
딥러닝과 전통적인 머신러닝: 최상의 접근 방식 선택하기

딥러닝과 전통적인 머신러닝: 최상의 접근 방식 선택하기

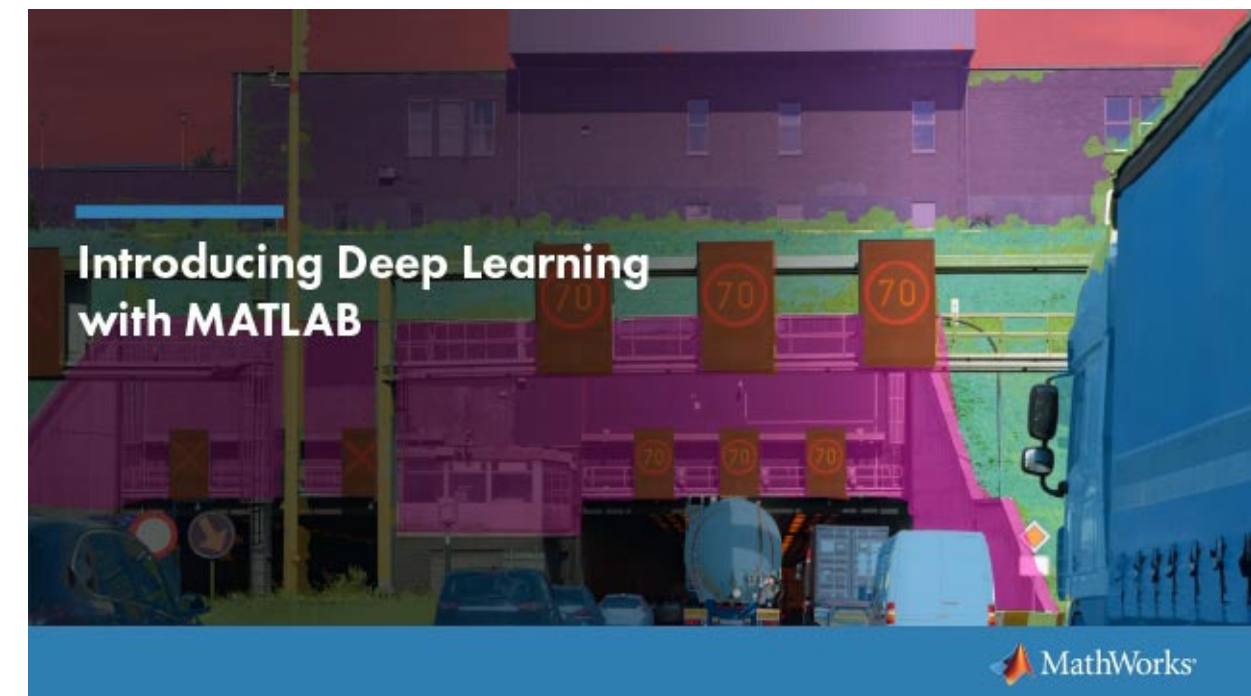
개요	3
용어	4
프로젝트	7
데이터	26
하드웨어	35
결론	52
알고리즘 추측: 답변	56

개요

AI, 딥러닝 및 머신 러닝의 중요성을 이야기하는 문서가 인터넷에 가득합니다. 엔지니어 또는 연구원으로서 새롭게 떠오르는 기술을 활용해야 하지만 어느 지점에서 시작합니까?

이 eBook에서는 초기 작업에 집중하는 시점에서 결정하는 데 도움이 되도록 딥러닝과 기존 머신 러닝 사이의 주요 차이점 몇 가지를 논의합니다. 머신 러닝 또는 딥러닝을 시작해야 하는지 선택의 요인이 되는 3가지 요소(프로젝트, 데이터 및 하드웨어)를 살펴보고, 이들을 함께 사용하고자 할 수 있는 경우를 설명하는 사례 연구를 살펴봅니다.

이 eBook에서는 AI 기술에 대한 기본 지식이 있음을 가정하며, 어떤 알고리즘을 먼저 사용해야 하는지 여부를 선택하는 데 있어서의 고려 사항을 논의합니다. 해당되는 기술에 대한 소개는 [MATLAB을 활용한 머신 러닝](#) 및 [MATLAB을 활용한 딥러닝 소개](#)를 참조하십시오.





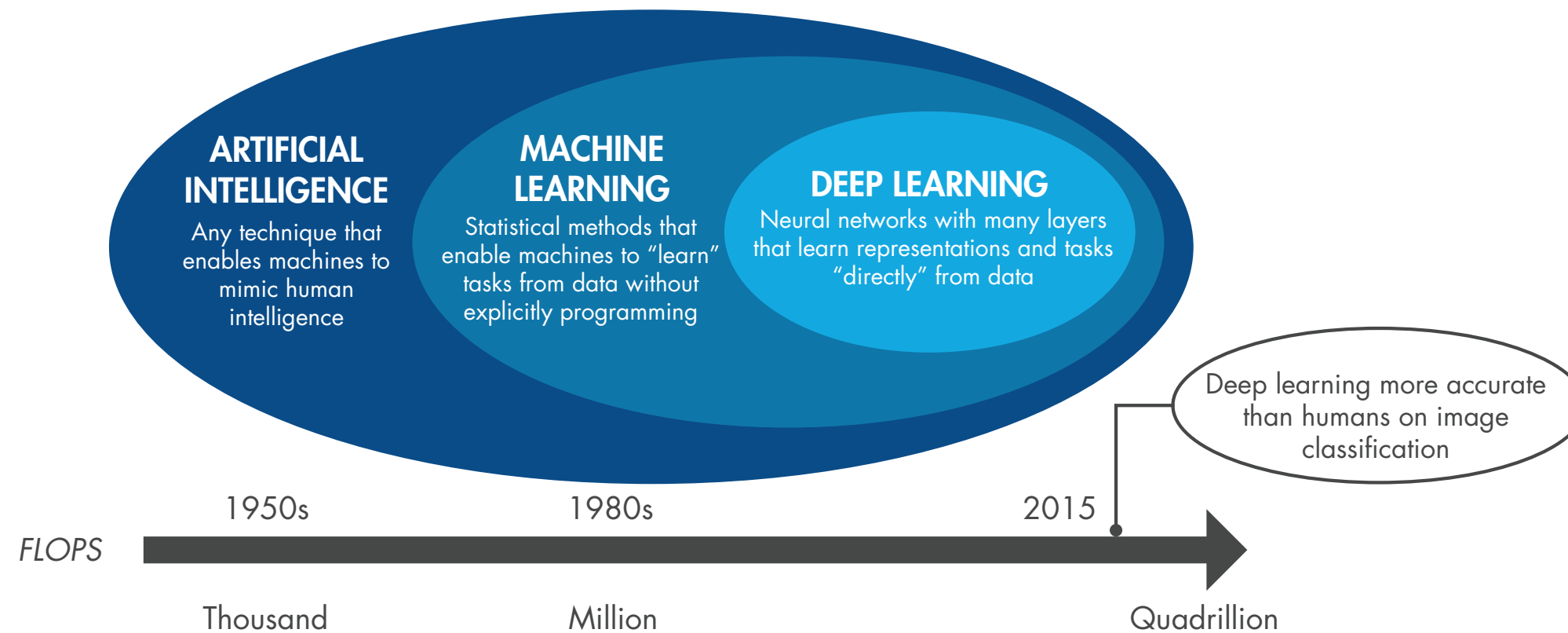
용어

용어

AI: 인공 지능(AI)은 환경을 인식하고, 결정을 내리고, 행동을 취하도록 학습된 컴퓨터 시스템.

머신 러닝: 데이터로부터 자동으로 학습하는 모델의 빌드와 연관된 기술. 이 eBook에서는 머신 러닝을 “기존의 머신 러닝”에 대한 약칭으로

사용합니다. 이는 사용할 관련 특징을 수동으로 선택하고 모델을 교육하는 워크플로를 의미합니다. 머신 러닝을 언급하는 경우 딥러닝은 제외됩니다. 일반 기술에는 결정 트리, 회귀, 서포트 벡터 머신 (SVM) 및 앙상블 기법이 있습니다.



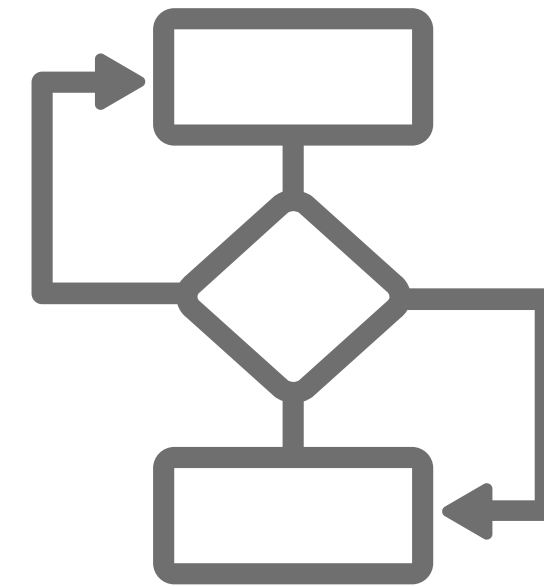
데이터 분석 알고리즘의 진행 과정

용어

딥러닝: 인간 두뇌의 뉴럴 경로에 느슨하게 모델링된 머신 러닝의 서브셋. 머신 러닝과의 주요 차이점으로 관련 특징을 수동으로 선택하는 대신 알고리즘이 자동으로 어떤 특징이 유용한지 학습합니다. 일반적인 기술로 컨벌루션 뉴럴 네트워크 (CNN), 순환신경망 (LSTM) 및 Deep Q 네트워크가 있습니다.

알고리즘: 모델이 해야 하는 일을 교육하는 규칙 또는 명령 세트.

모델: 일련의 입력에 대한 출력을 예측하는 학습된 프로그램.





프로젝트

프로젝트: 무엇을 하고자 합니까?

달성하고자 하는 바에 대한 명확한 그림을 그린 채 시작하는 것이 유용합니다. 머신 러닝 알고리즘 대신 딥러닝을 선택하는 데 있어 변치 않는 규칙은 거의 없기 때문에 이를 범위에 두고 선택을 고려하십시오.

하나의 작업만으로는 머신 러닝이 더욱 적합할 수도 있지만 전체 애플리케이션에 딥러닝이 더욱 적합한 여러 개의 단계가 포함될 수 있습니다.

다음 목록에는 일반적으로 수행되는 작업이 포함되어 있지만 이에 국한되지는 않습니다.



[▶ 딥러닝 소개 머신 러닝과 딥러닝의 차이점\(3:48\)](#)

프로젝트: 무엇을 하고자 합니까?

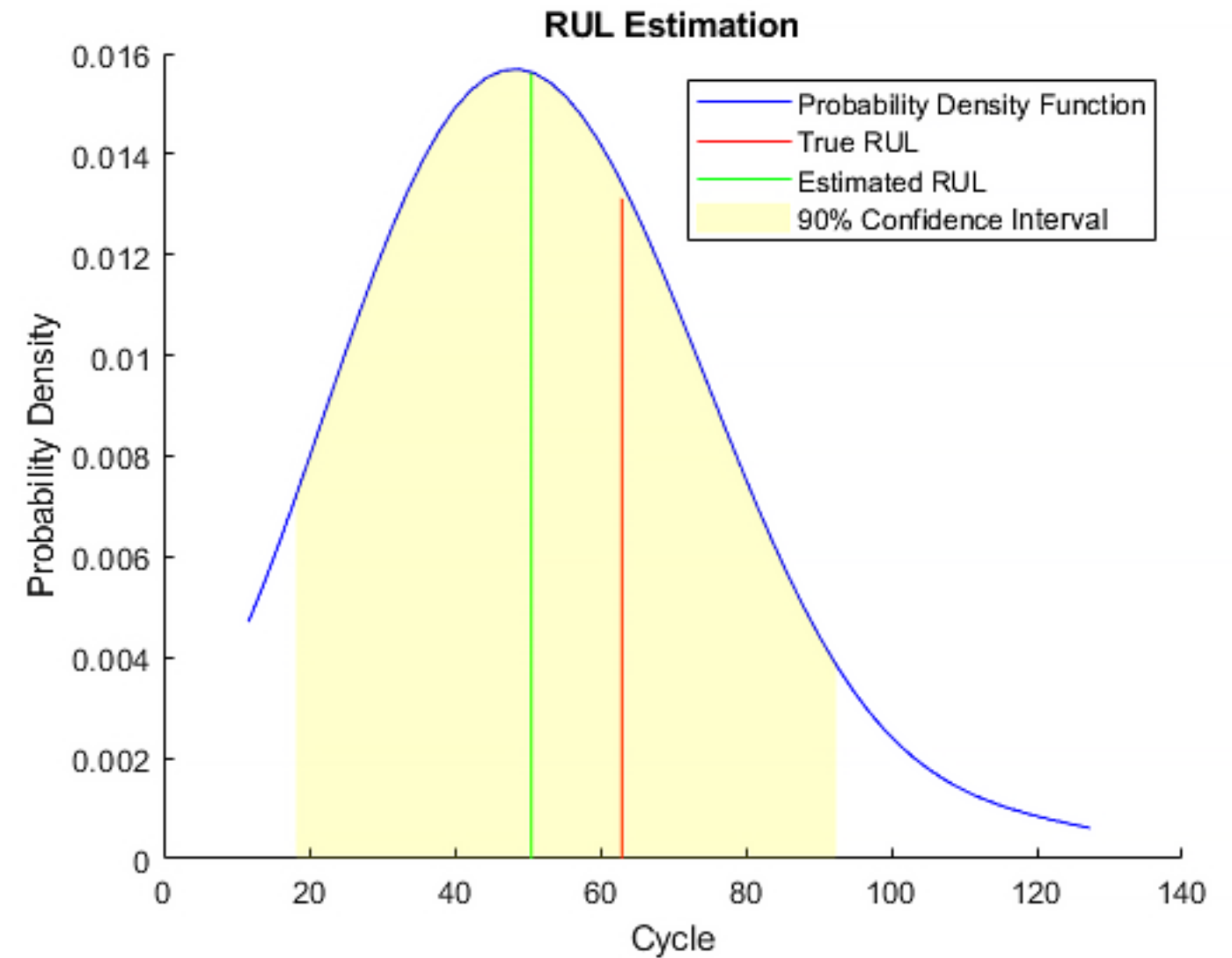
과거 및 현재 데이터를 기준으로 한 출력 예측.

사례: 모터의 실시간 센서 데이터를 사용하여 회전식 기계의 남은 수명을 예측합니다. 유사성 기반 잔여 수명 추정 사례는 선형 회귀를 사용합니다.

응용 분야: 건전성 예측관리, 재무 거래, 추천 시스템

입력: 센서 데이터, 타임스탬프 재무 데이터, 수치 데이터

일반 알고리즘: 선형 회귀, 결정 트리, 서포트 벡터 머신(SVM), 뉴럴 네트워크, 연결 규칙



잔여 수명 추정.

프로젝트: 무엇을 하고자 합니까?

이미지, 비디오 및 신호 데이터의 객체 또는 동작 식별.

사례: 차량을 감지할 수 있는 컴퓨터 비전 애플리케이션 생성. *Faster R-CNN* 딥러닝을 사용한 객체 감지 사례는 컨벌루션 뉴럴 네트워크 (CNN) 를 사용합니다.

응용 분야: 객체 감지를 통한 첨단 운전자 보조(ADAS), 로봇틱스, 이미지 인식을 위한 컴퓨터 비전 인식, 활동 감지, 음성 생체 인식(음성 지문)

입력: 이미지, 비디오, 신호

일반 알고리즘: CNN, 클러스터링, Viola-Jones



R-CNN을 사용하여 차량 감지.

프로젝트: 무엇을 하고자 합니까?

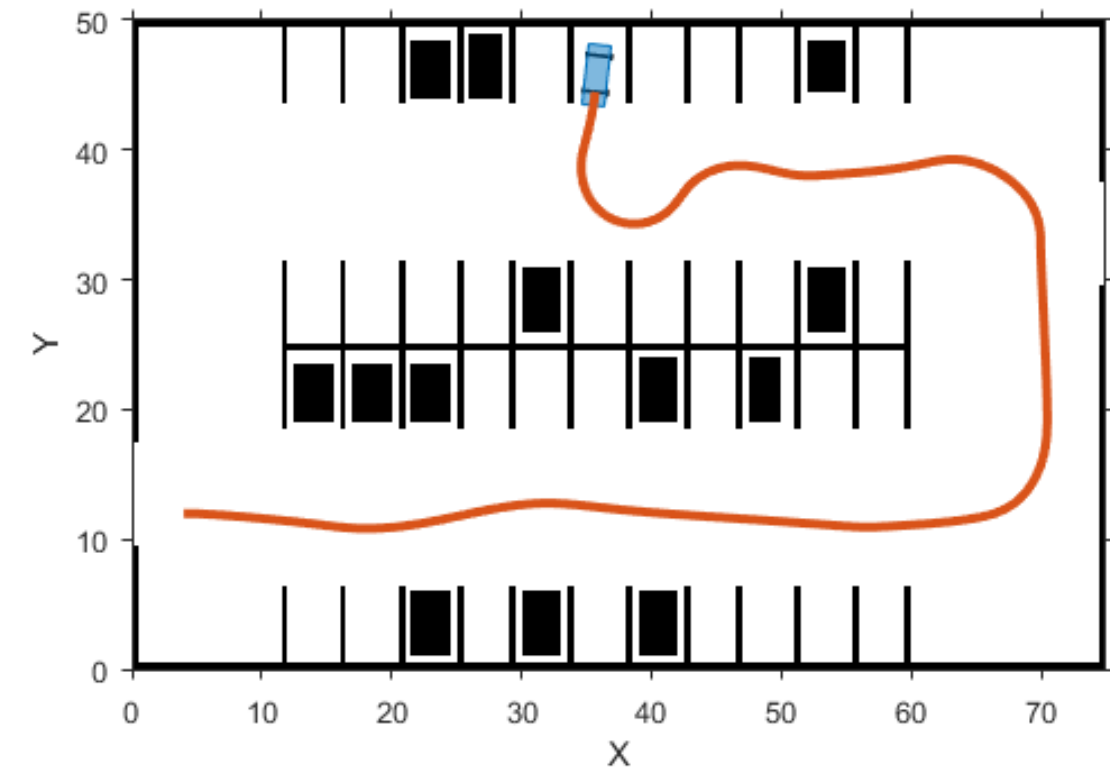
물리적 또는 시뮬레이션에서의 이동.

사례: 로봇 경로 계획을 수행하여 목적지까지의 최적의 경로를 학습.
강화 학습(Q-Learning) File Exchange 제출물은 Deep Q 네트워크를 사용합니다.

응용 분야: 제어 시스템, 제조 로봇틱스, 자체 주행 차량, 드론, 비디오 게임

입력: 삼각법, 물리학, 센서 데이터, 비디오, LIDAR 데이터

일반 알고리즘: 강화 학습(Deep Q 네트워크), 인공 신경망(ANN), CNN, 순환 신경망(RNN)



주차 경로 계획.

프로젝트: 무엇을 하고자 합니까?

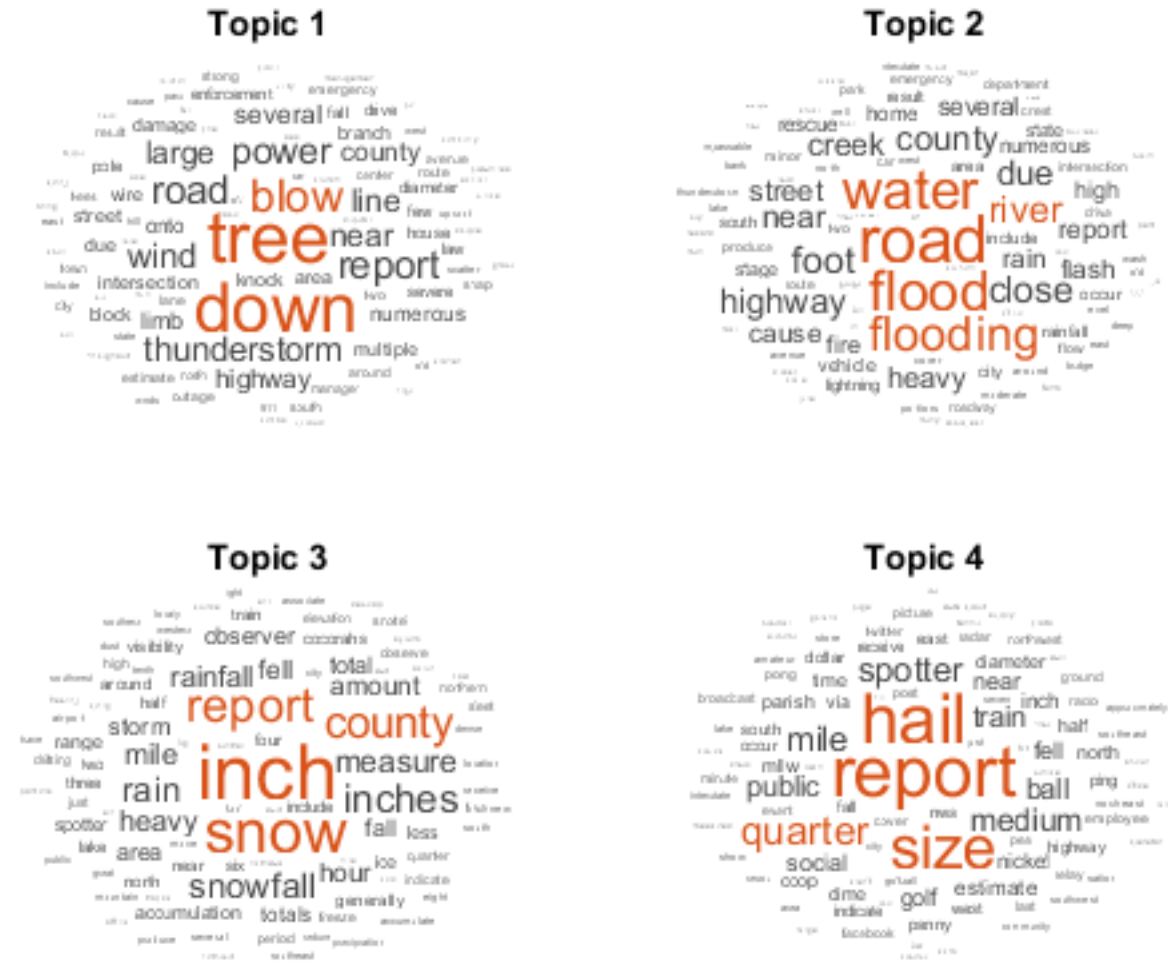
동향, 감정, 사기 또는 위협 파악.

사례: 텍스트 데이터에 몇 개의 주제가 있는지 판단. *주제 모델을 사용한 텍스트 데이터 분석* 사례는 latent Dirichlet allocation(LDA) 주제 모델을 사용합니다.

응용 분야: 안전 기록을 위한 자연어 처리, 시장 또는 의료 연구, 감정 분석, 사이버 보안, 문서 요약

입력: 스트리밍 텍스트 데이터, 정적 텍스트 데이터

일반 알고리즘: RNN, 선형 회귀, SVM, Naïve Bayes, latent Dirichlet allocation, latent semantic analysis, word2vec



주제와 관련이 있는 공통의 단어 시각화.

프로젝트: 무엇을 하고자 합니까?

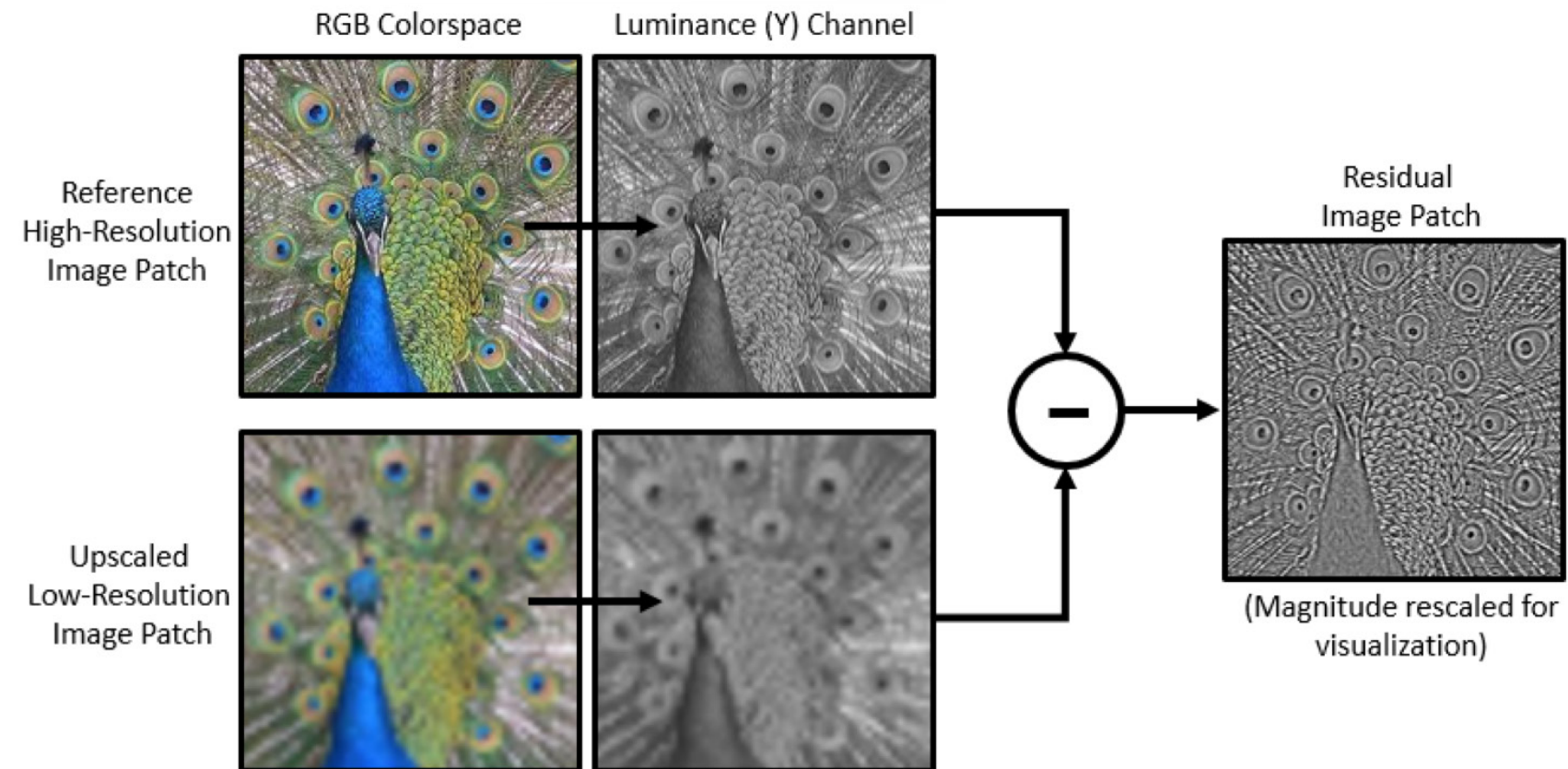
이미지 및 신호 향상.

사례: 저해상도 이미지에서 고해상도 이미지 생성. *딥러닝을 사용한 단일 이미지 슈퍼 해상도* 사례는 Very-Deep Super-Resolution(VDSR) 뉴럴 네트워크를 사용합니다.

응용 분야: 이미지 해상도 개선, 오디오 신호의 소음 제거

입력: 이미지 및 신호 데이터

일반 알고리즘: LSTM, CNN, VDSR 뉴럴 네트워크



저해상도 이미지에서 고해상도 이미지 생성.

프로젝트: 무엇을 하고자 합니까?

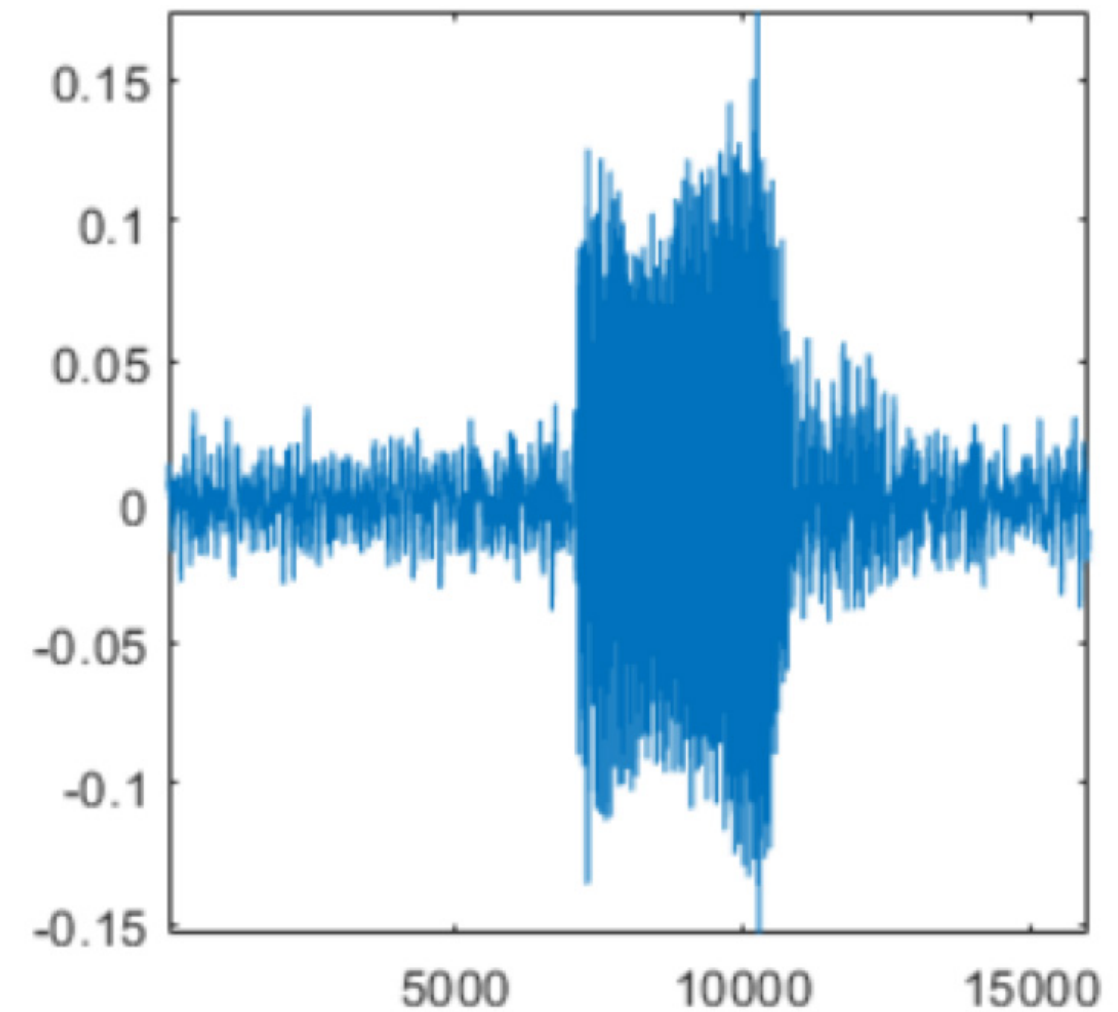
문맥 및 학습된 루틴을 기반으로 스피치 또는 텍스트 명령에 응답.

사례: “on,” “off,” “stop” 및 “go”와 같은 구두 명령을 자동으로 인식.
딥러닝을 사용한 명령 인식 사례는 CNN을 사용합니다.

응용 분야: 고객 관리 통화, 스마트 디바이스, 가상 비서, 기계 번역 및 딕테이션

입력: 음향 데이터, 텍스트 데이터

일반 알고리즘: RNN(특히 LSTM 알고리즘), CNN, word2vec



발화된 명령 “on”에 대한 오디오 신호.

프로젝트: 얼마나 정확해야 합니까?

정확도는 모든 머신 러닝 및 딥러닝 프로젝트에 있어 주요 지표입니다. 프로젝트 초기에는 정확도를 극대화하는 데에만 초점을 맞추고, 다른 고려 사항(모델의 메모리 용량 또는 예측 속도)은 나중에 해결하도록 두는 경우가 일반적입니다.

직관적으로는 이 접근 방식이 타당합니다. 프로덕션에서의 정확도를 낮출 수 있는 제약 조건에 대해 우려하기 전에 최대의 달성 가능한 정확도가 어느 정도인지 파악하는 것이 유용합니다.



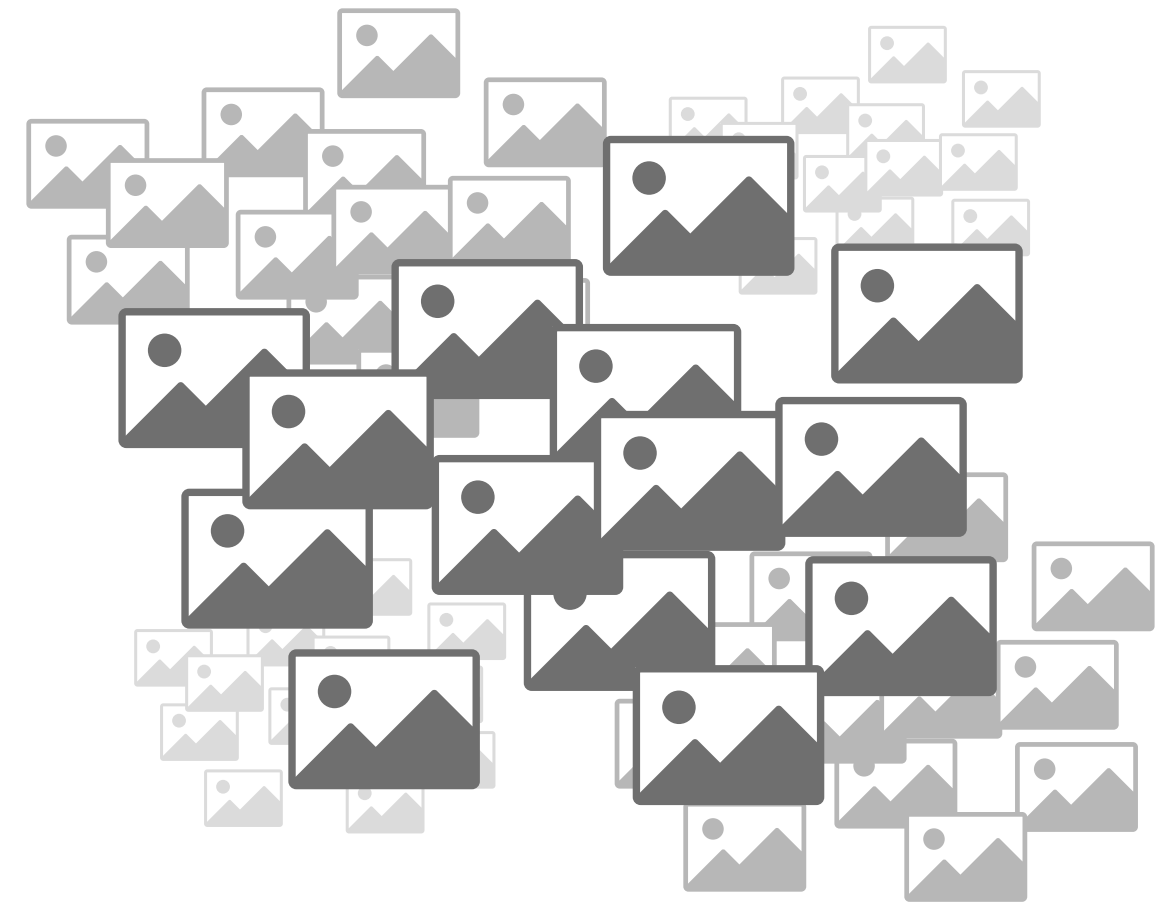
네트워크가 비슷한 이미지(예: 아프리카와 유럽 제비) 사이에서 구분할 수 있도록 교육하려면 더 많은 데이터가 필요합니다.

프로젝트: 얼마나 정확해야 합니까?

일반적으로 딥러닝 기술은 기존 머신러닝 기술보다 더욱 정확한 결과를 생산할 수 있는데, 이는 더 많은 파라미터가 포함된 더욱 복잡한 모델을 사용하고 그에 따라 데이터에 더욱 근접하게 “피팅”할 수 있기 때문입니다.

딥러닝이 문제 해결에 더욱 적절한 정도의 대량의 데이터와 애플리케이션을 보유한 경우 딥러닝 기술로 시작하고자 할 것입니다. 더욱 정확한 결과를 제공할 수 있기 때문입니다.

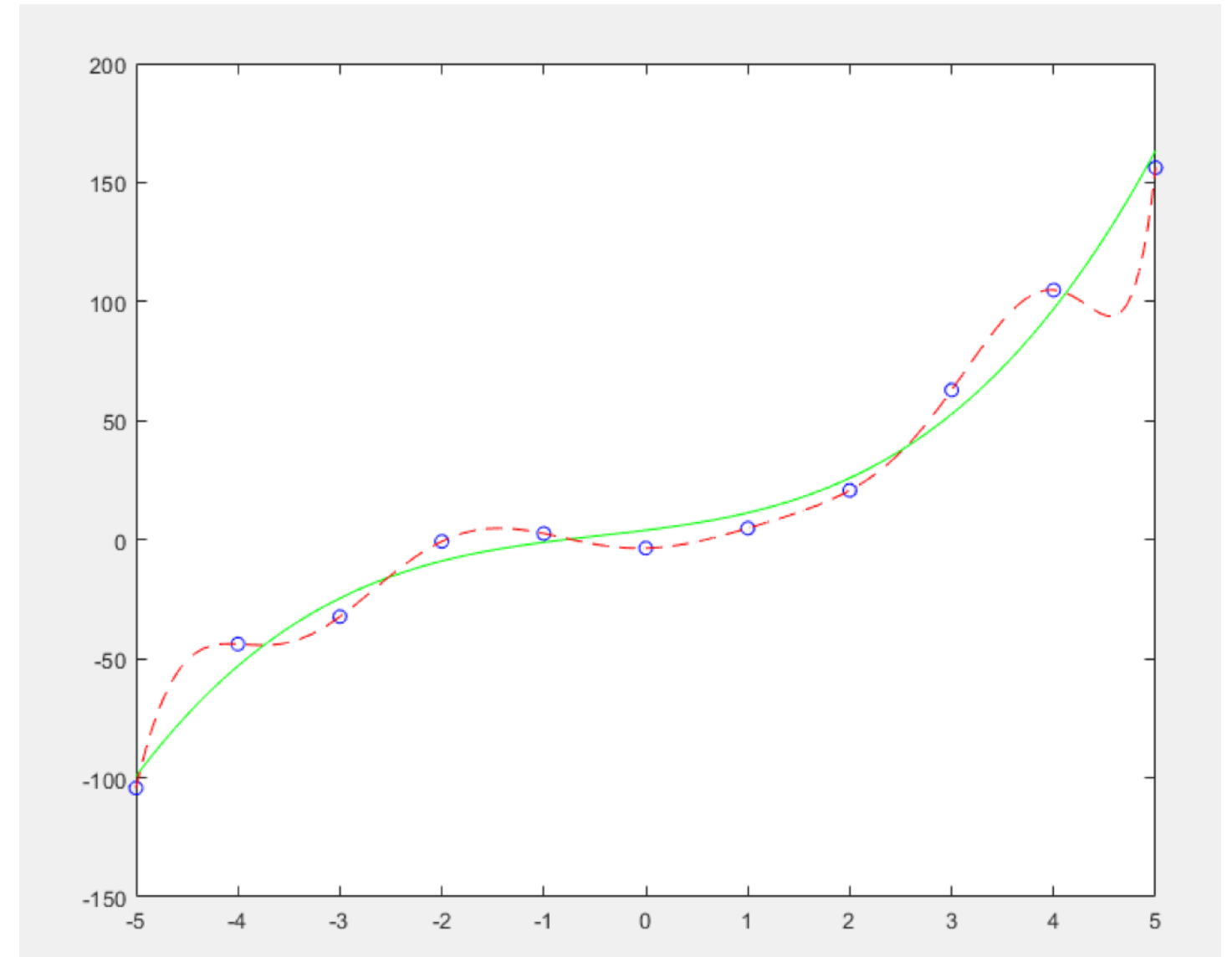
그렇다면 어느 정도의 데이터가 “대량”의 데이터셋입니까? 이는 경우에 따라 다릅니다. 전이 학습에 사용할 수 있는 일부 널리 사용되는 이미지 분류 네트워크는 1,000개의 다양한 범주에 속한 120만 개의 이미지가 포함된 데이터셋을 통해 학습됩니다.



프로젝트: 얼마나 정확해야 합니까?

정확도에 초점을 맞출 때 모델의 오버피팅 위험이 발생합니다. 이는 알고리즘이 학습 데이터와 너무 밀접하게 관련이 있어서 더욱 폭넓은 데이터셋을 일반화할 수 없을 때 발생합니다.

시작 단계에서 오버피팅을 방지하려면 많은 학습, 유효성 검사 및 테스트 데이터를 보유해야 합니다. 우선 학습 및 유효성 검사 데이터를 사용하여 모델을 학습시켜야 합니다. 데이터는 실제 데이터를 대표해야 하고 그 양이 충분해야 합니다. 모델이 학습되면 완전히 새로운 테스트 데이터를 사용하여 모델이 원활하게 작동하는지 확인합니다



오버피팅 모델은 예기치 않은 데이터를 원활하게 처리하지 못합니다.

프로젝트: 얼마나 정확해야 합니까?

모델이 데이터를 오버피팅하기 시작하는 경우 다음을 살펴보십시오.

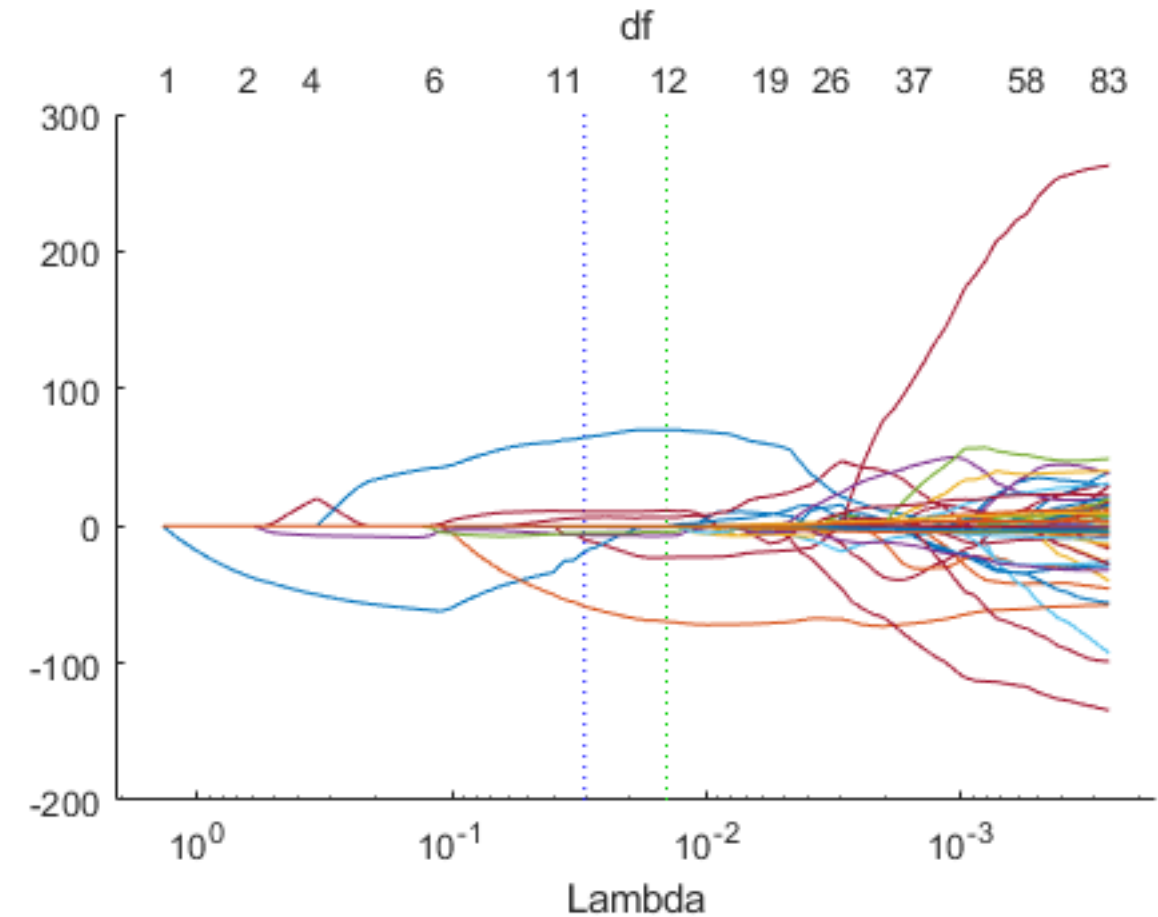
정규화 - 큰 파라미터에 페널티를 부여하여 모델이 개별 데이터 포인트에 너무 의존하지 않도록 합니다.

확률 드롭아웃 - 임의로 일부 데이터를 건너뛰어 모델의 데이터셋 기억을 방지합니다.

현실과 마찬가지로 상식적인 요소와 시행착오가 있습니다. 대부분의 실무자는 실험을 통해 어떻게 다양한 파라미터가 정확도에 영향을 미치는지에 대한 직관을 개발합니다.

명심할 사항...

일부 지점에서 수확 체감에 가까워질 수 있으며, 이 시점에서 정확도를 크게 개선하지 않고 모델 결과를 수정합니다. 최종 목표를 염두에 두고, 가급적 정확도 및 오버피팅의 위험에 대한 증분 개선의 비즈니스 영향을 염두에 두는 것이 좋습니다.



Lasso에 맞춰 조정된 계수의 추적 플로팅.

프로젝트: 결과를 설명해야 합니까?

데이터 과학자는 모델 해석 가능성으로서의 결과를 공유 및 설명할 수 있는 능력과 관련이 있는 경우가 많습니다. 쉽게 해석 가능한 모델에는 다음이 포함되어 있습니다.

- 일반적으로 시스템의 일부 물리적 이해로부터 생성된 소수의 특징
- 투명한 의사 결정 프로세스

해석 가능성은 다음을 수행해야 하는 많은 애플리케이션에 있어 중요합니다.

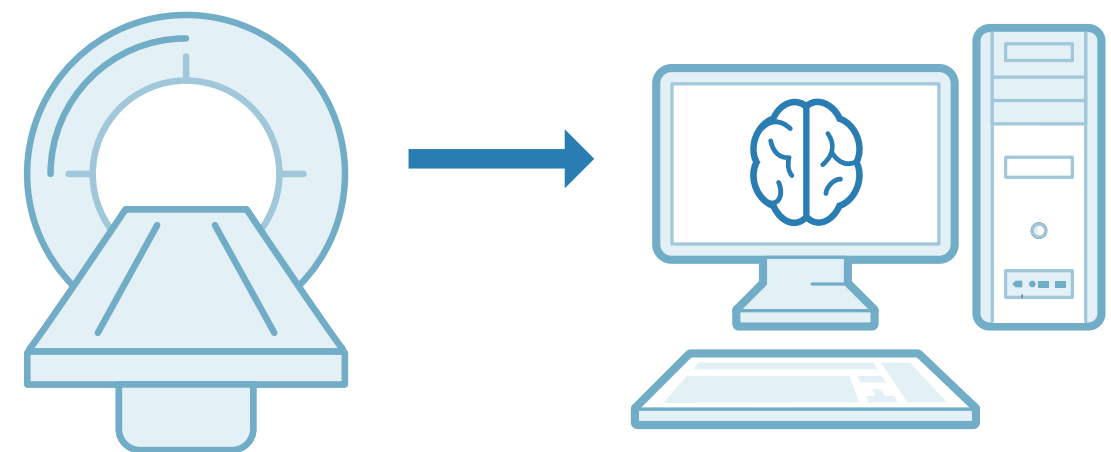
- 모델이 정부 또는 업계 표준을 준수하는지 여부를 입증
- 진단에 작용되는 요소를 설명
- 의사결정에 있어서 편향되지 않음을 보여줌

알고리즘 추측

연구원이 초저선량 CT 스캔(방사능 노출량이 감소하지만 이미지 해상도 또한 줄어듦)을 수행하고 이미지 처리 기술을 적용하여 이미지 해상도를 복원하는 방법을 설계했습니다.

연구원이 사용한 기술은 무엇입니까?

SVM 또는 CNN



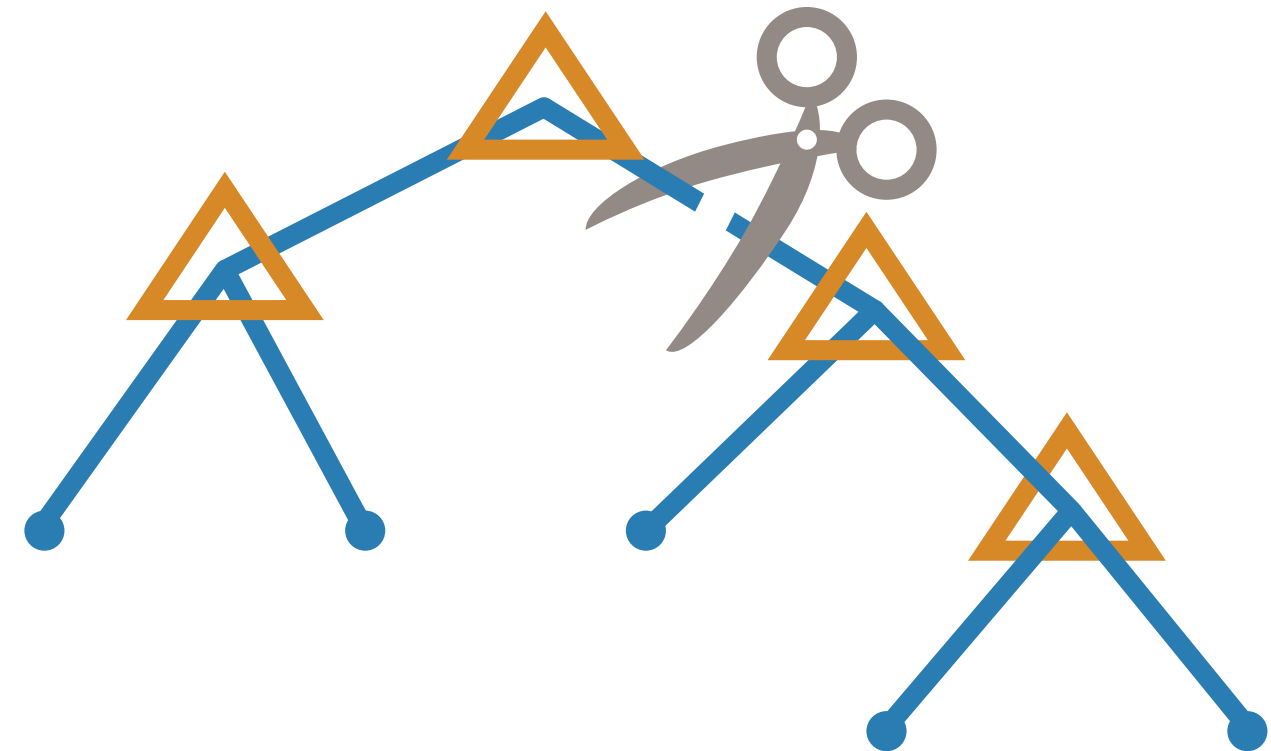
프로젝트: 결과를 설명해야 합니까?

알고리즘이 결론으로 도달하기까지 밟은 단계를 표시해야 하는 경우 머신러닝 기술에 초점을 맞추십시오. 결정 트리는 “if x, then y”와 같은 이진 경로를 따라가기 매우 쉽습니다.

선형 및 로지스틱 회귀와 같은 기존 통계 기술을 사용할 수 있습니다. 한 번에 하나의 트리를 가져오는 경우 랜덤 포레스트조차도 상대적으로 설명하기 간편합니다.

로컬 해석 가능 모델 진단 설명(LIME)은 일련의 개별 입력 및 출력을 가져와 의사 결정의 근사치를 냅니다. *문서를 읽어보십시오.*

연구의 또 다른 영역은 결정 트리를 더욱 복잡한 모델을 보여주는 방법으로 사용하는 것입니다. *문서를 읽어보십시오.*

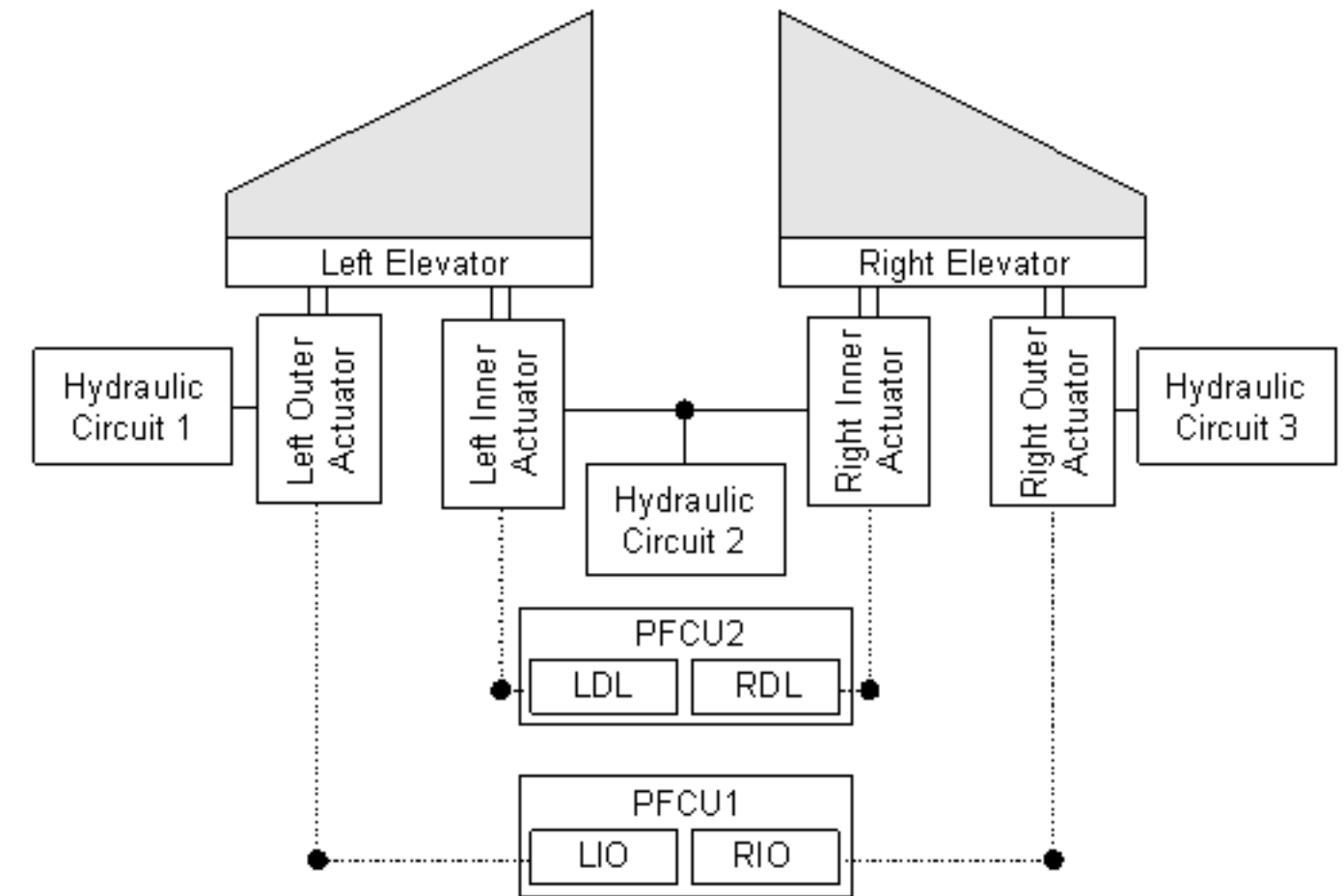


프로젝트: 도메인 지식

당신의 프로젝트에 포함된 시스템에 대해 얼마나 알고 있습니까?

제어 애플리케이션을 작업 중인 경우 프로젝트에 영향을 미칠 수 있는 관련 시스템을 이해하고 있습니까? 또는 경험이 더욱 폐쇄적입니까?

도메인 지식은 모델에 포함시킬 데이터를 선택하고 이러한 데이터의 가장 중요한 특징을 결정하는 역할을 할 수 있습니다.



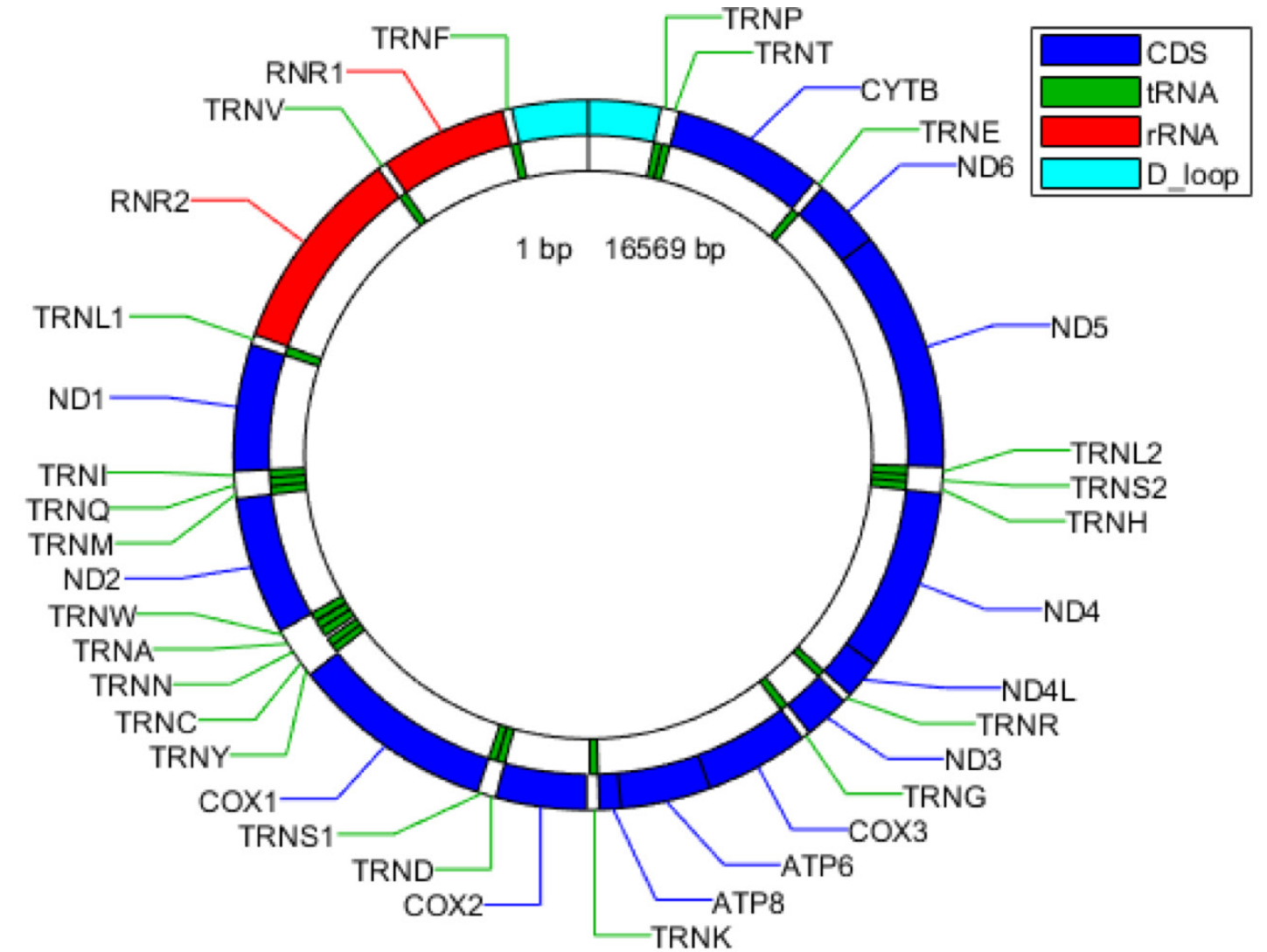
엘리베이터 시스템의 구성요소가 다른 시스템과 어떻게 연결되어 있는지를 보여주는 도식.

프로젝트: 도메인 지식

어떤 데이터를 포함해야 하나?

예를 들어 의료분야 연구자는 대량의 환자 데이터를 파악하고자 합니다. DNA 특성에 따른 질병의 특성부터 환경적 요소에 이르기까지 환자 통계에는 수천 가지의 특징이 존재할 수 있습니다.

데이터를 확실히 이해한 경우 가장 영향력이 높을 것으로 생각되는 특징을 선택하고 머신 러닝 알고리즘을 시작합니다. 다차원 데이터를 보유한 경우 **주성분 분석(PCA)**과 같은 차원 축소 기술을 사용하여 특징의 수를 줄이고 결과를 개선합니다.



인간 미토콘드리아 게놈의 특징.

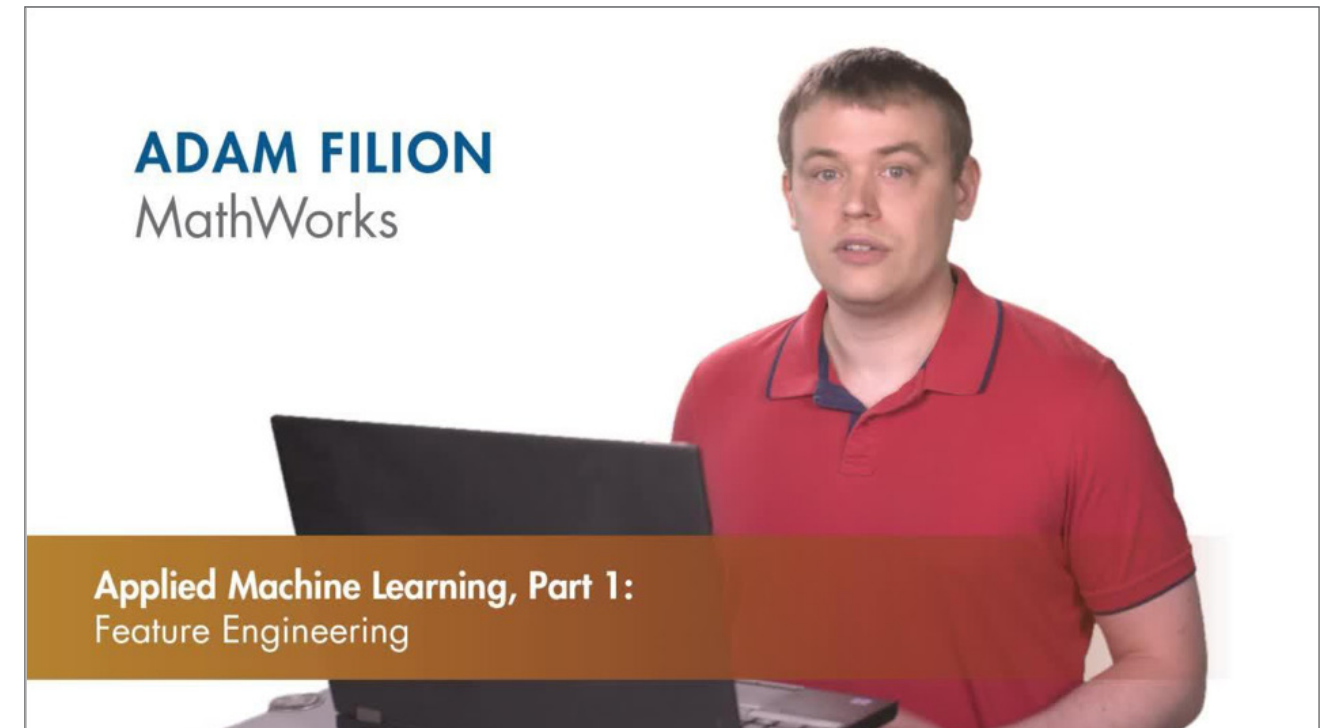
프로젝트: 도메인 지식

특징 선택

예측 검정력이 가장 높은 데이터에 모델의 초점을 맞추고 의사 결정에 영향을 미치지 않는 데이터에 집중하지 않도록 해야 합니다. 정확한 특징 선택에 따라 모델의 속도, 효율성 및 해석 가능성이 증가합니다.

지도 학습 알고리즘을 사용하는 경우 도메인 지식을 사용하여 데이터의 중요 특징을 수동으로 선택해야 합니다.

도메인 지식이 제한적인 경우 또는 데이터에 폭넓은 범위의 특징이 존재하는 경우 NCA(Neighborhood Component Analysis)와 같은 자동 특징 선택 기능을 사용해 보십시오.

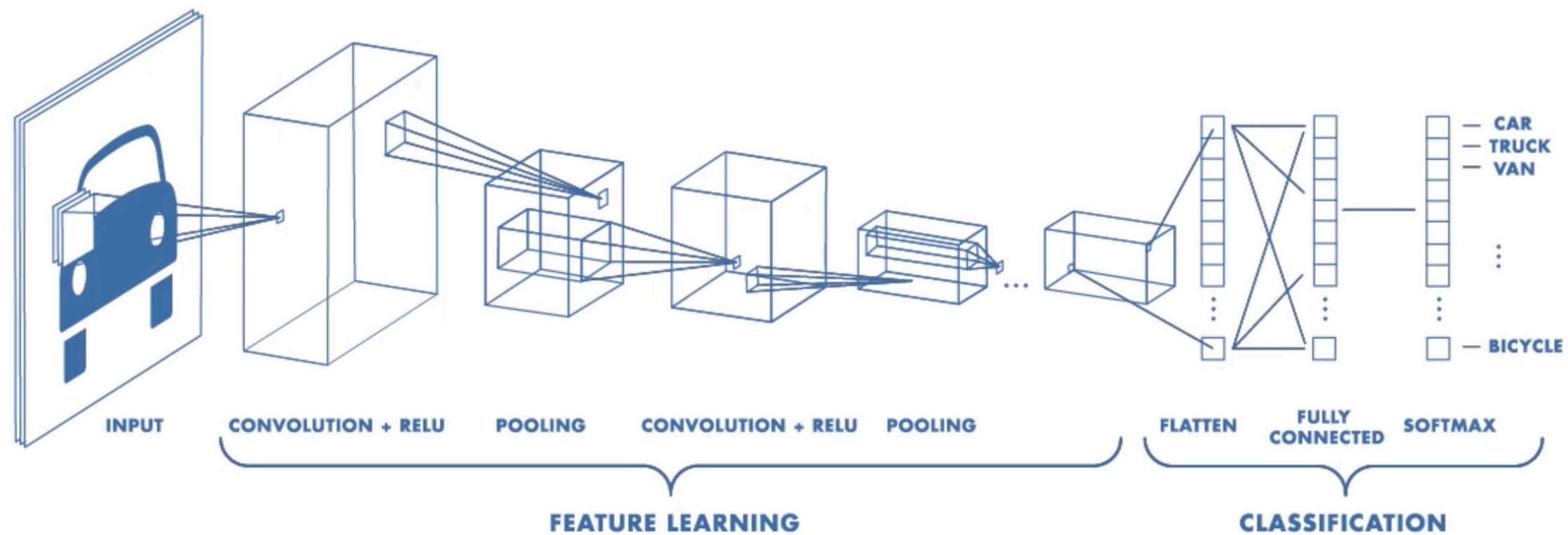


▶ 어플라이드 머신 러닝, 파트 1:
특징 엔지니어링 (4:35)

프로젝트: 도메인 지식

여러 딥러닝 알고리즘의 이점 중 하나는 특징 추출 및 선택 작업을 자동으로 수행한다는 점입니다. 데이터를 처리하고자 하는 경우 모델에서 자체적으로 데이터의 어떤 특징이 가장 중요한지 결정합니다.

딥러닝 및 머신 러닝을 함께 사용하는 일반적인 예로 CNN을 사용하여 특징을 선택한 다음 머신 러닝 알고리즘에 공급하는 방식이 있습니다. 이를 수행하는 방법에 대한 예시는 섹션 4에서 살펴봅니다.



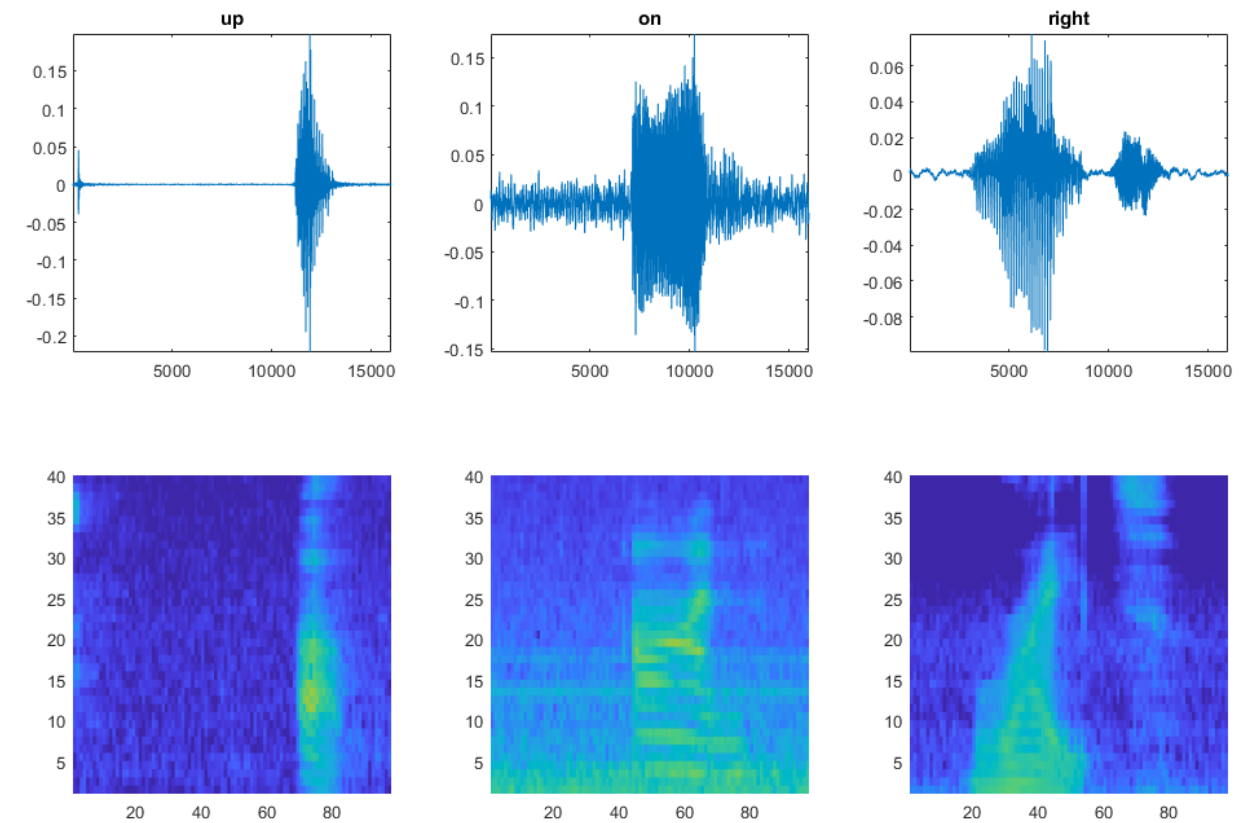
CNN을 사용한 특징 추출 및 선택 수행.

프로젝트: 도메인 지식

신호 처리 엔지니어의 경우 종종 1D 신호를 변형하여 차원을 축소해야 합니다. 신호는 주로 높은 주파수율로 발생하기 때문에 많은 데이터를 원시 형태로 처리할 수 없습니다.

한 가지 일반적인 방법으로 스펙트로그램과 같은 변형을 사용하여 1D 신호를 2D 표현으로 변환하는 방법이 있습니다.

이 변환은 신호에서 가장 가시적인 주파수를 강조 표시합니다. 이를 통해 CNN에 대한 입력으로 사용할 수 있는 “이미지”가 생성됩니다.



원래 신호(상단)와 단어에 대한 해당 스펙트로그램(하단).



데이터

데이터: 유형

테이블 형식 데이터

테이블 형식이란 무엇입니까? 각각에 대해 독립적인 열로 구성된 데이터베이스 또는 직원 정보에 대해 생각해 보십시오.

테이블 형식 데이터는 수치형 또는 범주형이 될 수 있습니다 (하지만 범주형 데이터는 결국 수치형으로 변환됨).

기존 머신 러닝 기술은 테이블 형식 데이터를 열두에 두고 설계되었습니다. 따라서 데이터가 테이블 형식인 경우 머신 러닝으로 시작할 수 있습니다. 딥러닝 모델에서 사용하도록 테이블 형식 데이터를 변환하는 방법이 있지만 처음 시작할 때에 있어서는 최적의 옵션이 아닙니다.

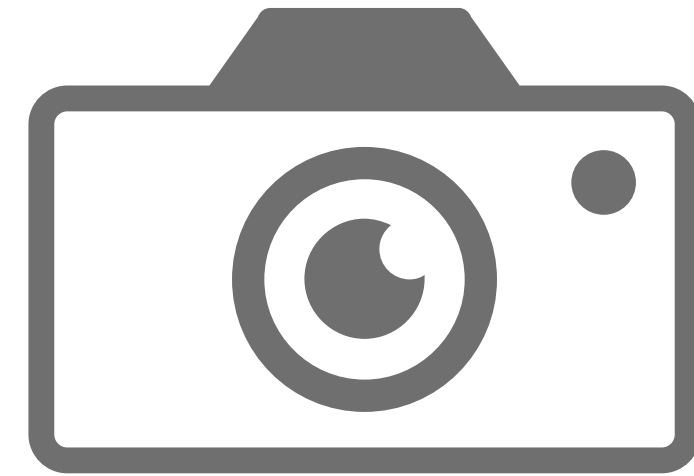
<u>EmployeeID</u>	<u>AddressLine1</u>	<u>AddressLine2</u>	<u>StartDate</u>
1111	"5 Maple St"	""	01-Jan-2005 00:00:00
7654	"8 Main Ave"	"Apt 13"	31-Dec-2014 00:00:00
80	"835 High St"	""	31-May-2000 00:00:00
6424	"42 Oakridge Rd"	"Unit 4"	02-Aug-2013 00:00:00

데이터: 유형

이미지 및 비디오

딥러닝 접근 방식은 여러 이미지 및 비디오 분류 문제에 있어 표준이 되었습니다. 구체적으로 설명하자면 컨벌루션 뉴럴 네트워크 (CNN)는 이미지로부터 특징을 추출하도록 설계되었고, 이를 통해 최고 수준의 분류 정확성이라는 결과를 낳는 경우가 많습니다.

직관적으로 컨벌루션 필터에 의해 수행되는 작업은 이미지에서 더 높은 수준의 특징을 추출할 수 있어 높은 수준의 차이점(예: 고양이와 개의 차이점)을 파악할 수 있습니다.



데이터: 유형

센서 및 신호

기존 접근 방식으로는 신호로부터 특징을 추출하고(시간 및 주파수 도메인 모두) 이러한 특징을 머신 러닝 알고리즘에 사용하는 방법이 있습니다.

최근으로 올수록 신호가 LSTM 네트워크로 직접 전달되거나 이미지로 변환된 다음 컨벌루션 뉴럴 네트워크에서 사용되었습니다.

웨이블릿은 신호로부터 특징을 추출하는 또 다른 방법을 제공합니다. 웨이블릿 스퀘터링과 같은 기술을 머신 러닝 알고리즘과 결합하는 경우 더욱 가능성 높은 결과가 제공됩니다.



데이터: 유형

텍스트

기존 접근 방식에는 단어 주머니(Bag-of-Words) 모델 및 정규화 기술(예: TF-IDF)을 통한 텍스트의 수치 표현 변환이 포함되어 있습니다.

이 수치 데이터는 서포트 벡터 머신 (SVM) 또는 Naïve Bayes와 같은 기존 머신 러닝 기술에 사용할 수 있습니다. 최신 기술의 경우 순환 신경망 및 컨벌루션 뉴럴 네트워크 아키텍처에 텍스트를 사용합니다.

이러한 경우 텍스트는 주로 word2vec과 같은 단어 임베딩 모델을 사용하여 수치 표현으로 변환됩니다.



데이터: 데이터 레이블 지정

지도 모델을 학습하려면 머신 러닝 또는 딥러닝 여부 사용 여부와 무관하게 레이블 지정된 데이터가 필요합니다.

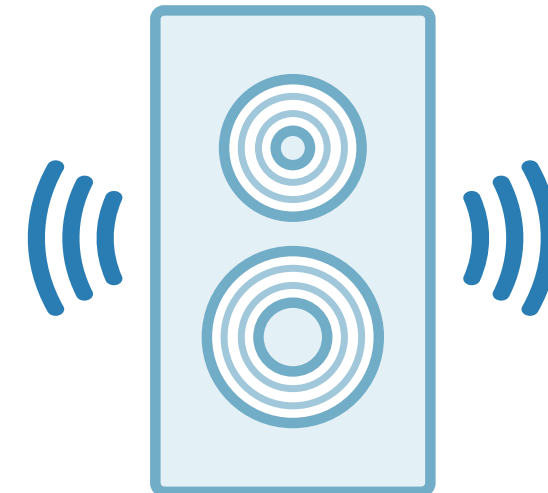
데이터가 레이블 지정되지 않은 경우 도움이 되는 기술이 있습니다.

머신 러닝의 경우 클러스터링과 같은 비지도 기술을 사용하여 데이터를 그룹으로 세분화할 수 있습니다. 이후 이러한 그룹은 어떤 데이터 포인트가 유사한지 이해하기 위해 분석된 다음 적절히 레이블 지정됩니다.

알고리즘 추측

가전제품 업체에서 스피커가 전문 음악인에게 적합한 제품인지 여부를 결정하는 모델을 생성했습니다. 60개의 스피커와 50개 이상의 특징이 있는 경우 **최적의 알고리즘은 무엇입니까?**

SVM 또는 인공 신경망



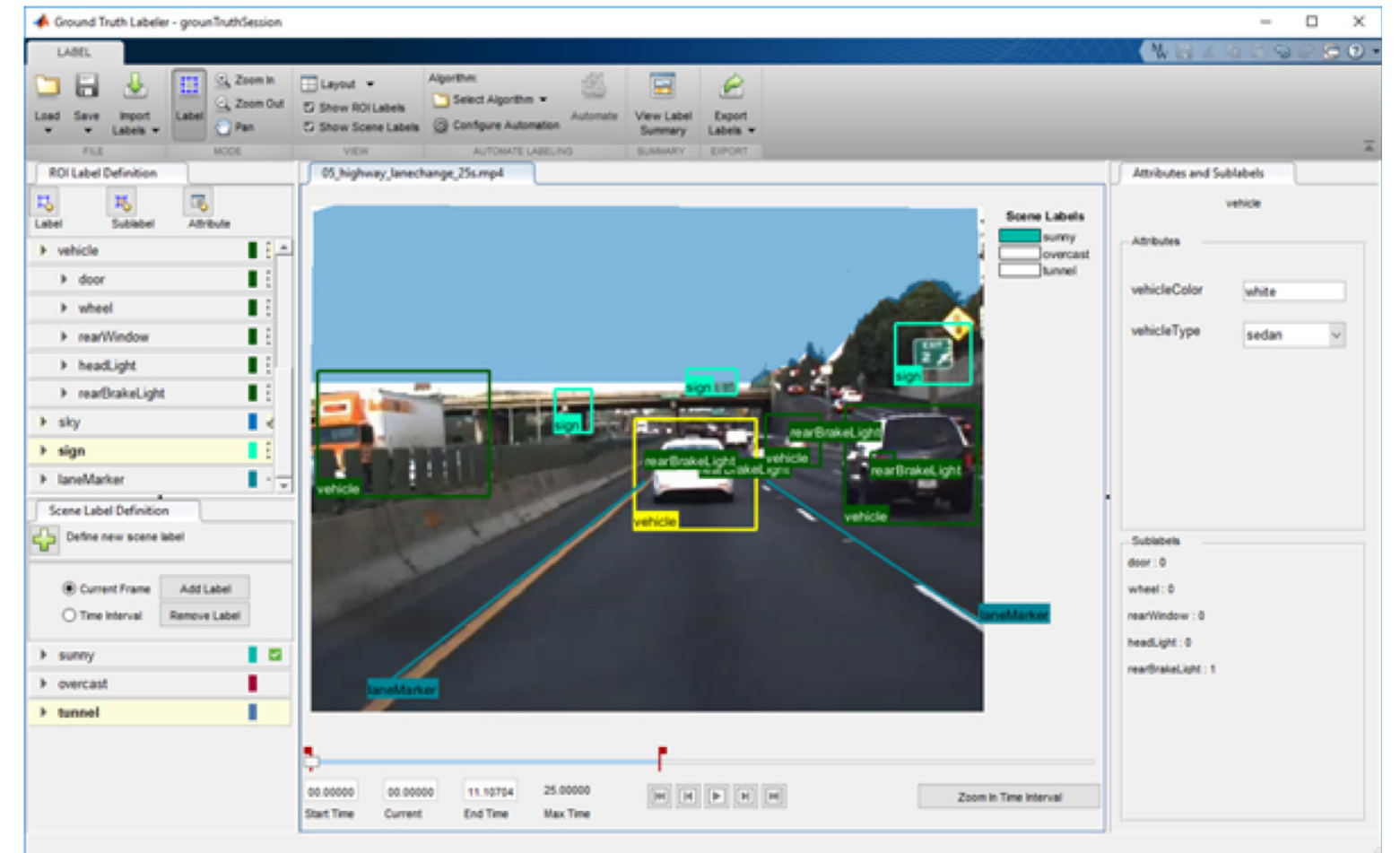
데이터: 데이터 레이블 지정

대량의 데이터에 소급적으로 레이블을 지정해야 하는 경우 이는 엄청난 시간이 소요될 수 있습니다.

딥러닝에 대한 레이블 지정은 시맨틱 세그멘테이션을 위해 이미지의 객체, 또는 이미지나 비디오의 각 픽셀에 있는 객체에 주석을 추가하는 것을 의미할 수 있습니다.

최근에는 이미지 및 신호 데이터 레이블 지정 전용 그래픽 툴이 제공되기도 합니다.

일부 툴은 이전 프레임의 레이블 및 객체를 기반으로 비디오 프레임의 레이블 지정을 추천하는 것과 같이 수동으로 레이블을 지정해야 하는 데이터의 양을 줄이려는 정교한 기술을 제공합니다.



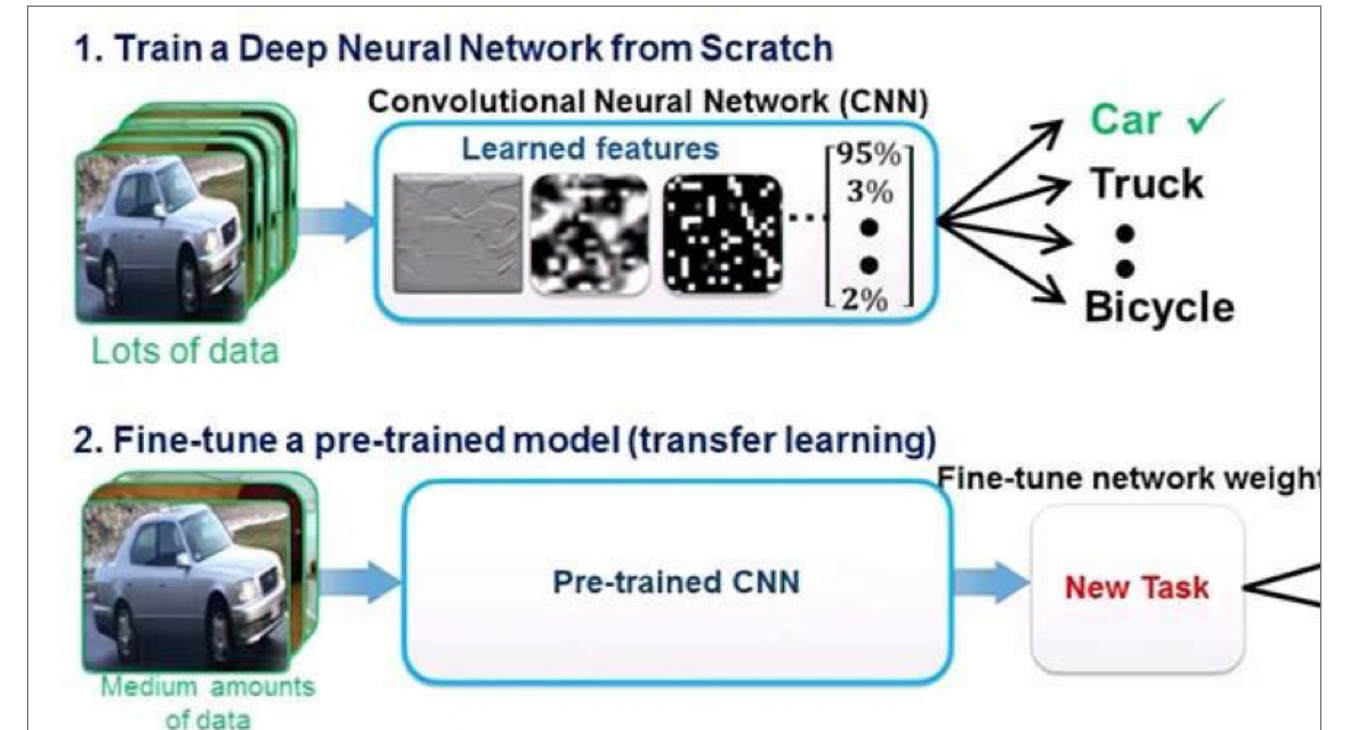
MATLAB의 Ground Truth Labeler 앱.

데이터: 데이터 레이블 지정

딥러닝을 사용하고자 하지만 레이블 지정된 데이터가 많지 않은 경우 GoogLeNet, ResNet-101 또는 VGG-16과 같은 사전 학습된 네트워크를 사용한 전이 학습을 고려해 보십시오.

전이 학습은 심층 신경망에 있는 소수의 파라미터 학습에 초점을 맞추기 때문에 소량의 레이블 지정된 데이터를 필요로 합니다.

MATLAB®에는 도메인에 따라 레이블 지정을 자동화하는 앱이 있는데, 그 예로 *Image Labeler*, *Ground Truth Labeler* 및 *Audio Labeler*가 있습니다. 이러한 앱은 전이 학습을 위해 새로 레이블이 지정된 데이터에 유용합니다.



▶ 전이 학습을 사용한 딥러닝 모델 교육(4:19)

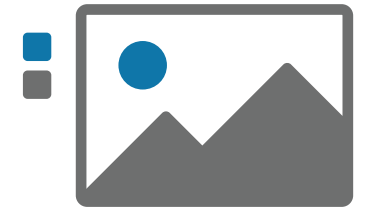
데이터: 데이터 레이블 지정

소량의 레이블 지정된 데이터를 처리하는 또 다른 접근 방식으로 이러한 데이터를 확장하는 방법이 있습니다. 예를 들어 이미지 데이터셋의 경우 레이블 지정된 이미지의 다양한 변형(예: 반사, 회전, 확대 및 변형)을 통해 학습 데이터를 증강하는 것이 일반적입니다.

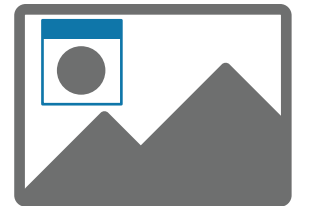
데이터 증강으로 더 많은 샘플이 제공될 수 있지만 샘플의 포괄적 표현 부족으로 인해 바이어스가 발생할 수 있음을 명심해야 합니다. 정확한 예측을 위해서는 알고리즘에 데이터에 대한 뛰어난 표현이 제공되어야 합니다.

명심할 사항...

MATLAB의 *Image Labeler*는 이미지를 수집할 때 실측 데이터의 레이블 지정에 도움이 됩니다. 직사각형 대상 영역(ROI) 레이블, 픽셀 ROI 레이블 및 장면 레이블을 정의합니다.



*Ground Truth Labeler*는 Image Labeler 앱과 동일한 방식으로 작동하지만 자율 주행 어플리케이션에 특화되어 있습니다.



*Audio Labeler*를 사용하여 실측 오디오 데이터를 영역 수준 및 파일 수준에서 모두 레이블 지정할 수 있습니다.





하드웨어

하드웨어

하드웨어를 2개의 그룹에서 생각하는 것이 유용합니다.

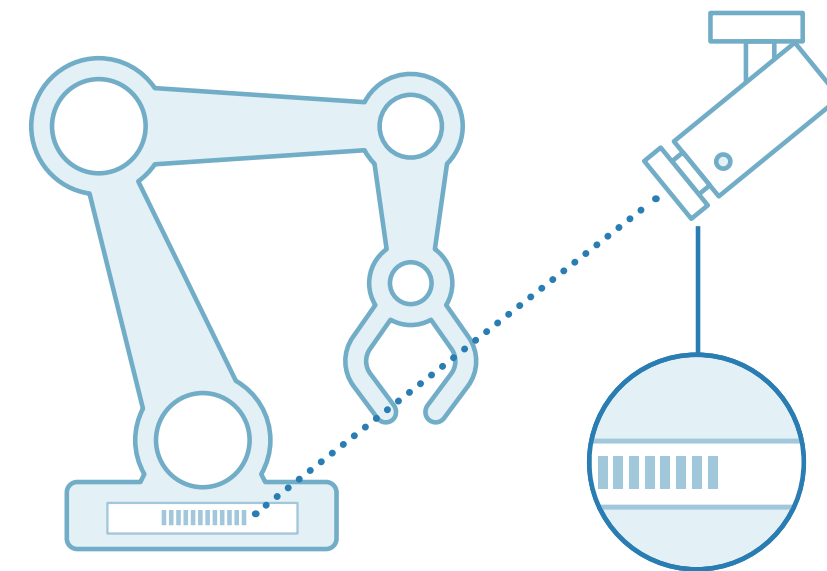
- 모델을 학습하는 데 이용할 수 있는 하드웨어.
- 프로덕션에서 모델이 실행되는 하드웨어.

알고리즘 추측

어느 석유 회사에서 유지 관리 예약을 위해 지리 태그가 지정된 기계 인벤토리를 추적하는 데 더욱 효율적인 방법을 만들었습니다. 이들은 머신 비전 시스템을 설정하여 일련번호가 포함된 태그를 식별하고, 객체 문자 인식을 사용하여 번호를 추출하고, 이미지와 인벤토리를 연결합니다.

이들이 사용한 알고리즘은 무엇입니까?

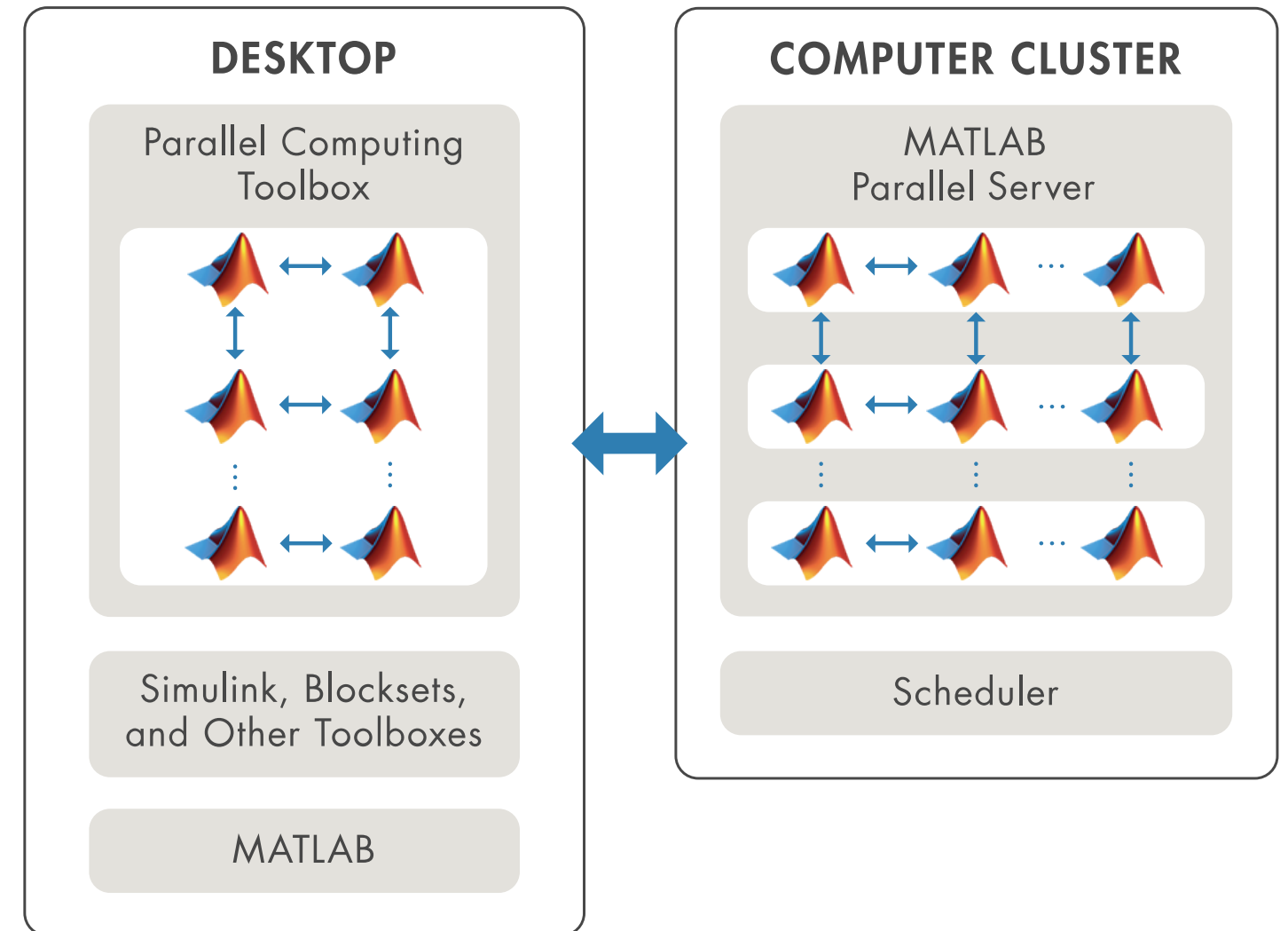
클러스터링 또는 지역 컨벌루션 뉴럴 네트워크(R-CNN)



하드웨어: 모델 학습

모델은 일반적으로 데스크톱 머신 또는 클러스터나 클라우드에서 학습됩니다. GPU는 대부분의 딥러닝 모델 학습에 있어 표준인데, CPU에서의 학습에 비해서 큰 속도 개선을 제공하기 때문입니다. 추가로 클러스터 또는 클라우드 옵션은 GPU를 구매하여 사용하는데 따른 고비용 문제 때문에, 이에 대비해서 널리 선택되고 있습니다. 이를 통해 여러 연구원이 하드웨어를 공유할 수 있습니다.

딥러닝 모델은 학습에 오랜 시간이 걸리기 때문에(약 몇 시간에서 며칠 소요) 여러 개의 모델 학습을 병행으로 두고 하나(또는 일부)에서 개선된 결과가 제공되기를 바라는 것이 일반적입니다.

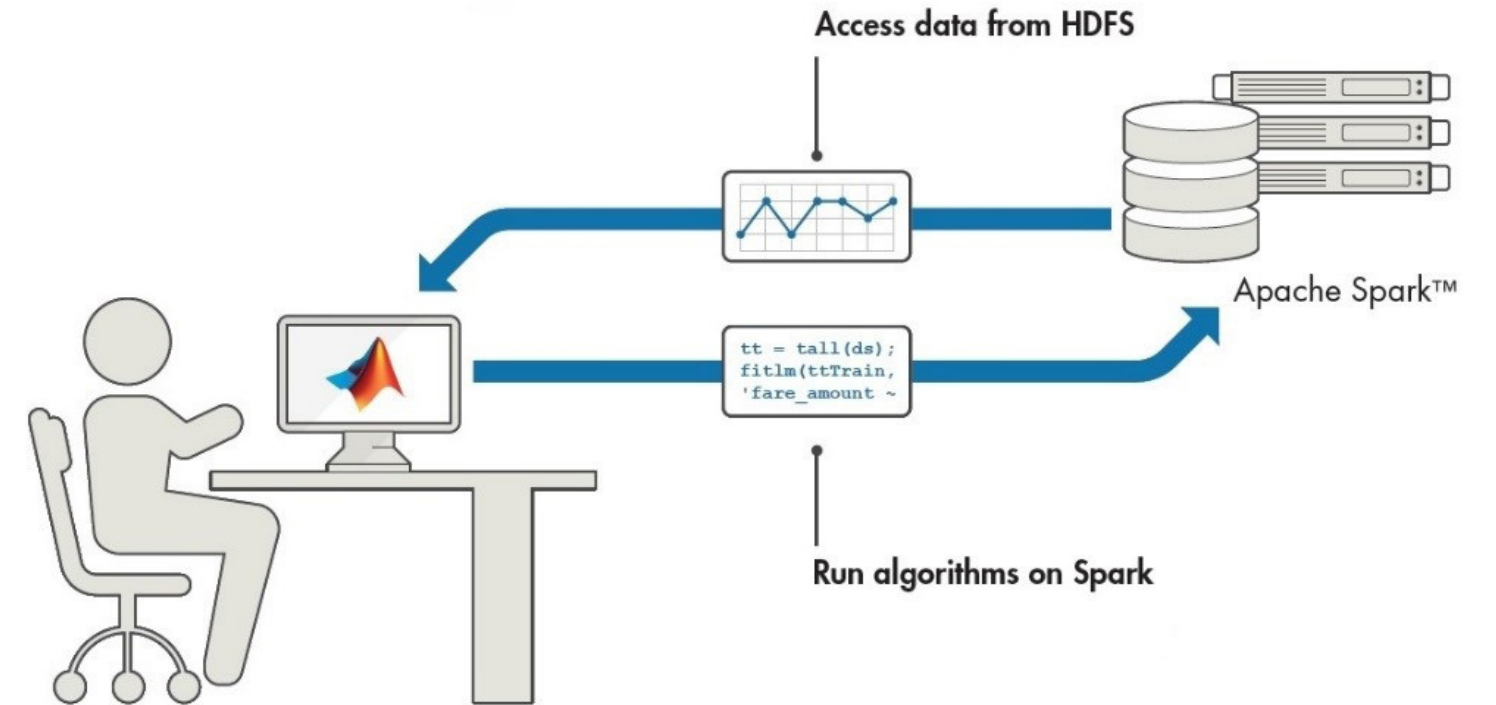


로컬 작업자를 통해 멀티코어 데스크톱에서 애플리케이션을 실행하고, GPU를 활용하고, Parallel Computing Toolbox로 클러스터를 확장합니다.

하드웨어: 모델 학습

데스크톱 CPU는 여전히 대부분의 머신 러닝 모델 학습에 효율적입니다.

대량의 데이터에서 모델을 학습하는 경우 Apache Spark™와 같은 빅 데이터 프레임워크를 사용하여 CPU 클러스터 전역에 걸쳐 계산을 분배합니다.



HDFS 및 Spark의 데이터에 MATLAB 사용.

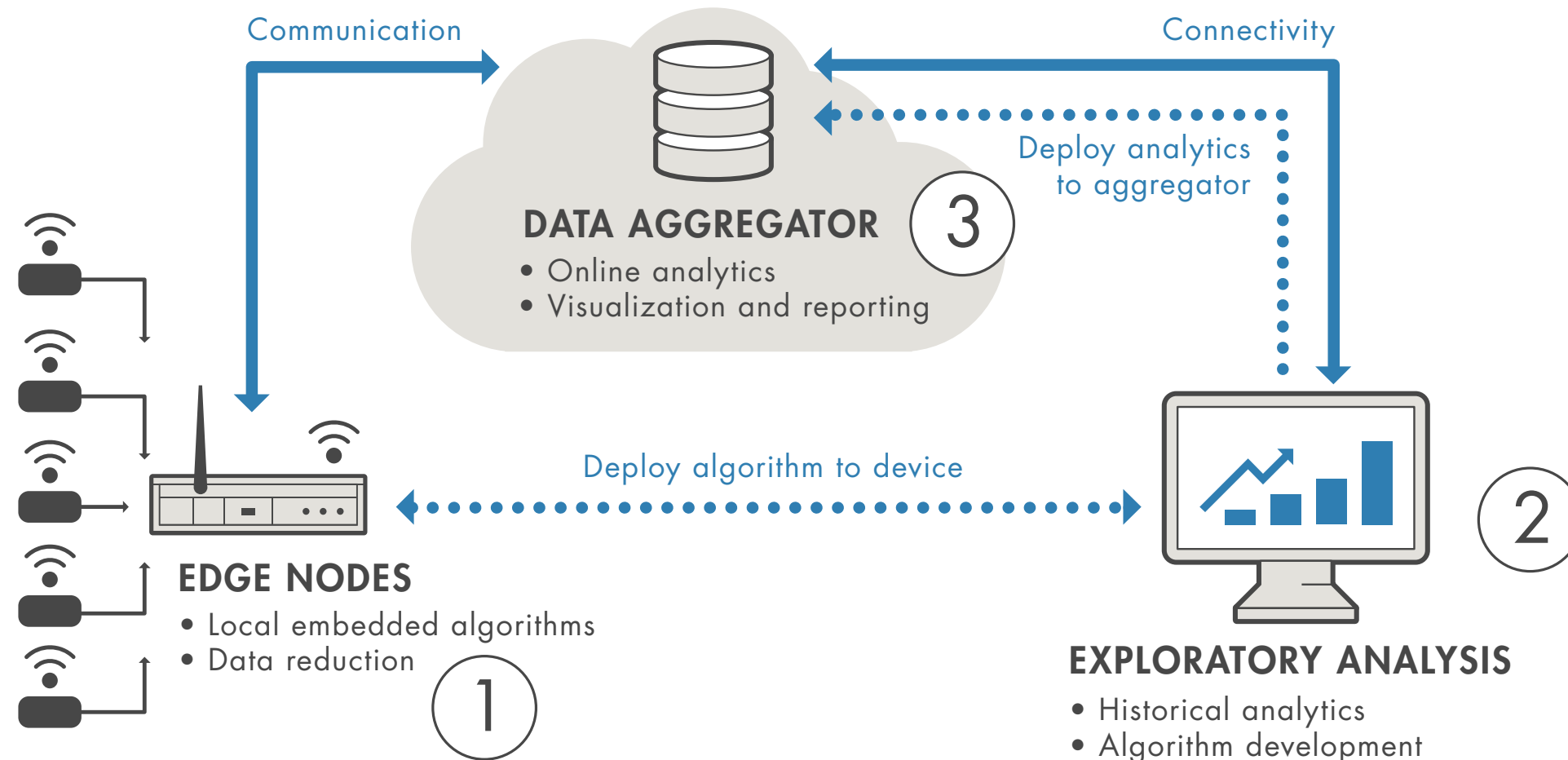
하드웨어: 프로덕션에서 모델 실행

프로덕션 시스템의 아키텍처를 구동하는 여러 요소가 있습니다.

- 시스템에서 얼마나 빨리 결과를 필요로 합니까?
- 네트워크 연결을 항상 이용할 수 있습니까?
- 에지 노드에서 이용 가능한 하드웨어는 무엇입니까?
- 모델이 얼마나 자주 업데이트되어야 합니까?

명심할 사항...

임베디드 시스템 설계자의 경우 알고리즘 성능뿐 아니라 전체적인 시스템 견고성, 안정성 및 아키텍처 및 설계 비용을 고려하는 것이 중요합니다.



하드웨어: 프로덕션에서 모델 실행

더욱 스마트하고 연결된 센서에 대한 트렌드로 인해 가급적 센서에 근접한 처리와 분석을 달성해야 한다는 압력이 더해지고 있습니다. 이 접근 방식에는 네트워크를 통해 전송되는 데이터의 양을 줄인다는 이점이 있습니다. 이를 통해 전송 비용이 감소하고, 무선 디바이스의 전력 소모량이 감소할 수 있습니다.

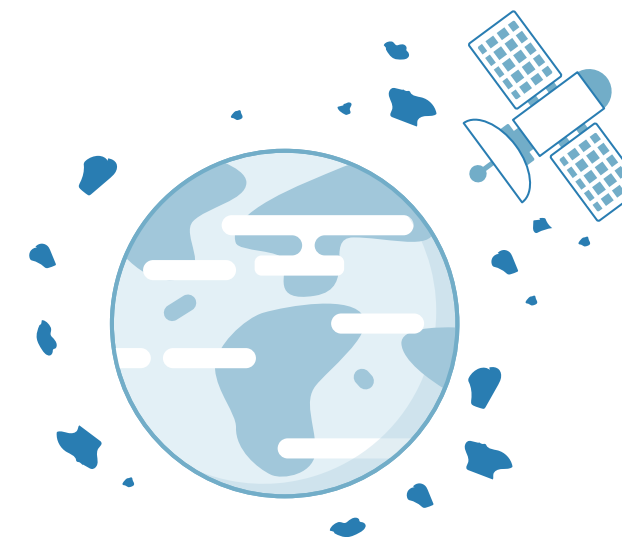
에지의 하드웨어에서 실행되는 모델은 네트워크 연결 없이 빠른 결과를 제공합니다. 하지만 머신 러닝 모델을 실행하려면 에지에 충분한 하드웨어가 제공되어야 하고, 모델이 중앙 서버에 상주하는 경우 모델을 업데이트하기 더욱 어렵습니다.

알고리즘 추측

저지구 궤도의 궤도 위성 및 우주선은 50만 개 이상의 우주 잔해로 인한 충돌 위험을 안고 있습니다. 이러한 잔해 추적을 통해 우주선은 충돌 지역 밖으로 이동할 수 있습니다. 현재 추적 전략은 비선형 천체 방해로 인해 계속해서 변하는 천체 역학 때문에 발생하는 우주 잔해 더미의 궤도 변화에 취약합니다.

안전을 개선하기 위해 사용할 수 있는 알고리즘의 종류는 무엇입니까?

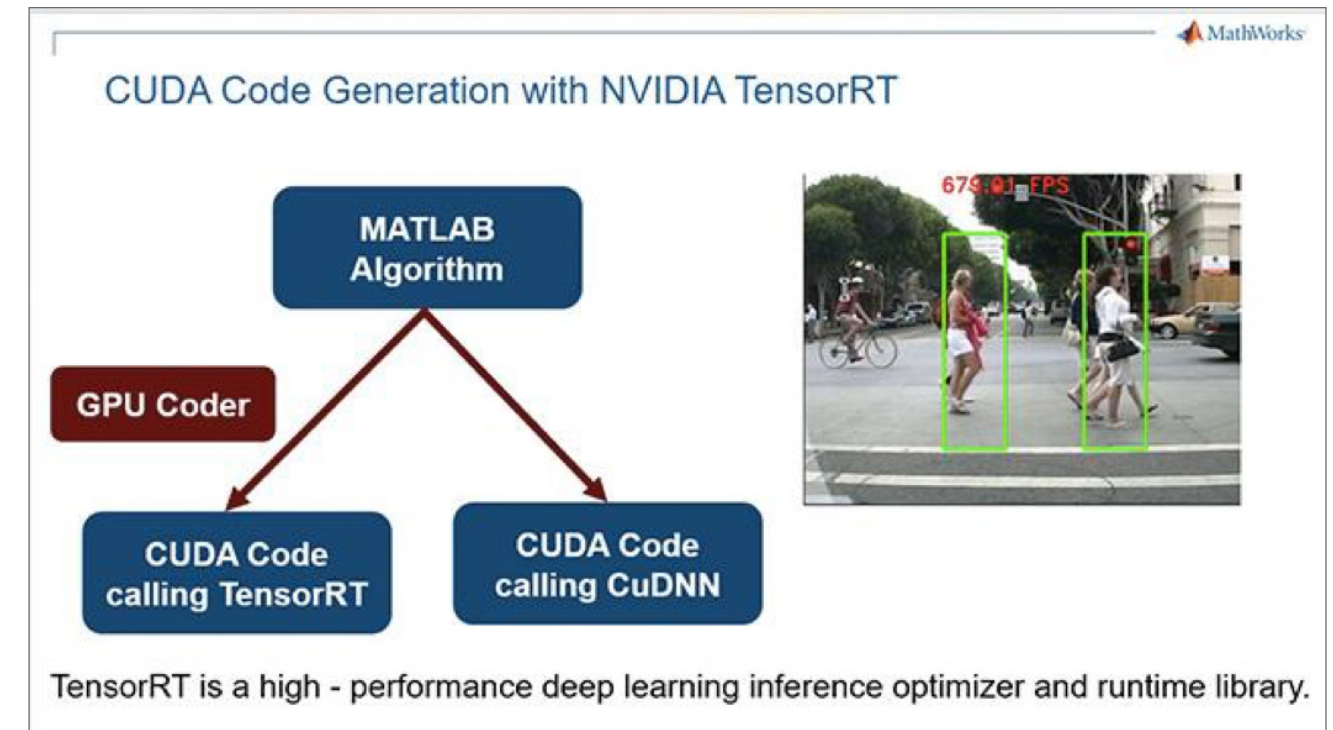
Deep Q 네트워크 또는 인공 신경망



하드웨어: 프로덕션에서 모델 실행

일반적으로 추상화 레벨이 높은 인터프리터 방식의 언어로부터 개발된 머신러닝 모델을 독립실행가능한 C/C++ 코드로 변환 가능하게 해 주는 도구들이 존재하는데, 이는 저전력 임베디드 디바이스에서 실행 가능합니다.

딥러닝 모델의 경우 높은 메모리 및 계산 요구 사항 때문에 특화된 하드웨어가 일반적으로 필요합니다. GPU Coder™를 통해 Intel® (MKL-DNN), NVIDIA®(TensorRT, cuDNN) 및 Arm®(Arm Compute Library)의 최적화된 라이브러리를 활용하는 코드 생성으로 고성능의 인퍼런스 속도로 배포 가능한 모델을 생성할 수 있습니다.



▶ *TensorRT가 포함된 NVIDIA GPU에서 보행자 감지(1:35)*

하드웨어: 프로덕션에서 모델 실행

NVIDIA GPU용 GPU Coder Support Package를 사용하여 생성된 CUDA 코드를 임베디드 GPU(예: NVIDIA Drive 플랫폼 또는 NVIDIA Jetson® 보드)의 독립형 애플리케이션으로 교차 컴파일 및 배포할 수 있습니다.

명심할 사항...

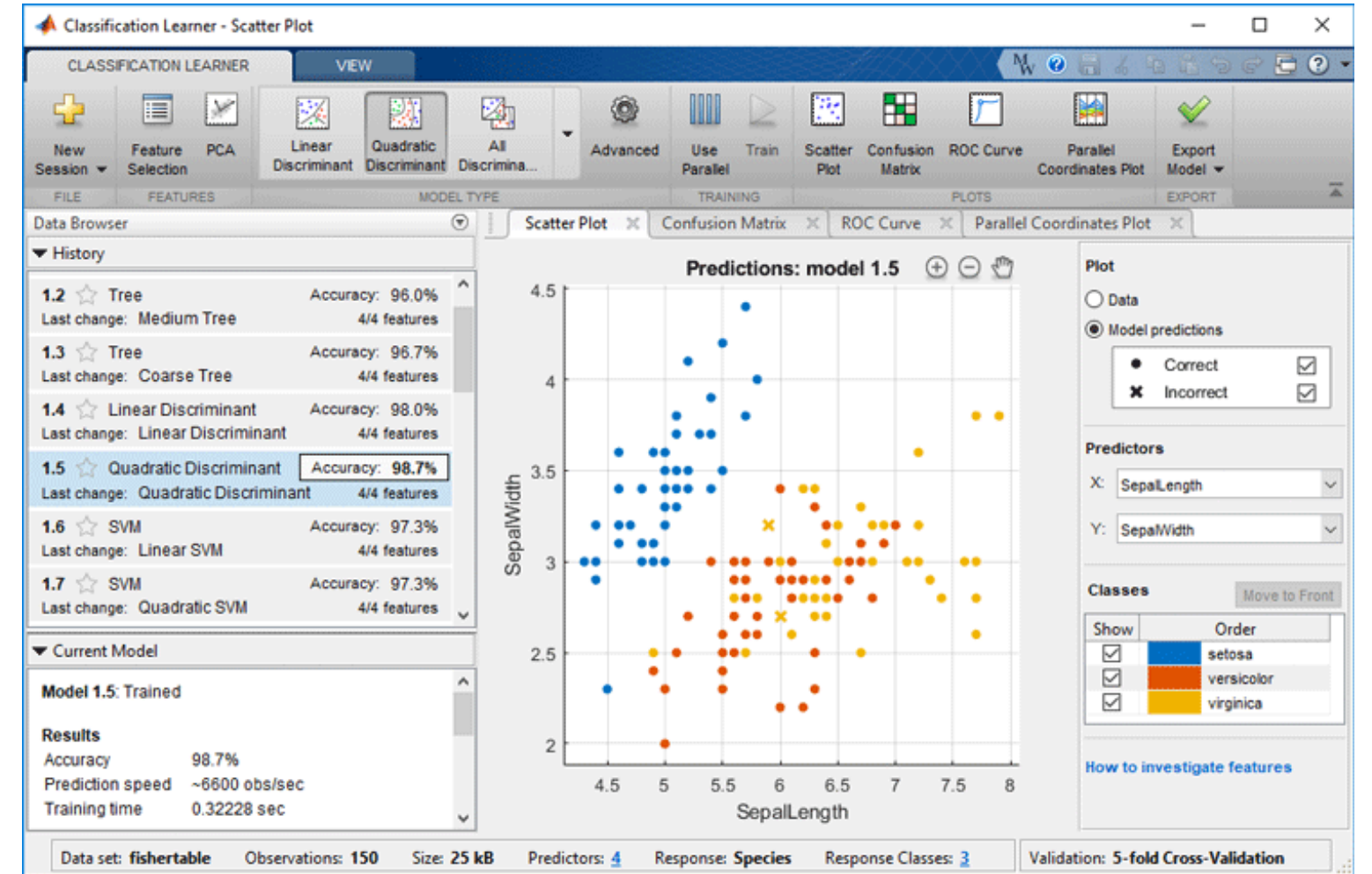
애플리케이션에 따라 프로덕션에서 모델을 사용하기 전 요구되는 유효성 검사의 수준이 크게 다릅니다. 예를 들어 마케팅에서 광고 추천에 사용되는 모델은 안전에 중요한 애플리케이션에 사용되는 모델에 비해 크게 낮은 수준의 유효성 검사를 필요로 합니다. 이러한 애플리케이션에서 모델은 HIL(Hardware-In-the-Loop)과 같은 기존 유효성 검사 프로세스와 통합되어 프로덕션 환경에서 모델이 예상한 것과 같이 실행되도록 할 수 있습니다.



하드웨어: 속도

얼마나 빨리 결과를 내야 합니까? 속도는 하드웨어, 데이터의 복잡성 및 크기, 사용하고자 하는 알고리즘의 영향을 받습니다.

빠르게 결과를 내야 하는 경우 머신 러닝 알고리즘을 먼저 사용해 보십시오. 일반적으로 학습 속도가 더욱 빠르고 더 적은 계산 능력을 필요로 합니다. 학습 시간의 주요 요소는 학습 데이터 내 변수와 관측값의 수입니다. 소량 데이터셋에 대한 단순 모델은 즉각적인 학습이 가능하지만 대량 데이터셋에 대한 복합 모델은 학습에 약 며칠이 걸릴 수 있습니다.

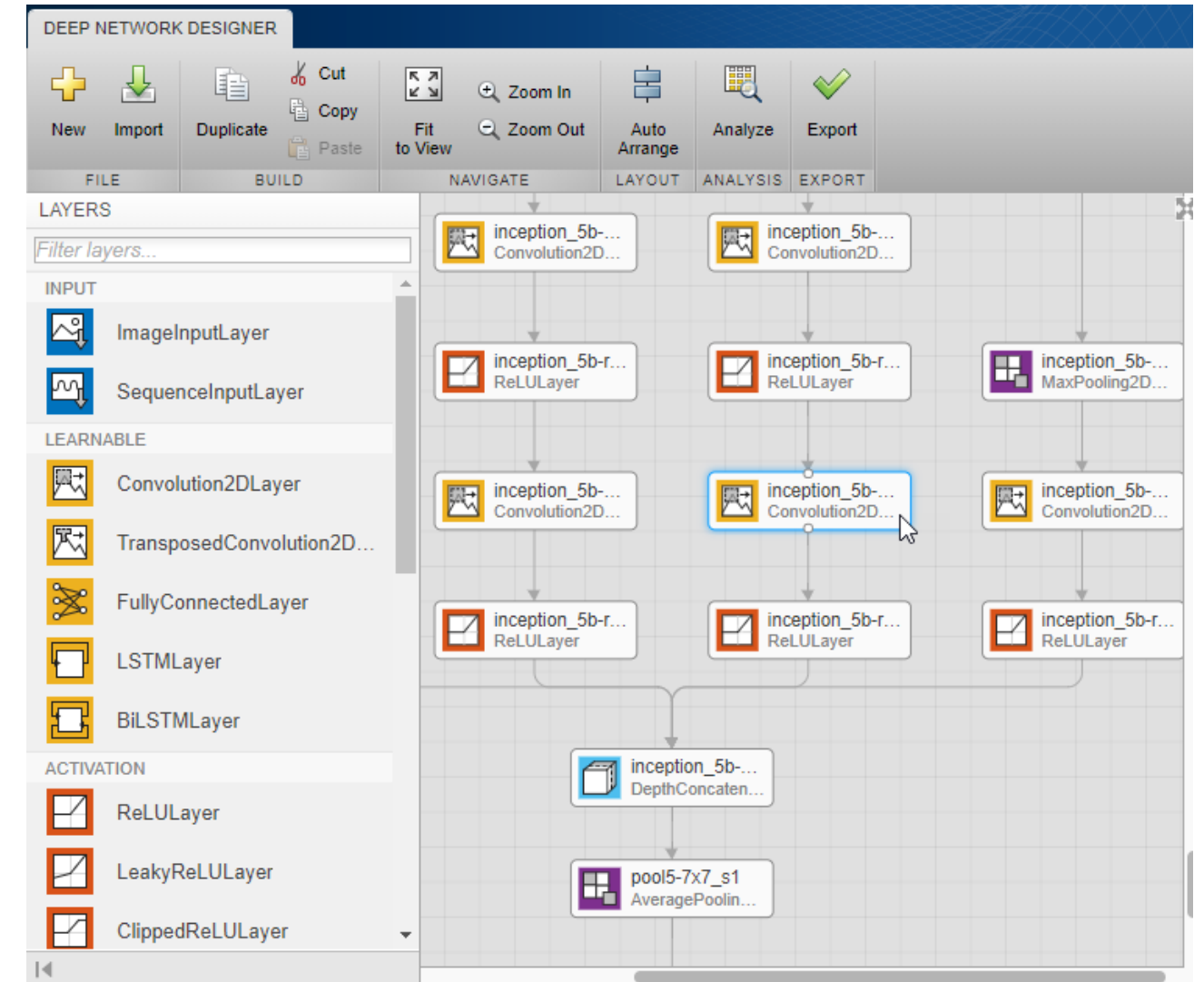


분류 학습기 앱을 사용하여 데이터셋에서 여러 가지 분류기를 빠르게 시도합니다.

하드웨어: 속도

딥러닝 모델의 경우 학습에 시간이 걸립니다. 사전 학습된 네트워크 및 퍼블릭 데이터셋이 많으면 전이 학습을 통해 딥러닝 모델 학습 시간이 감소하지만 학습 데이터를 이러한 네트워크로 통합하는 데 있어서의 현실적인 가능성을 과소평가하기 쉽습니다. 계층, 클래스 및 레이블을 수정해야 할 수 있습니다.

최신 딥러닝 알고리즘은 학습에 몇 분에서 몇 주가 소요될 수 있으며, 이 시간은 하드웨어와 계산 능력에 크게 좌우됩니다. 실무자가 여러 개의 딥러닝 모델 학습을 동시에 수행하는 것이 일반적입니다 (추가 하드웨어 필요).



딥 네트워크 디자이너 앱을 사용하여 딥러닝 네트워크를 빌드, 시각화 및 편집할 수 있습니다.

예시

AlexNet 및 CIFAR-10 데이터셋을 사용한 특징 추출

일부 애플리케이션은 딥러닝과 머신 러닝의 조합을 사용합니다. 이 MATLAB 사례는 사전 학습된 컨벌루션 뉴럴 네트워크를 사용하여 이미지에서 특징을 추출하고, 이러한 특징을 사용하여 서포트 벡터 머신(SVM)을 학습하는 방법을 안내합니다.

이 사례에 사용되는 이미지는 *CIFAR-10 데이터셋*의 이미지입니다.

» *MATLAB에서 데모 따라하기*



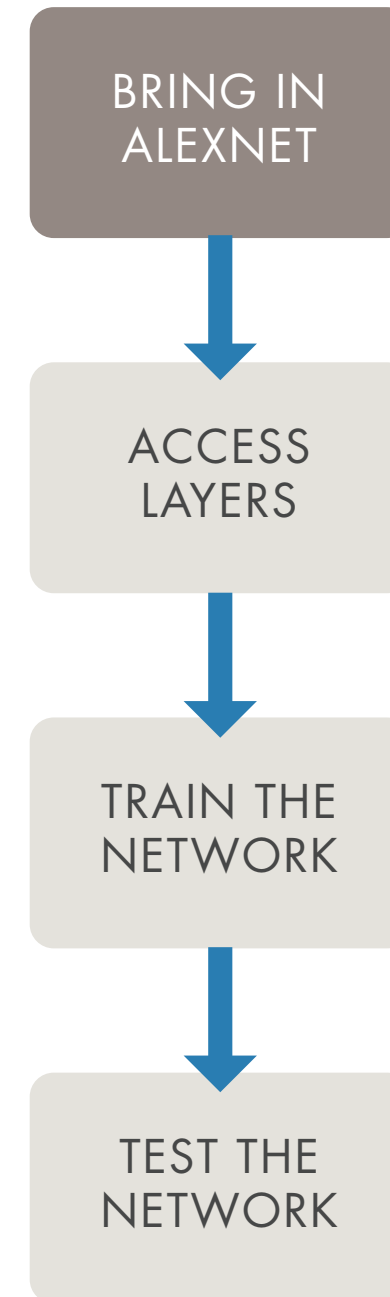
▶ *MATLAB에서 뉴럴 네트워크를 사용한 특징 추출(3:54)*

예시

1단계. Alexnet 가져오기

이 코드를 처음 실행하는 경우 코드에서 AlexNet 다운로드 [링크](#)가 포함된 오류 메시지가 표시될 수 있습니다. AlexNet은 한 번만 다운로드해야 합니다.

```
convnet = alexnet;
```



예시

2단계. 액세스 계층

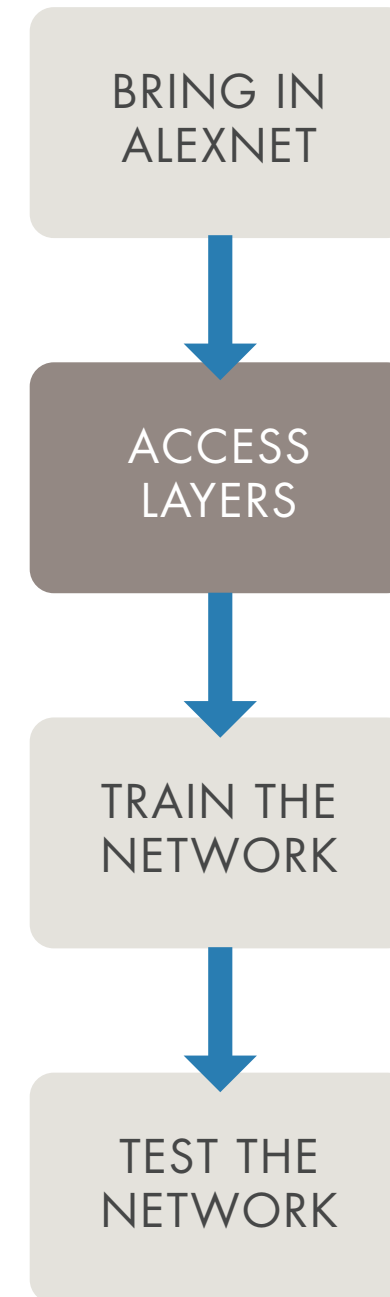
```
convnet.Layers % Take a look at the layers
ans =
 25x1 Layer array with layers:

 1  'data'      Image Input
    227x227x3 images with 'zerocenter'
    normalization

 2  'conv1'     Convolution
    96 11x11x3 convolutions with
    stride [4  4] and padding [0 0 0 0]

 ...

25  'output'    Classification Output
    crossentropyex with 'tench',
    'goldfish', and 998 other classes
```



예시

3단계. 학습 학습 데이터 설정

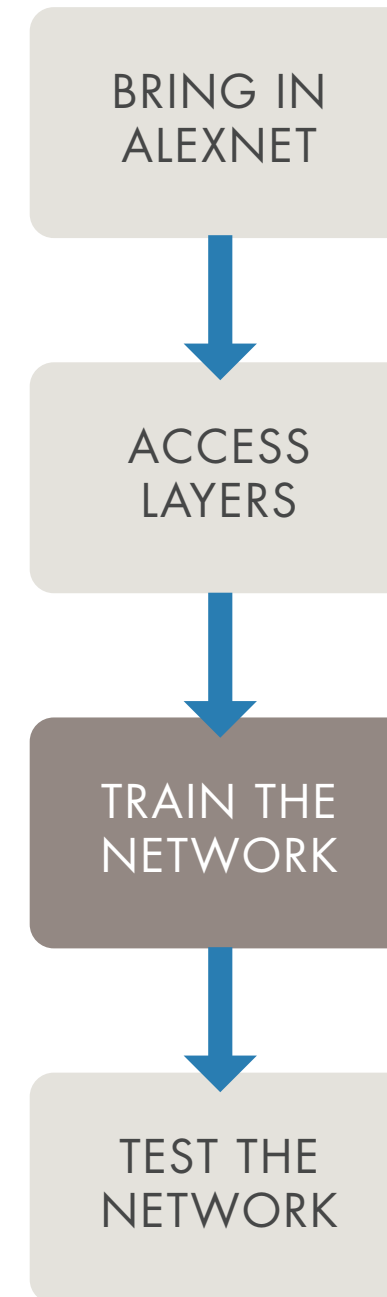
CIFAR-10을 사용 중인 경우 10개의 객체 범주 중에서 선택할 수 있습니다. 이 예시의 범주는 임의로 선택됩니다. 하지만 가장 적합한 범주를 선택할 수 있습니다.

```
rootFolder = 'cifar10Train';  
categories = {'Deer', 'Dog', 'Frog', 'Cat'};  
imds = imageDatastore(fullfile(rootFolder,  
    categories), 'LabelSource', 'foldernames');  
imds.ReadFcn = @readFunctionTrain;  
[trainingSet, ~] = splitEachLabel(imds, 50,  
    'randomize');
```

시도할 사항:

숫자 50을 사용하고자 하는 학습 이미지 수로 변경합니다.

이미지 수가 증가함에 따라 분류기의 정확성이 어떻게 변화하는지 확인합니다.



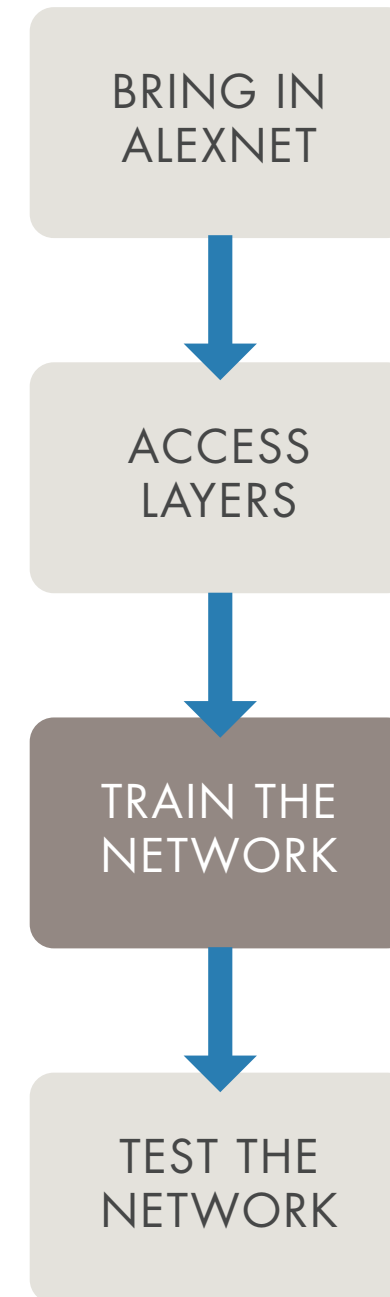
예시

3단계. 학습

학습 세트 이미지에서 특징 추출

특징이 활성화를 통해 추출되고, CNN에서 아키텍처의 해당 지점으로 학습된 특징을 가져옵니다. AlexNet과 같이 수백만 개의 이미지로 학습된 네트워크를 사용하는 경우 네트워크에서 가져온 특징이 객체를 설명하는 풍부하고 복잡한 특징일 것으로 예상할 수 있습니다.

```
featureLayer = 'fc7';  
trainingFeatures = activations(convnet,  
    trainingSet, featureLayer);
```



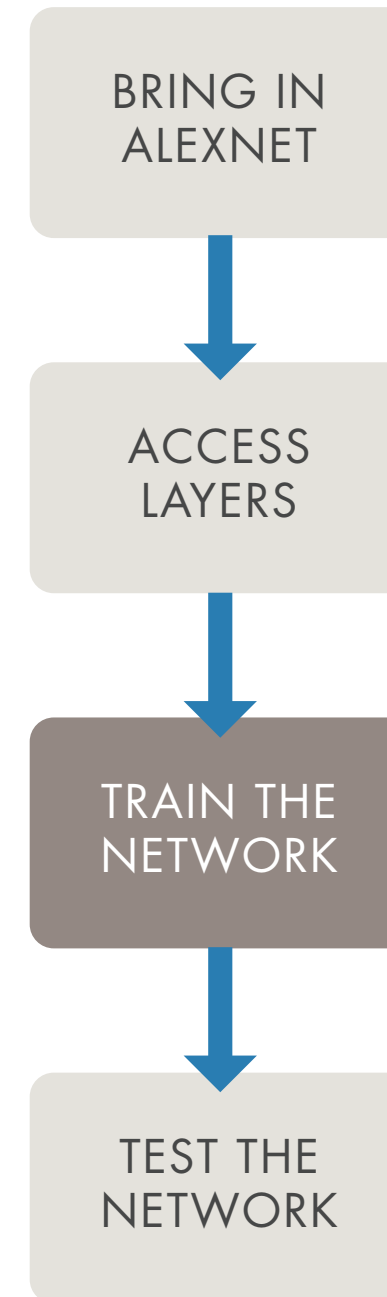
예시

3단계. 학습 SVM 분류기 학습

`fitcecoc`의 경우 사용할 수 있는 여러 분류기(Classifier) 중 하나입니다. 오류 정정 출력 코드를 사용하여 멀티클래스 SVM을 피팅합니다. `fitcknn` 또는 `fitcnb` 등과 같은 다른 피팅 함수는 어떻습니까?

최적의 분류기를 찾는 포괄적 접근 방식으로 분류 학습기 앱을 사용해 보십시오.

```
classifier = fitcecoc(trainingFeatures,  
                    trainingSet.Labels);
```



예시

4단계. 테스트

테스트 데이터 설정

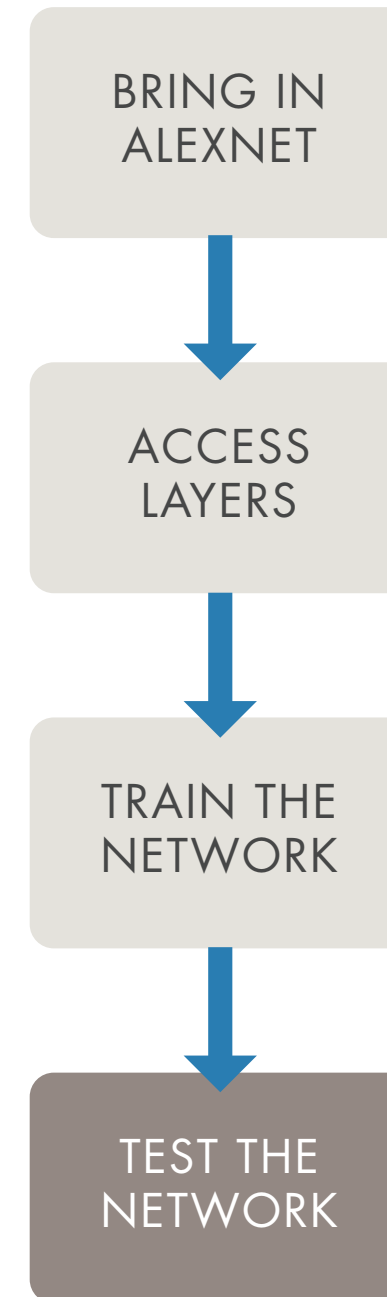
```
rootFolder = 'cifar10Test';  
testSet = imageDatastore(fullfile(rootFolder,  
    categories), 'LabelSource', 'foldernames');  
testSet.ReadFcn = @readFunctionTrain;
```

테스트 세트 이미지에서 특징 추출, SVM 분류기 테스트

```
testFeatures = activations(convnet, testSet,  
    featureLayer);  
predictedLabels = predict(classifier, testFeatures);
```

전반적 정확도 판단

```
confMat = confusionmat(testSet.Labels,  
    predictedLabels);  
confMat = confMat./sum(confMat,2);  
mean(diag(confMat))
```





결론

결론

지금까지 살펴본 것처럼 프로젝트에 대한 최적의 알고리즘을 선택하는데 있어 불변의 규칙은 거의 없습니다.

대부분의 알고리즘은 특정 상황에서 최적으로 작동하는 것을 확인하는 시행착오적인 과정을 통해 선택됩니다.

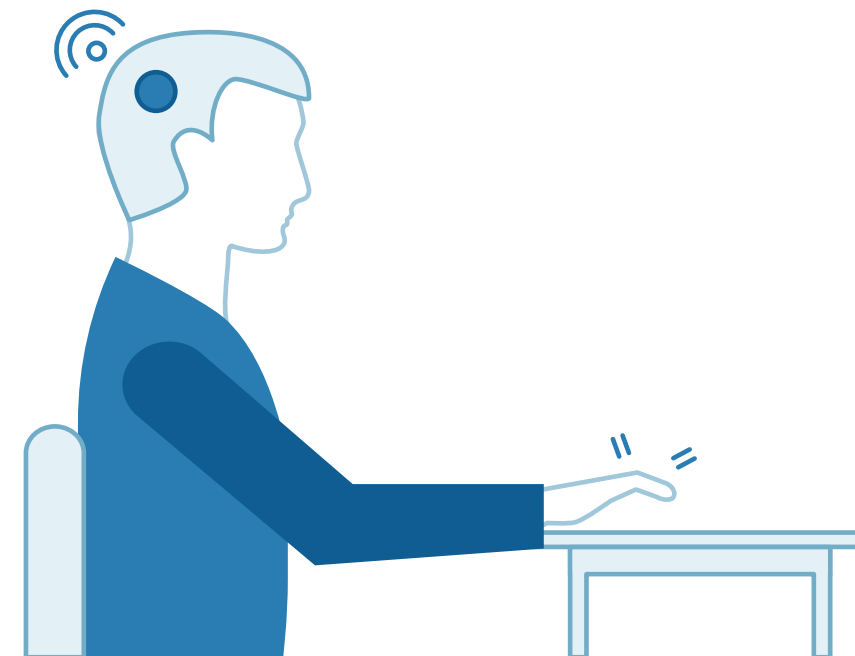
기존 머신 러닝 알고리즘 또는 딥러닝 알고리즘을 선택하는 것과 무관하게 MATLAB은 이러한 기술을 빠르게 시작할 수 있도록 도구와 다양한 지원을 제공합니다.

알고리즘 추측

연구 및 개발 조직에서 사지 마비 환자의 뇌에 이식한 전극 배열에서의 신호 처리를 통해 환자의 팔과 손 제어 능력을 복원했습니다.

이들이 사용한 알고리즘은 무엇입니까?

SVM 또는 RNN

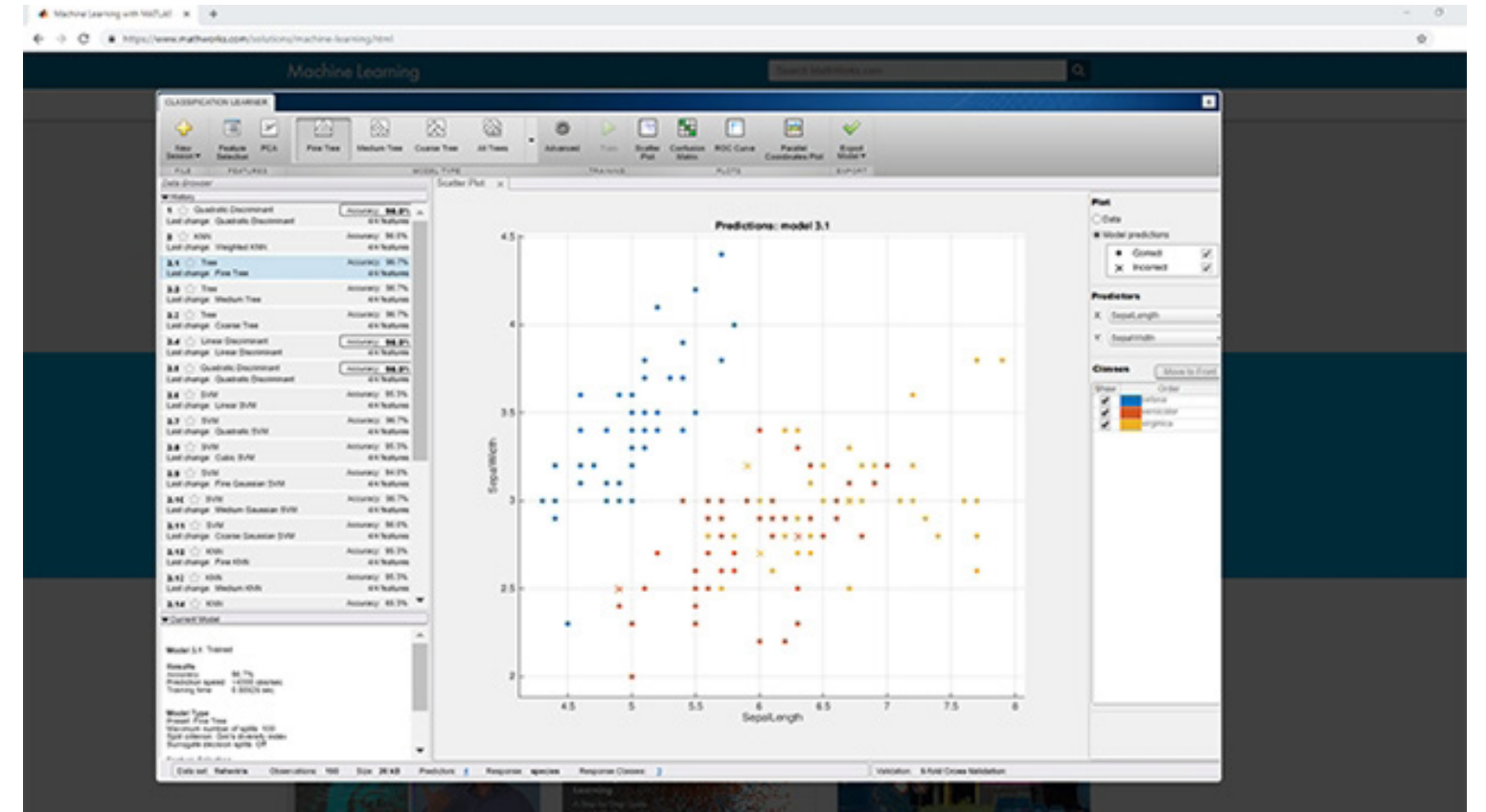


결론

머신 러닝

엔지니어 및 다른 분야의 전문가들은 MATLAB을 사용하여 예측 유지보수, 센서 분석, 금융, 통신 전자를 위한 수천 개의 응용 프로그램을 배포해 왔습니다. MATLAB을 이용하면 다음을 통해 머신 러닝의 어려운 부분이 용이해집니다.

- 모델 학습 및 비교를 위한 Point-and-Click 앱
- 모델 성능을 최적화하기 위한 자동 하이퍼파라미터 튜닝 및 특징 선택
- 지도 학습 및 비지도 학습을 위한 모든 종류의 널리 사용되는 분류, 회귀, 클러스터링 알고리즘
- 대부분의 통계 및 머신 러닝 연산에서 오픈 소스보다 더욱 빠른 실행



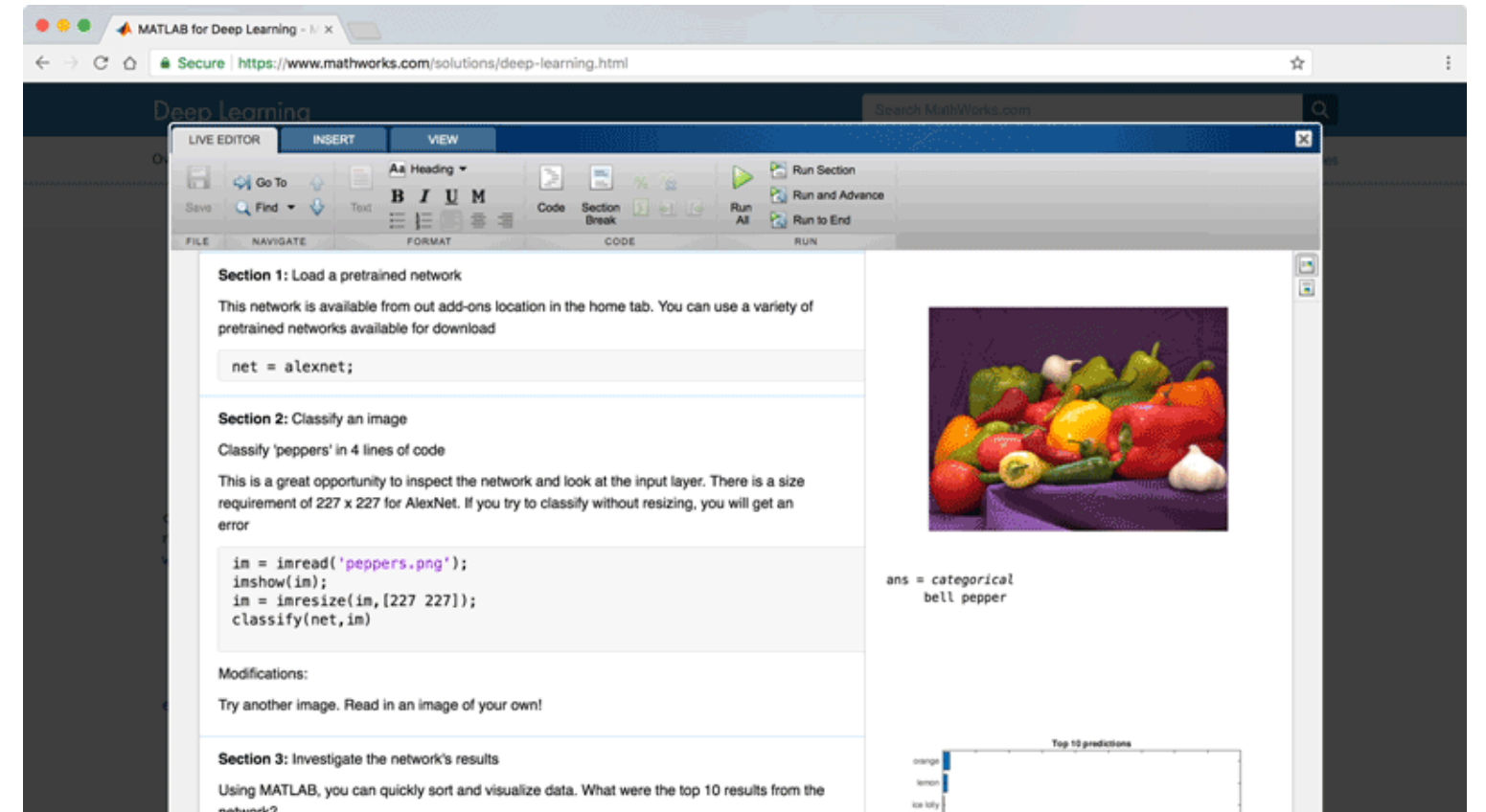
브라우저에서 Classification Learner 앱 사용해보기

결론

딥러닝

전문가가 아니어도 MATLAB 코드 몇 줄만으로 딥러닝 모델을 빌드할 수 있습니다. MATLAB으로 딥러닝 작업을 수행하는 방법을 알아보십시오.

- 최신 사전 학습 모델에 손쉽게 액세스
- 앱을 사용하여 실측데이터 (Ground-Truth)의 레이블 지정 자동화
- NVIDIA GPU, 클라우드 및 데이터센터 리소스의 알고리즘 가속화
- 복잡한 심층 신경망 아키텍처 생성, 수정 및 분석
- Caffe와 TensorFlow-Keras의 모델을 불러와서 작업
- PyTorch 및 Apache MxNet™과 같은 프레임워크 사용, ONNX™ 모델 임포터 및 익스포터에 대한 지원 사용을 통한 동료와의 협업



브라우저에서 딥러닝 사용해보기

알고리즘 추측: 답변

질문	답변	추가 정보
연구원은 무엇을 사용하여 CT 스캔의 방사선을 줄였습니까?	컨벌루션 뉴럴 네트워크	» 기사 읽기
가전제품 업체에서 무엇을 사용하여 새 스피커를 판단했습니까?	SVM(서포트 벡터 머신)	» 기사 읽기
석유 회사에서 이미지 내 재고를 식별하는 데 무엇을 사용했습니까?	지역 컨벌루션 뉴럴 네트워크	» 프레젠테이션 보기
연구원은 우주 쓰레기의 경로를 어떻게 예측했습니까?	인공 신경망	» 프레젠테이션 보기
R&D 조직은 무엇을 사용하여 수족 조절을 복원했습니까?	SVM(서포트 벡터 머신)	» 사례 읽기