식별 응용프로그램에 적합합니다.



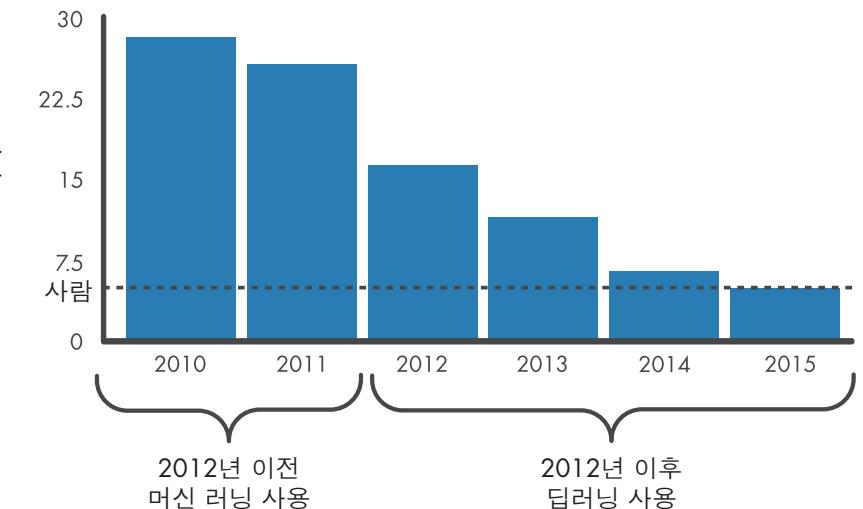
딥러닝이 최첨단 기술인 이유는 무엇일까요?

바로 정확성 때문입니다. 고급 툴 및 기술을 통해 딥러닝 알고리즘이 크게 개선되어 이미지 분류에 있어서 인간을 능가하거나, 세계 최고의 바둑 기사를 이기기도 하며, 좋아하는 신곡을 찾아서 다운로드할 수 있는 Amazon Echo® 및 Google Home과 같은 음성 비서를 사용할 수 있을 정도가 되었습니다.

UCLA 연구진들은 고차원 데이터 세트를 산출하는 최신 현미경을 개발했고, 이를 조직 표본에서 암세포를 식별하는 딥러닝 네트워크를 학습하는데 사용했습니다.



ILSVRC의 IMAGENET에 대한 상위 5개 오류

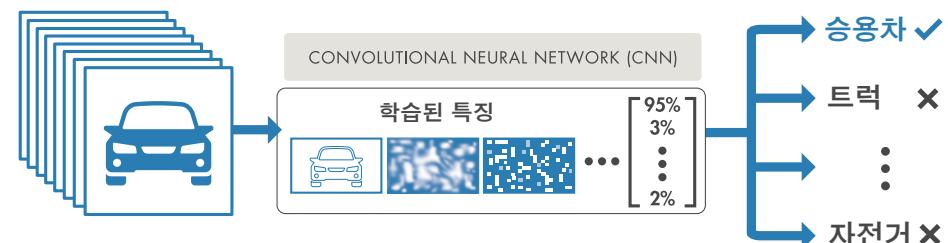


딥러닝이 최첨단 기술인 이유는 무엇일까요?

계속

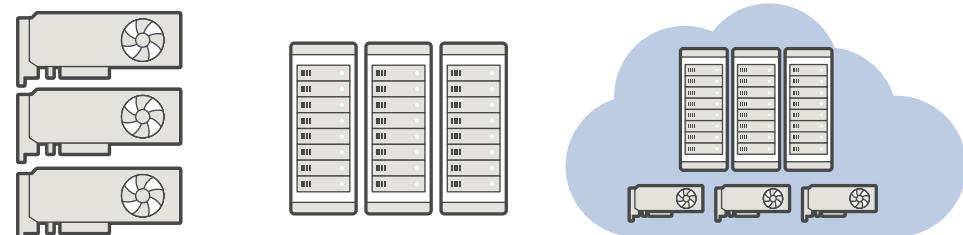
딥러닝은 세 가지 기술적 요소로 인해 높은 정확성을 갖추게 됩니다.

레이블이 지정된 대량의 데이터 집합에 간편하게 액세스
ImageNet 및 PASCAL VoC 등의 데이터 세트가 무료로 사용 가능하며 다양한 유형의 객체 학습에 유용합니다.



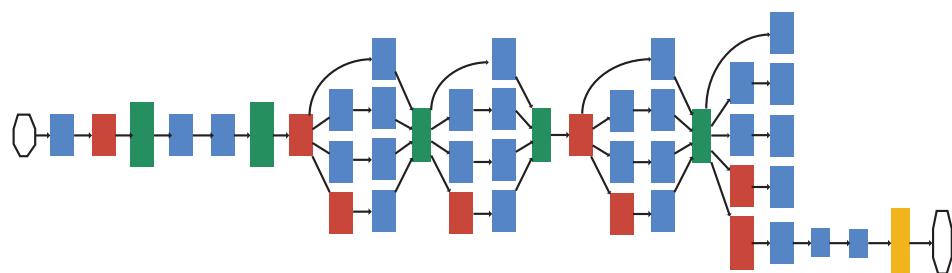
향상된 연산 능력

고성능 GPU를 사용하여 딥러닝에 필요한 대량의 데이터를 빠르게 학습할 수 있으므로 학습 시간이 몇 주에서 몇 시간으로 단축됩니다.



전문가가 구축한 사전 학습된 모델

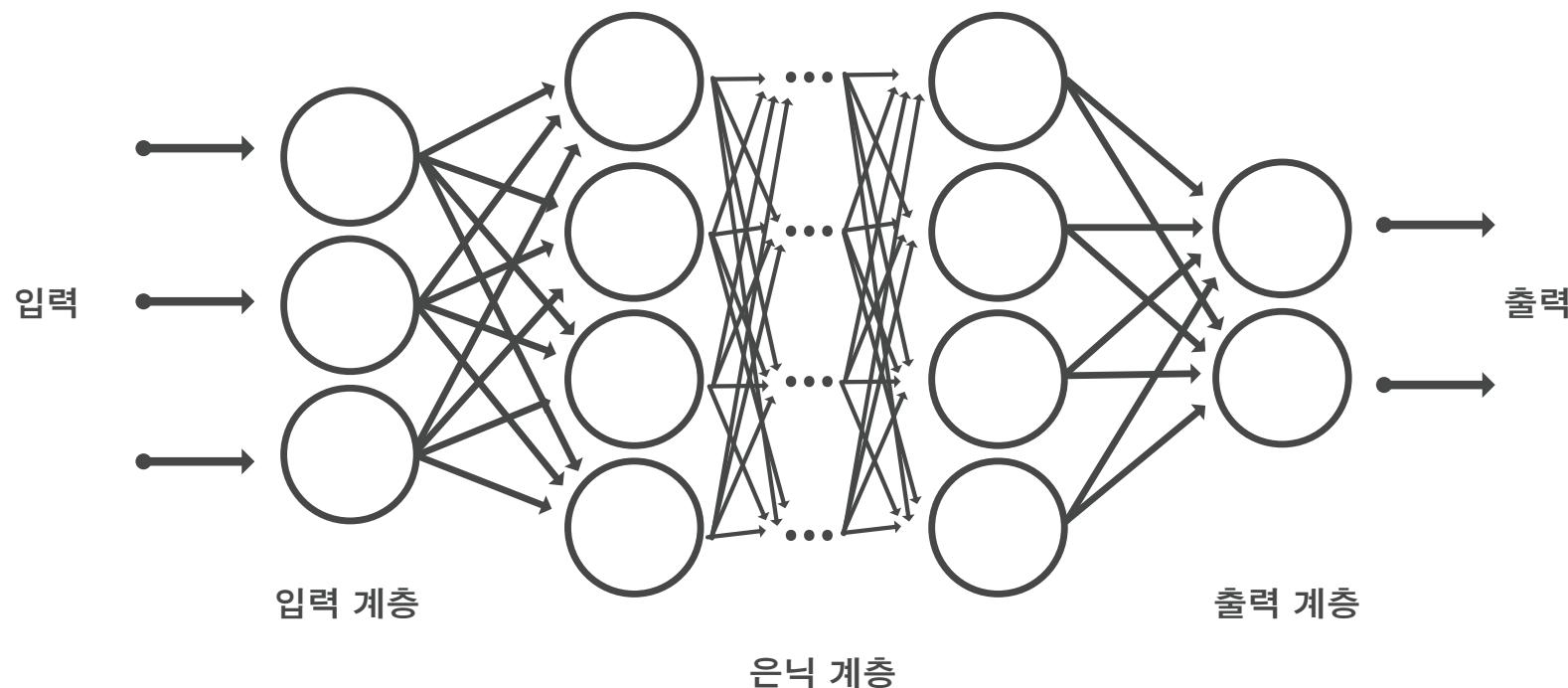
전이 학습이라는 기술을 사용하여 새로운 인식 작업을 수행하도록 AlexNet 등의 모델을 재학습할 수 있습니다. AlexNet은 1000개의 서로 다른 객체를 인식하기 위해 130만 개의 고해상도 이미지를 학습해야 했지만, 그보다 훨씬 더 작은 데이터 세트로 정확한 전이 학습을 달성할 수 있습니다.



심층 신경망 내부

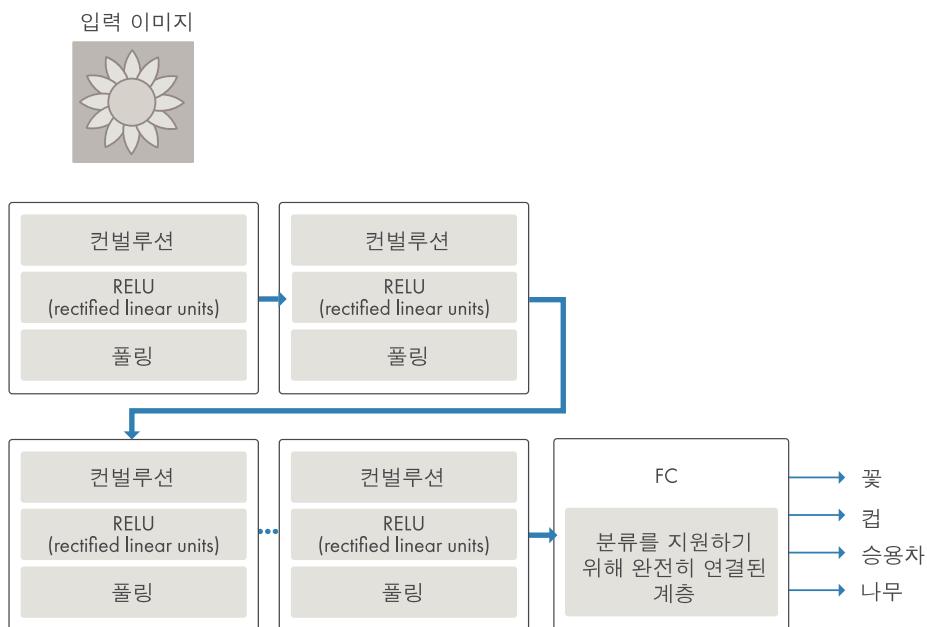
심층 신경망은 병렬로 작동하고 생물학적 신경계에서 영감을 받은 단순 요소를 사용하여 다중 비선형 처리 계층을 결합합니다.

심층 신경망은 입력 계층, 여러 개의 은닉 계층 및 출력 계층으로 구성됩니다. 계층은 이전 계층의 출력을 입력으로 사용하여 노드 또는 뉴런을 통해 각각의 은닉 계층과 상호 연결됩니다.



심층 신경망의 학습 방법

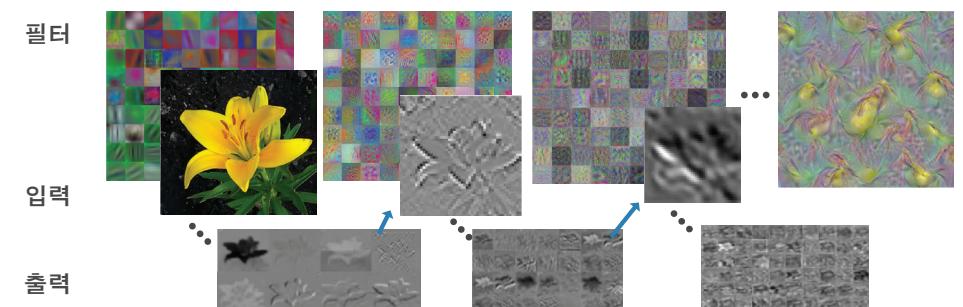
어떤 이미지 집합 내의 각 이미지가 4가지 서로 다른 객체 중 하나를 포함하며, 딥러닝 네트워크를 통해 그 객체가 무엇인지를 자동으로 인식하는 상황을 가정합니다. 이때 학습 데이터를 네트워크에 사용하기 위해서는 이미지에 레이블을 지정해야 합니다.



이러한 학습 데이터를 통해 네트워크는 객체의 고유한 특성을 이해하고, 그 특징을 해당 범주와 연결할 수 있습니다.

네트워크의 각 계층은 이전 계층에서 데이터를 가져와서 변환한 후 전달합니다. 네트워크는 계층에서 계층으로 학습하는 내용의 복잡성과 세부 정보를 증가시킵니다.

네트워크는 데이터로부터 직접 학습한다는 것에 유의해야 합니다. 어떤 특징들을 학습할 지 영향을 끼칠 수 없습니다.



CNN(Convolutional Neural Network) 정보

CNN(Convolutional Neural Network 또는 ConvNet)은 이미지 및 비디오를 사용한 딥러닝에 가장 많이 사용하는 알고리즘 중 하나입니다.

CNN은 다른 신경망과 마찬가지로 입력 계층, 출력 계층 및 두 계층 사이의 여러 은닉 계층으로 구성됩니다.

특징 검출 계층

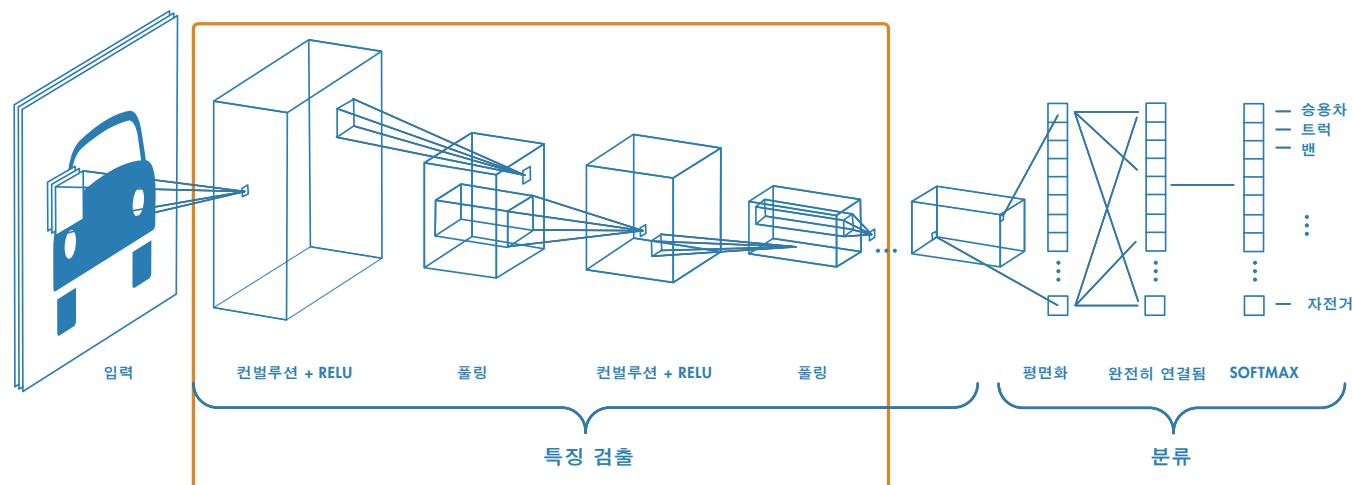
이러한 계층은 데이터에 대해 세 가지 유형의 작업, 즉 컨벌루션, 풀링 또는 ReLU(Rectified Linear Unit) 중 하나를 수행합니다.

컨벌루션은 입력 이미지를 컨볼루션 필터 집합에 통과시키고, 각 필터는 이미지에서 특정 특징을 활성화 시킵니다.

풀링은 비선형 다운샘플링을 수행함으로써 출력을 간소화하여, 네트워크가 학습해야하는 파라미터의 수를 줄입니다.

ReLU(Rectified Linear Unit)는 음수 값을 0에 매핑하고 양수 값을 유지하여 더 빠르고 효과적인 학습을 가능하게 합니다.

이러한 세 가지 작업이 수십 개 또는 수백 개의 계층에서 반복되어 각 계층은 서로 다른 특징들을 검출하도록 학습됩니다.



CNN(Convolutional Neural Network) 정보 계속

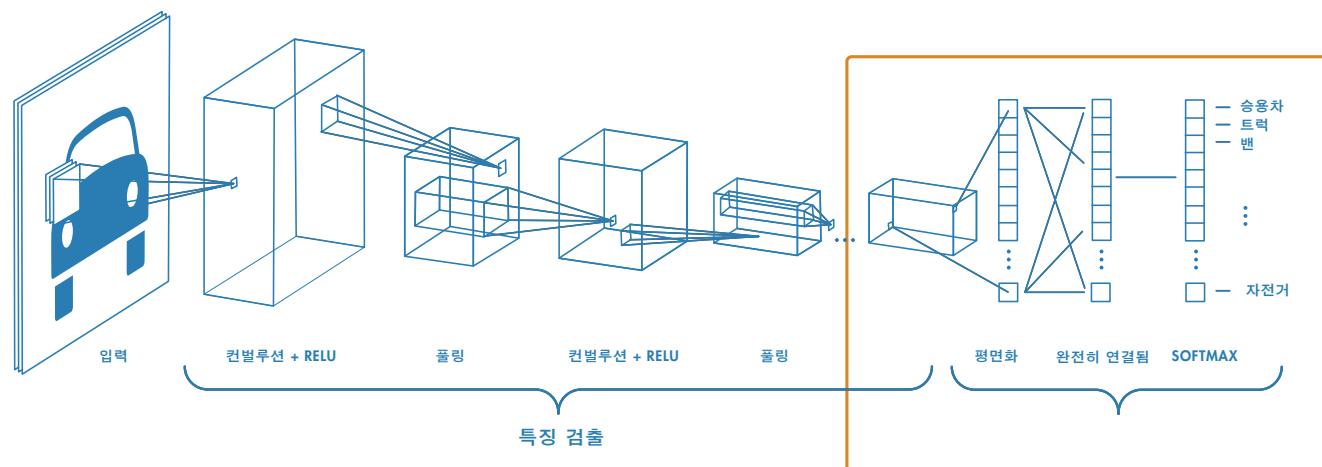
분류 계층

특징 검출 후 CNN의 아키텍처가 분류로 이동합니다.

끝에서 두 번째 계층은 K 차원의 벡터를 출력하는 완전 연결 계층 (FC)입니다. 여기서 K 는 네트워크가 예측할 수 있는 클래스의 수입니다. 이 벡터에는 분류되는 이미지의 각 클래스에 대한 확률이 포함됩니다.

CNN 아키텍처의 마지막 계층에서는 **softmax** 함수를 사용하여 분류 출력을 제공합니다.

계층을 선택하기 위한 정확한 공식은 없습니다. 가장 좋은 접근 방식은 몇 가지를 시도한 후 잘 작동하는지 확인하거나 사전 학습된 네트워크를 사용하는 것입니다.

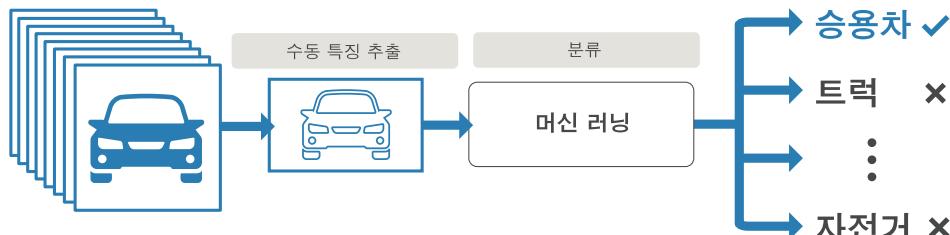


» [CNN 아키텍처 살펴보기](#)

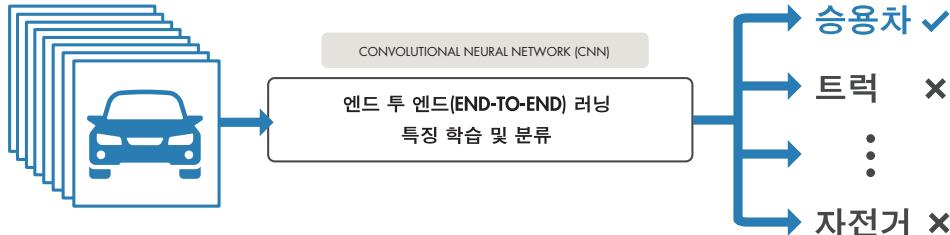
딥러닝과 머신러닝의 차이점

딥러닝은 머신 러닝의 서브타입입니다. 머신 러닝을 사용하는 경우 수동으로 이미지의 특징을 추출합니다. 반면 딥러닝을 사용하는 경우 자동으로 특징을 학습하는 심층 신경망에 직접 원본 이미지를 공급합니다. 딥러닝에서 종종 최상의 결과를 얻으려면 수십만 개 또는 수백만 개의 이미지가 필요합니다. 또한 딥러닝은 계산 집약적이며 고성능 GPU가 필요합니다.

기존의 머신 러닝



딥러닝

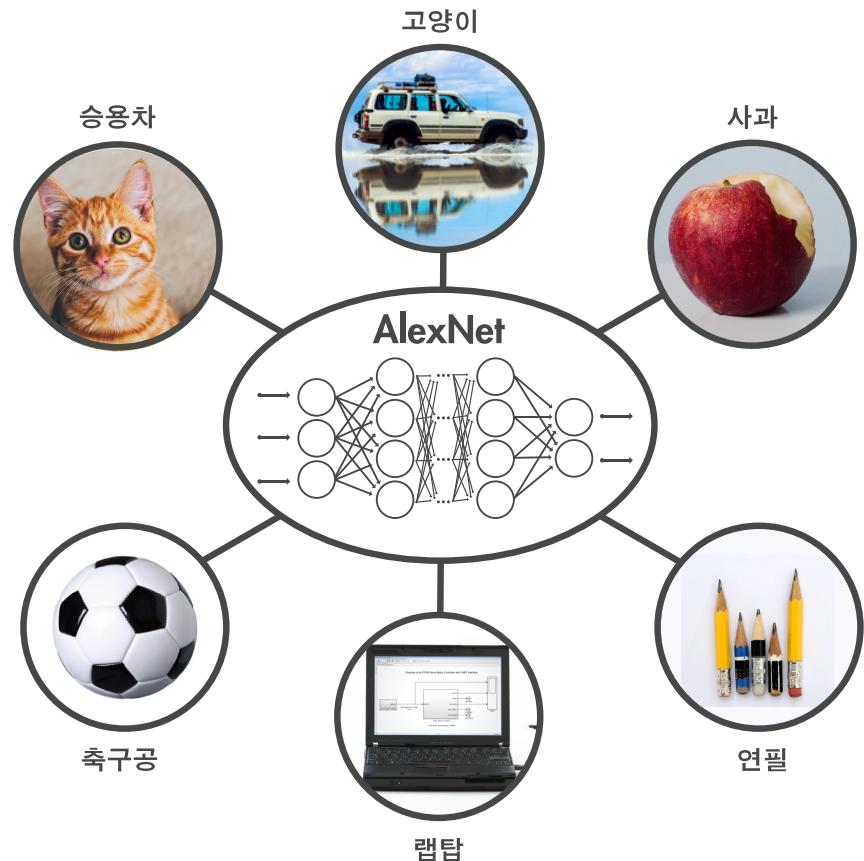


머신 러닝	딥러닝
+ 작은 데이터 세트로 좋은 결과를 얻을 수 있음	- 대용량 데이터 세트가 필요함
+ 모델을 빠르게 학습할 수 있음	- 계산 집약적
- 최상의 결과를 얻으려면 여러 특징과 분류기를 시도해야 함	+ 특징과 분류기를 자동으로 학습함
- 정확성이 안정적으로 유지됨	+ 제한 없는 정확성

딥러닝 시작하기

딥러닝을 처음 사용하는 경우 쉽고 빠르게 시작하는 방법은 100만 개 이상의 이미지에 대해 사전 학습된 CNN인 AlexNet과 같은 기존 네트워크를 사용하는 것입니다. AlexNet은 이미지 분류에 가장 일반적으로 사용됩니다. AlexNet은 다양한 품종의 개, 고양이, 말 및 기타 동물과 키보드, 컴퓨터 마우스, 연필 및 기타 사무용품을 포함한 1000개의 다양한 범주로 이미지를 분류할 수 있습니다.

AlexNet은 2012년에 처음 공개되었으며 연구 단체가 사용하는 대표적인 모델입니다.



» 사전 학습된 네트워크에 대해 자세히 알아보기

AlexNet 사용 예제

이미지의 객체를 분류하기 위해 AlexNet을 활용할 수 있습니다. 이 예제에서는 AlexNet을 사용하여 데스크탑에 설치된 웹캠에서 받은 이미지의 객체를 분류합니다. MATLAB® 과 함께 다음을 사용합니다.

- Deep Learning Toolbox™
- MATLAB에서 웹캠을 사용하기 위한 지원 패키지
- AlexNet 사용을 위한 지원 패키지

AlexNet을 로드한 후 웹캠에 연결하여 라이브 이미지를 캡처합니다.

```
camera = webcam; % 카메라에 연결  
  
nnet = AlexNet; % 신경망 로드  
  
picture = camera.snapshot;  
% 사진 찍기
```

다음으로, AlexNet에서 필요한 크기인 227x227픽셀로 이미지의 크기를 조정합니다.

```
picture = imresize(picture,[227,227]);  
% 사진 크기 조정
```

이제 AlexNet이 이미지를 분류할 수 있습니다.

```
label = classify(nnet, picture);  
% 사진 분류  
  
image(picture); % 사진 표시  
  
title(char(label)); % 레이블 표시
```



커피 머그잔

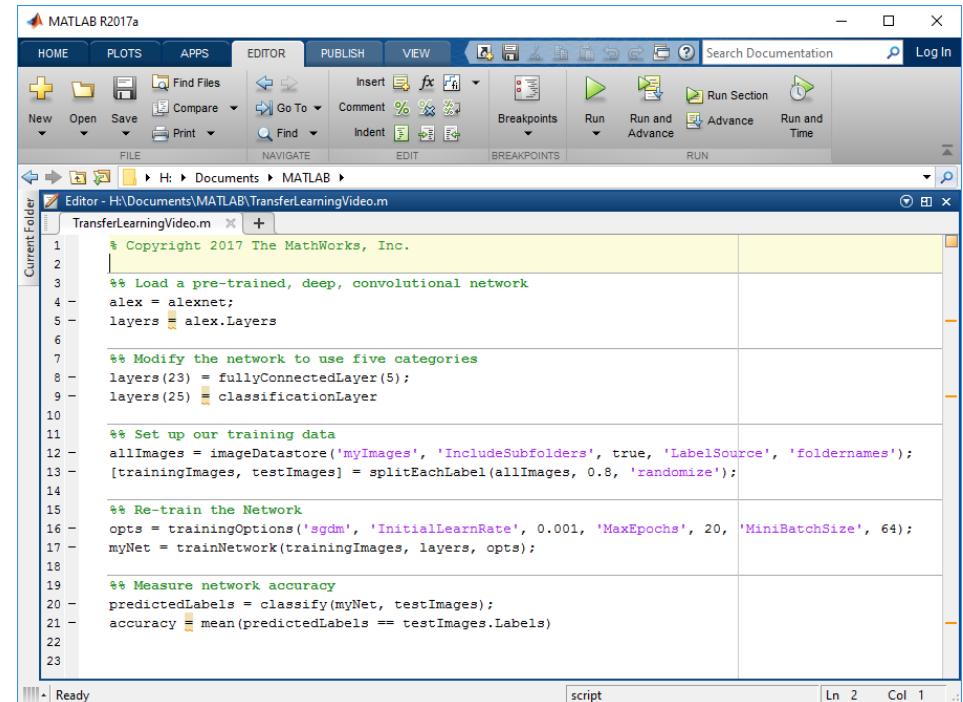
» 영상 보기: 단 11줄의 MATLAB 코드로 딥러닝하기

기존 네트워크 재학습

이전 예제에서는 분류하려고 했던 이미지와 유사한 이미지를 기반으로 AlexNet이 학습되었으므로 어떤 방식으로든 네트워크를 수정하지 않고 그대로 사용했습니다.

원래 네트워크에서 학습되지 않은 객체에 대해 AlexNet을 사용하려면 전이 학습을 통해 재학습할 수 있습니다. 전이 학습은 한 가지 유형의 문제에 대한 지식을 연관성이 있는 또 다른 문제에 적용하는 접근 방식입니다. 이 경우 네트워크의 마지막 3개 계층을 간단히 잘라내고 자체 이미지로 재학습합니다.

전이 학습이 응용프로그램에 적합하지 않으면 처음부터 자체 네트워크를 학습해야 할 수 있습니다. 이 방법은 가장 정확한 결과를 생성하지만, 일반적으로 수십만 개의 레이블이 지정된 이미지와 상당한 계산 리소스가 필요합니다.



```
% Copyright 2017 The MathWorks, Inc.

%% Load a pre-trained, deep, convolutional network
alex = alexnet;
layers = alex.layers;

%% Modify the network to use five categories
layers(23) = fullyConnectedLayer(5);
layers(25) = classificationLayer;

%% Set up our training data
allImages = imageDatastore('myImages', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
[trainingImages, testImages] = splitEachLabel(allImages, 0.8, 'randomize');

%% Re-train the Network
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
myNet = trainNetwork(trainingImages, layers, opts);

%% Measure network accuracy
predictedLabels = classify(myNet, testImages);
accuracy = mean(predictedLabels == testImages.Labels)
```

» 전이 학습 시작하기

딥러닝을 위한 계산 리소스

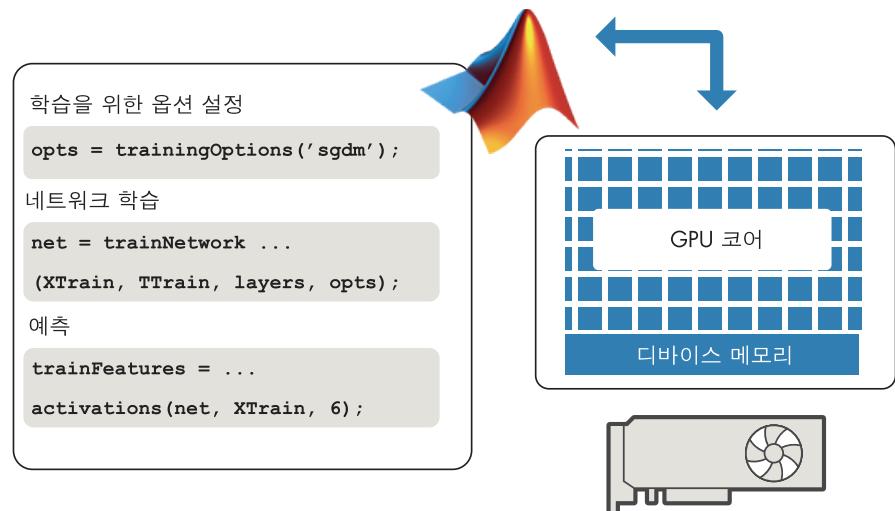
데이터의 크기와 사용 가능한 처리 능력의 양에 따라 딥러닝 모델을 학습하는 데 몇 시간, 며칠 또는 몇 주가 걸릴 수 있습니다. 계산 리소스 선택은 워크플로를 설정할 때 중요한 고려 사항입니다.

현재 CPU 기반, GPU 기반 및 클라우드 기반의 세 가지 계산 옵션이 있습니다.

CPU 기반 계산은 가장 단순하고 쉽게 사용할 수 있는 옵션입니다. 이전 섹션에 설명된 예제는 CPU에서 작동하지만, 사전 학습된 네트워크를 사용하는 간단한 예제에만 CPU 기반 계산을 사용하는 것이 좋습니다.

GPU 사용으로 네트워크 학습 시간이 며칠에서 몇 시간으로 단축됩니다. 추가 프로그래밍을 수행하지 않고 MATLAB에서 GPU를 사용할 수 있습니다. NVIDIA® 3.0 계산 가능 GPU가 권장됩니다. 여러 GPU를 사용하여 처리 속도를 더 향상시킬 수 있습니다.

클라우드 기반 **GPU** 계산은 하드웨어를 직접 구매하여 설치할 필요가 없음을 의미합니다. 로컬 GPU를 사용하기 위해 작성한 MATLAB 코드를 몇 가지 설정만 변경하여 클라우드 리소스를 사용하도록 확장할 수 있습니다.



» [GPU 기반의 빅데이터를 활용한 딥러닝에 대해 자세히 알아보기](#)

딥러닝 추가 리소스

딥러닝 소개

MATLAB을 활용한 딥러닝: 쿠 스타트 비디오

전이 학습을 사용하여 더 빠르게 딥러닝 시작

AlexNet을 사용한 전이 학습

CNN(Convolutional Neural Network) 소개

분류를 위한 단순 딥러닝 네트워크 생성

MATLAB을 활용한 컴퓨터 비전 딥러닝

딥러닝 및 포토닉 타임 스트레치(Photonic Time Stretch)를 사용하여 암 진단

