

사랑은 언제까지나 떨어지지 아니하되 예언도 폐하고 방언도 그치고 지식도 폐하리라. (고린도전서 13:4)

Love never fails. But where there are prophecies, they will cease; where there are tongues, the will be stilled; where there is knowledge, it will pass away. (1 Corinthians 13:4)

파이썬으로 배우는 기계학습(Machine Learning with Python)

Lecture Notes by Youngsup Kim, idebtor@gmail.com, Handong Global University

NOTE: The following materials have been compiled and adapted from the numerous sources including my own. Please help me to keep this tutorial up-to-date by reporting any issues or questions. Send any comments or criticisms to idebtor@gmail.com. Your assistances and comments will be appreciated.

들어가면서

"파이썬으로 배우는 기계학습" 강의에 오신 여러분을 환영합니다.

본 강의는 대학교 1학년 학생을 대상으로 개발된 과정입니다. 고등학교에서 인문사회계열 과정을 마친 학생들이면 수강이 가능하도록 설계하였습니다. 다만, 고교 교육 과정에서 다루지 않는 기본적인 파이썬 프로그래밍 지식은 필요합니다. 기계학습(Machine Learning) 혹은 딥러닝(Deep Learning)에서는 다른 프로그램 언어보다 파이썬을 많이 사용하므로, 아직 파이썬 프로그래밍의 경험이 없는 학생들은 본 강의를 수강하기 전에 파이썬 프로그래밍을 공부하거나 이미 프로그래밍 지식이 있는 학생들은 파이썬 학습을 필히 병행할 것을 권장합니다.

본 과정에서 다루는 기계학습의 원리를 배우기 위해 필요한 기본적인 행렬 연산과 미분, 추가적인 파이썬 넘파이 프로그래밍에 대한 강의들은 준비되어 있습니다.

최근에는 다양한 채널을 통해 탁월한 기계 학습을 강의들을 얼마든지 접할 수 있습니다. 그런데, 이 강의를 제공하게 된 것은 제 자신이 기계학습을 공부할 때 혹은 학생들을 가르치다가 겪은 어려움을 학생들과 나누면서 더 많은 사람들이 기계학습을 시작할 수 있도록 도울 수 있다면 좋겠다는 동기와 목적이 있습니다. 어떤 탁월한 강의는 기존의 개발된 라이브러리 코드 중심으로 활용에 중심을 두고 있습니다. 어떤 강의들은 많은 수학적 지식을 바탕으로 알고리즘 중심으로 그 원리를 설명하면서 진행합니다. 이미 구현된 라이브러리 함수들을 많이 사용하기 때문에 실제 알고리즘이 어떻게 작동이 되는 이해하기 힘듭니다. 기계학습으로 실제적인 문제를 해결하고자 할 때 어려움을 겪습니다. 또한 알고리즘 중심의 강의인 경우는 내용을 이해할 수 있을지라도 어떻게 구현하는지 몰라서 막막할 때가 많습니다. 대학에서 이러한 경험과 시행 착오를 거친 후에, 대학 입학생 정도의 수학의 기초지식과 기본적인 파이썬 프로그래밍 개념을 알고 있는 학생들을 대상으로 기계학습 강좌를 개발하게 되었습니다.

본 과목은 수강생이 기계학습(machine learning)의 원리를 이해하고, 그 원리를 적용하여 문제를 해결하는 역량을 키우기 위한 것입니다. 또한 4차 산업 혁명의 주요 부분을 차지하고 있는 소프트웨어와 인공지능의 기술적 시대의 흐름에 대비한 계산적 사고력과 창의력을 향상하는 것이 목적입니다. 본 강의는 기계학습의 원리를 이해하기 위해, 일차 방정식과 한 개의 인공 뉴론으로 시작합니다. 단계별로 다양한 기계학습 알고리즘을 배우면서, 그에 걸맞는 인공신경망을 하나씩 모두 구현해 봅니다. 다양한 학습자료들을 다루어 보면서 일어나는 기계학습의 문제들을 다루기도 합니다. 단순한 퍼셉트론에서 시작하여, 아달라인 퍼셉트론, XOR 3층 신경망, MNIST 신경망, 끝으로 딥러닝을 시작할 수 있는 심층신경망까지 구현합니다.

본 강의를 다 마친다면, 더 높은 수준의 기계학습 공부할 수 기본 지식을 갖추게 되는 것을 물론이고, 기계학습 오픈 프레임워크로 널리 알려진 텐서플로(Tensorflow) & 케라스(Keras), 파이토치(PyTorch)와 같은 플랫폼도 어렵지 않게 사용할 수 있게 됩니다. 또한,

- **CNN**(Convolution Neural Network, 합성곱신경망),
- **RNN**(Recurrent Neural Network, 순환신경망),
- **NLP**(Natural Language Processing, 자연어처리), 그리고
- **GAN**(Generative Adversarial Nets, 적대적 생성망)과 같은

다양하고 복잡한 딥러닝 알고리즘 모델들도 이해할 수 있는 기초 지식을 갖게 될 것입니다.

주차별 강의 주제

1주차

1. 기계학습 소개
2. 기계학습 개발환경
3. 행렬

2주차

1. 함수와 뉴론
2. 넘파이 튜토리얼 - 배열의 형상
3. 넘파이 튜토리얼 - 브로드캐스팅

3주차

1. 인공뉴론의 동작원리
2. 미분
3. 활성화 함수

4주차

1. 퍼셉트론 (Perceptron)
2. 퍼셉트론 알고리즘
3. 퍼셉트론 코딩

5주차

1. 기계학습 작업흐름도 - 학습자료 준비
2. 기계학습 작업흐름도 - 학습자료 전처리
3. 객체지향 프로그래밍

6주차

1. 객체지향 퍼셉트론 구현
2. 객체지향 퍼셉트론 활용
3. 다층 퍼셉트론

7주차

1. 순방향 신경망
2. 순방향 신경망 실습
3. 아달라인과 경사하강법

8주차

중간고사

9주차

1. 아달라인과 경사하강법 구현
2. 아달라인과 경사하강법 적용
3. 역전파 오차 계산법

10주차

1. 역전파 오차함수 미분
2. XOR 신경망 모델링
3. XOR 신경망 구현

11주차

1. 다층신경망 모델링
2. 로지스틱 회귀 알고리즘
3. 로지스틱 회귀 비용함수 미분

12주차

1. 로지스틱 회귀 신경망 구현
2. MNIST Dataset
3. 배치/확률적 경사하강법

13주차

1. 미니배치 경사하강법
2. 심층신경망 구현
3. 심층신경망 성능 테스트

14주차

1. 기계학습 오픈 프레임워크 - TensorFlow&Keras
2. 기계학습 오픈 프레임워크 - PyTorch

제 1-1 강: 기계학습 소개

학습목표

- 기계학습에 대한 기본적 이해

학습 내용

- 기계학습의 개념
- 기계학습의 정의
- 기계학습의 종류

기계학습의 개념

컴퓨터는 계산하는 것을 아주 잘 합니다. 정확하고 아주 빠르게 합니다. 사람보다 훨씬 빠릅니다. 그런데, 컴퓨터가 사람들보다 잘하지 못하는 것들이 있었습니다. 어떤 것이 있을까요? 예를 들면, 여러분이 다음의 곱셈을 하는 것과 이미지를 인식하는 것 중에서 어느 것을 더 쉽게 할 수 있을까요?

$$\begin{array}{r}
 3.14159\ 26535 \\
 \times 2.71828\ 18284 \\
 \times 1.41421\ 35623 \\
 \times 1.61803\ 39887 \\
 \hline
 \end{array}$$

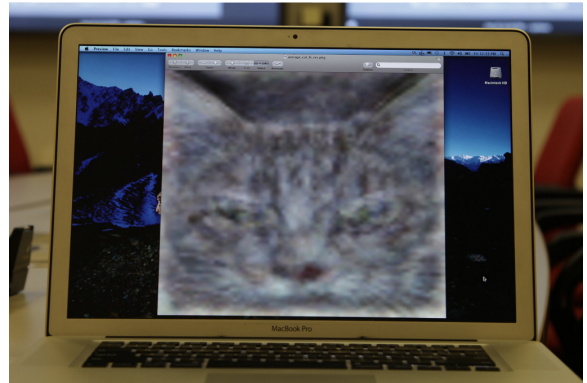


그림 1: 곱셈과 이미지 인식(사진 출처:Jim Wilson, The New York Times, 2012)

여러분은 곱셈보다는 사람과 고양이를 더 쉽게 알아 볼 수 있으며, 컴퓨터는 사람과 고양이를 알아 보기보다는 곱셈을 더 잘 할 수 있었습니다. 어린 아이는 새를 쉽게 인식 할 수 있지만, 지난 50년 이상 컴퓨터 과학자들은 컴퓨터로 물체를 인식하는 방법을 알아내기 위해 노력했으나 아주 힘들었습니다. 그래서 컴퓨터를 조롱하는 이런 만화까지 있었습니다.



그림 2: 컴퓨터가 할 수 있는 일과 할 수 없는 일(출처: <https://xkcd.com/1425/>)

그 둘의 차이는 무엇일까요? 컴퓨터는 왜 곱셈은 우리보다 더욱 잘하지만, 고양이는 알아보지 못하는 걸까요? 이를 알기 위해 컴퓨터가 정말 잘하는 것, 못하는 것이 무엇인지 생각해봐야 합니다. 컴퓨터가 가장 잘하는 것은 덧셈과 뺄셈을 엄청나게 빠른 속도로 연산하는 것입니다. 만약 2GHz 쿼드코어 CPU를 탑재한 컴퓨터라면, 1초의 80억번의 연산을 수행할 수 있습니다. 곱셈은 결국 덧셈의 반복일 뿐입니다. 따라서 곱셈과 같은 단순하면서 반복되는 작업은 컴퓨터가 사람이 처리하는 것에 비해 엄청난 속도로 일을 수행할 수 있습니다.

그러나 컴퓨터의 단점은 만들어진 대로만 동작한다는 것입니다. 사람이 컴퓨터가 어떤 동작을 하기 원할 때, 머리 속에는 컴퓨터가 어떤 상황에서 일할지에 대한 가정을 생각하고 있습니다. 즉 입력 값을 미리 생각하고 있는 것이지요. 다양한 입력 값을 처리하게 하려면 기계는 더욱 복잡해 질 수 밖에 없습니다. 문제는 여기서 발생합니다. 곱셈이나 덧셈, 사칙연산과 같은 단순 반복된 작업은 들어오는 입력 값이 한정되어 있습니다. 따라서 프로그래머가 컴퓨터가 어떤 동작을 하도록 만드는데 별로 어려움을 겪지 않습니다. 그러나 고양이 사진과 같은 영상 파일의 경우, 너무나도 다양한 입력 값이 발생하게 됩니다. 예를 들어, 고양이가 한쪽 눈을 찡그리더라도 고양이로 인식해야 하며, 털의 색깔이 갈색이 아닌 회색이라도 고양이로 인식해야 합니다. 더욱 단적인 예로 픽셀 값 하나만 바뀌어도 고양이로 인식해야 합니다. 프로그래머는 이 입력 값을 고려하여 프로그램을 작성해야 하며, 이는 불가능한 일입니다. 그러니 지난 50년간 컴퓨터로 물체를 인식하는 방법을 알아내기 힘들었던 것입니다.

$$\begin{array}{r}
 3.14159\ 26535 \\
 \times 2.71828\ 18284 \\
 \times 1.41421\ 35623 \\
 \times 1.61803\ 39887 \\
 \hline
 \end{array}$$


그림 3: 곱셈과 고양이 인식

그러나, 이제 이야기가 달라졌습니다. 드디어 컴퓨터도 '기계학습'이라는 새로운 방법을 통해 사람과 고양이를 알아보기 **시작했습니다**. 다른 말로 컴퓨터가 발명된지 수십년 만에 컴퓨터도 우리 인간처럼 조금씩 볼 수 있게 되었습니다. 우리가 많은 자료를 사용하여 컴퓨터를 학습시킨다면 말입니다. 예를 들면, 일천만 장의 인터넷이미지(200x200) 자료들 가운데 고양이를 인식하는데 1,000대의 컴퓨터(1,600 코어)를 사용하였다고 2012년 6월에 앤드류 응(스탠포드 대) 교수가 발표하였습니다. 여기서 주목할 것은 사람과 새와 고양이를 알아볼 수 있는 프로그램을 통해서가 아니라 컴퓨터로 하여금 많은 자료를 통해 **학습하게 하여** 물체를 알아 볼 수 있게 한다는 것입니다. 컴퓨터가 스스로 배울 수 있도록 만들었던 말입니다.

- [How many computers to identify a cat?](#)
- [Building high-level features using large scale unsupervised learning](#)

전통적 프로그램과 기계학습의 차이

- 정해진 규칙 vs 스스로 학습:

전통적으로 프로그램은 **정해진 규칙에 따라(rule-based)** 동작하여 프로그래머가 상황에 맞게 모든 상황에 대해 프로그램을 작성하여 컴퓨터가 무엇을 해야할지 명확히 정해주어야 합니다. 예를 들어, 스팸메일을 구별하는 프로그램을 만든다고 한다면, 스팸 메일에서 주로 사용되는 단어 하나 하나를 프로그래머가 어떻게 처리할지 규칙을 정하고, 코딩을 해야합니다. 그러면, 메일이 들어올 때, 프로그램은 이미 프로그래머에 의해서 주어진 스팸 분류 단어들 중에서 메일 안에 있는 단어와 매칭되는 것이 있을 경우 스팸 메일로 처리하게 됩니다. 즉, 프로그래머가 모든 상황을 예측하여 프로그램을 작성해야 합니다. **프로그래머가 스팸메일을 학습하고 이를 프로그램에게 알려주어야 하는 것이지요.**

- 새로운 규칙 vs 스스로 학습

전통적인 프로그램의 성능을 향상하기 위해서는 프로그래머가 새로운 규칙을 찾아내서 프로그램의 성능이 향상되도록 일일이 새로운 규칙을 추가해야 합니다. 여러분이 페이스북을 쓰면 쓸수록 점점 여러분의 취향에 맞는 글들을 상단에 보여주는 페이스 북의 기능은 기계학습 덕분입니다. 그러나, 기계학습은 스팸메일함에 누적되어 있는 스팸메일들로 컴퓨터가 학습하여 스팸메일을 구별하도록 하는 것입니다.

- 자료 축적 불필요 vs 자료 축적 필요

그러므로, 또한 스팸메일이 많이 쌓이면 쌓일수록 혹은 사용자가 스팸메일인지 아닌지 구별할 때마다 이런 일들을 컴퓨터가 스스로 학습하여 점점 더 나은 스팸메일 분류를 할 수 있도록 그 성능이 향상되는 것이 기계학습입니다.

물론 스팸메일은 개발자들이 밤을 새가며 만든 기존의 프로그램으로 어느 정도 해결할 수 있지만, 현실에서 발생하는 다양한 문제들은 그렇게 하기 힘든 경우들이 많았습니다. 예를 들면, 자동차의 자율주행이나 사람의 표정을 인식하는 것은 다양한 경우의 수가 있기 때문에 이를 모두 기존의 방식으로 프로그램을 작성하기에는 어려웠던 것입니다.

그러나, 이제는 기계학습이 새와 고양이와 사람을 인식하는 할 수 있는 것은 물론이며, 자동차 자율주행이 가능하도록 하며, 음성을 알아 듣고 동시 통역을 하기도 합니다. 또한 우리가 다 익히 알듯이 컴퓨터가 바둑을 잘 둘 수 있을뿐만 아니라 의사와 같이 질병을 진단하며, 변호사를 대신하여 법률 서비스를 하며, 투자를 대행하는 일까지 가능하도록 만든 것이 기계학습입니다. 기계학습이 모든 면에서 인간을 뛰어넘을 수는 없겠지만, 기계학습은 인공지능이라는 꿈에 상당히 가까이 다가갈 수 있는 방법입니다.

인공지능, 기계학습과 딥러닝의 정의

인공지능

1959년 **아더 사무엘**(Arthur Samuel)은 **인공지능**을 "컴퓨터가 배울 수 있는 능력, 즉 프로그램으로 정의하지 않아도, 컴퓨터가 스스로 학습하여 실행할 수 있는 능력에 대한 연구분야("Field of Study that gives computers the ability to learn without being explicitly programmed")라고 정의했습니다. 다시말해 **인공지능**은 사람처럼 학습하고 추론할 수 있는 지능을 가진 컴퓨터 시스템을 만드는 기술입니다. 그러나, 인공지능을 정의하고 60~70년이 되었지만 인공지능을 실생활에서 체험하기는 아직 어렵습니다. 스마트폰의 음성 비서와 간단한 대화를 이어가기도 아직 어렵습니다. 그러나 최근 **ChatGPT**의 진전된 모습을 보면 앞으로 기계학습을 통해 무엇을 더 잘 할 수 있을지는 우리의 상상력에 달려 있을 것입니다.

강 인공지능(Strong AI) & 약 인공지능(Weak AI)

인공지능은 **강 인공지능**과 **약 인공지능**으로 구별할 수 있습니다. 강 인공지능은 사람과 구분이 안 될 정도로 강한 성능을 가진 인공지능을 의미합니다. 여기서 '사람과 구분이 안 될 정도로 강한 성능'이란 말은 영화에 등장하는 그야말로 사람처럼 생각하고 말하고 감정도 나타낼 수 있는 정도를 뜻합니다. 지금 우리가 사용하는 음성인식 스피커나 스마트폰과는 간단한 대화도 자연스럽게 이어지지 못하는 것을 보면 알 수 있듯이, 지금까지 개발되고 있는 인공지능은 모두 **약 인공지능**에 해당합니다. 그러나, 최근에 주목을 받고 있는 **ChatGPT**를 보면서, **강 인공지능**의 시대가 도래할 날도 그리 멀지 않았다고 말하는 사람들도 있습니다.

기계학습(머신러닝)

우리가 다룰 **기계학습** 또는 **머신러닝**(Machine Learning)은 규칙을 일일이 프로그래밍하지 않아도 데이터에서 규칙을 학습하는 알고리즘을 연구하는 분야입니다. 인공지능을 연구하는 분야 중에서 지능을 실제로 구현하기 위한 소프트웨어를 담당하는 핵심에 해당합니다. 1998년 **얀 르쿤**(Yann LeCun)이 신경망 모델을 만들어 손글씨 숫자를 인식하는데 성공했습니다. 이 신경망의 이름을 LeNet-5라고 하며 최초의 합성곱 신경망(CNN)입니다. 기계학습은 데이터로부터 규칙을 찾아내야 위하여 많은 데이터를 다루어야 하므로 통계 혹은 수학 이론이 필요할 때도 있지만, 많은 경험을 바탕으로 발전하는 경우도 상당합니다.

심화학습(딥러닝)

인공지능과 더불어 기계학습과 딥러닝이란 말을 많이 듣는데, 이 용어도 우리가 정리해둘 필요가 있습니다. 많은 기계학습 알고리즘 중에 인공신경망을 기반으로 한 방법들을 통칭하여 **딥러닝**이라고 부릅니다. 그러나 때로 사람들은 인공신경망과 딥러닝을 크게 구분하지 않고 사용하기도 합니다. 2016년 이세돌과 알파고의 바둑 대국으로 인하여 국내에서도 딥러닝에 대한 관심이 높아지는 계기가 있었습니다. 그러나, 그에 앞서 2012년에 **제프리 힌턴**(Geoffrey Hinton) 팀이 이미지 분류 대회인 ImageNet에서 기존의 기계학습 방법을 누르고 탁월한 성능으로 보여줌으로 우승을

함으로 많이 알려지기 시작했습니다. 이 때 힌튼 교수의 팀이 사용한 모델의 이름은 AlexNet이며 합성곱 신경망(딥러닝의 일종)을 사용했습니다. 그 후, 이미지 분류작업에는 합성곱 신경망이 널리 사용되기 시작했습니다.

정리하면, 기계학습은 인공 지능(Artificial Intelligence)의 한 분야이며, 데이터를 이용하여 학습시키는 방법론입니다. 심화학습(혹은 심층학습, Deep Learning)은 다층적인 인공신경망(ANN) 알고리즘을 이용한 기계학습의 한 분야입니다. 다음과 같이 도식화할 수 있습니다.

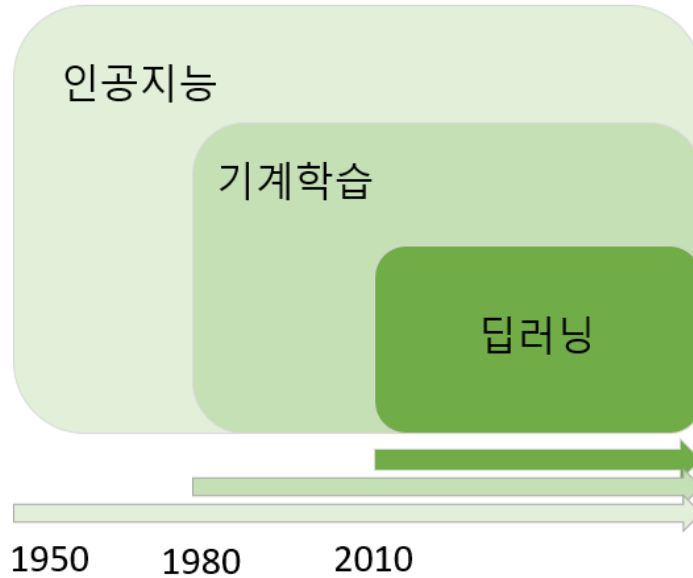


그림 4: 딥러닝, 기계학습, 인공지능 다이어그램

기계학습 종류

기계학습은 자료를 이용하여 컴퓨터를 학습시키는 방법론입니다. 이때 기계학습의 종류는 기계를 학습하는 방법에 따라 다음과 같이 크게 세가지 종류로 나눌 수 있습니다.

1. 지도학습(Supervised learning)

- 주어진 학습자료마다 레이블(label, 명시적인 답)이 주어진 상태에서 학습하여 새로운 문제의 답을 **예측**할 수 있는 학습 방법.
- 예측한 값과 실제 값을 비교하여 정확하게 예측했는지 여부를 피드백 받을 수 있음
- 예를 들어 지금까지의 날씨를 보고 내일의 날씨를 **예측**하는 것

1. 비지도학습(Unsupervised learning)

- 레이블(명시적인 답)이 없이, 데이터 자체에서 패턴을 찾아내는 학습 방법
- 레이블이 없기 때문에 제대로 된 패턴을 찾았는지 피드백 받을 수 없음
- 예를 들어 신문 기사를 분석하여 정치, 경제, 시사뉴스 클러스터 그룹을 만드는 것

1. 강화학습(Reinforcement learning)

- 주어진 환경(State)에서 어떤 행동(Action)을 할 때 얻는 보상(Reward)을 극대화함으로 학습하는 방법.
- 어떤 행위가 최종 결과에 어떤 영향을 주었는지 불명확한 문제에 자주 사용됨
- 예를 들어 바둑, 게임, 미로 찾기 등에 적용할 수 있음.

참고: 준지도학습(Semi-supervised learning) - 학습 데이터의 일부분이 레이블을 가지고 있는 경우를 준지도학습이라고 부릅니다.

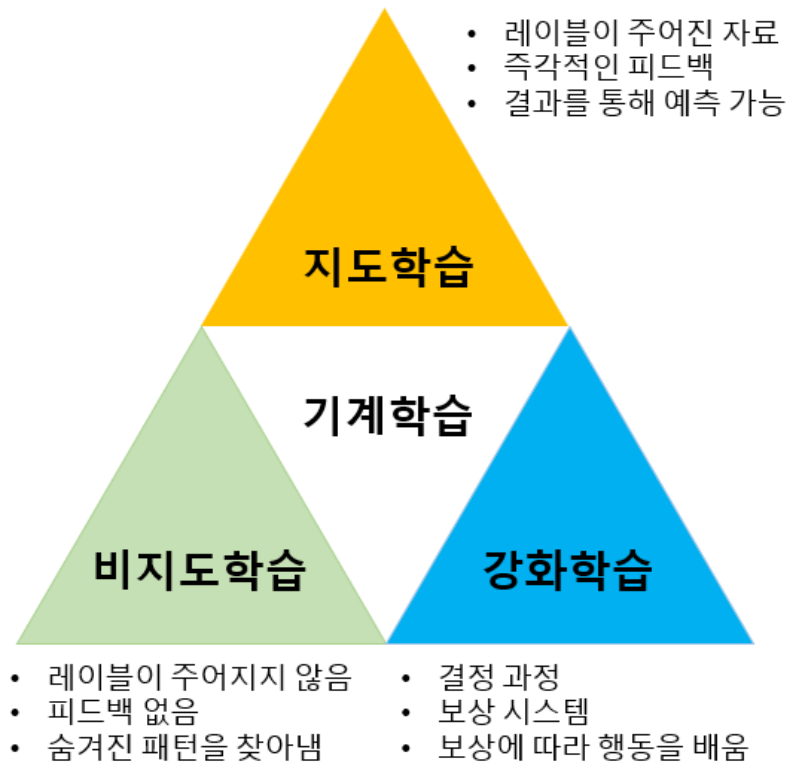


그림 4: 기계학습의 종류

출처: http://www.techjini.com/blog/Machine_learning/

지도학습

지도학습은 시스템에게 문제와 답을 주어 그 패턴을 미리 익히도록 하는 학습방법입니다. 그리고 결국에는 컴퓨터가 경험해보지 못하는 패턴이 들어올 경우에, 적절한 답을 예측해 내도록 하는 것입니다.

지도학습을 크게 두 종류로 나눈다면, **분류**(Classification)와 **회귀**(Regression)가 있습니다.

분류는 데이터가 어느 클래스에 속하느냐 문제입니다. 미리 정의된, 가능성이 있는 여러 클래스 레이블(label, 명시적인 답) 중 하나를 예측하는 것입니다. 예를 들면, 사진 속에 있는 사람들을 남자와 여자로 나눌 수 있고 혹은 '남자인가?'라는 질문에 예와 아니오라고 답을 해야 합니다. 분류에서 "남자도 아니고 여자도 아니고 중간인데요"같은 답은 없습니다. 이런 예를 **이진 분류**(binary classification)라고 합니다. 그러나, 경우에 따라 셋 이상으로 분류하는 **다중 분류**(multiclass classification)도 있습니다.

잘 알려진 또 다른 예로는, 다음과 같이 손으로 쓴 숫자 이미지인 ****MNIST**** 자료(레이블이 있는 자료)가 있으며, 이는 물론 다중 분류 자료로 사용됩니다. 이 자료 샘플의 각 숫자 이미지 하나는 28×28 크기의 이미지이며, 이 자료샘플들은 0부터 9까지의 숫자 레이블이 지정되어 있어 각각 어떠한 숫자를 쓴 것인지 확인할 수 있습니다. 이러한 레이블이 알려진 자료샘플로 학습을 진행한 후, 레이블이 지정이 되지 않은 임의의 자료들이 어떤 숫자인지 ****분류**** 혹은 ****예측****할 수 있는 모델 (****함수**** 혹은 인공지능망의 계수 등)을 유추해 내는 것을 ****지도학습****이라고 합니다.

지도학습은 이미지 분류, 진단, 명의 도용 탐지, 고객 유지(Customer Retention) 등에 활용됩니다.

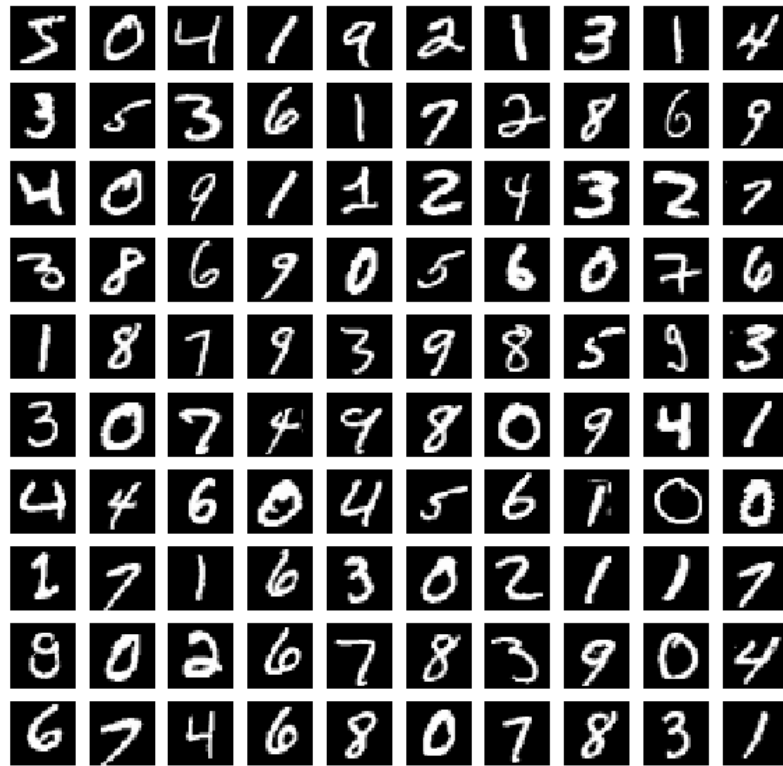


그림 5: MNIST 자료 샘플

회귀는 연속된 값(real number, 실수)을 **예측**하는 것입니다.

예를 들면, 사진 속에 있는 인물에 대하여 몸무게를 예측하는 문제가 회귀입니다. 키를 이용해 몸무게를 예측하거나 집의 면적으로 집값을 예측하는 것도 회귀 문제라고 볼 수 있습니다. 다음 도표가 학생들의 공부 시간에 따른 퀴즈의 점수를 나타낸다고 가정할 때, 2.5시간 공부한 학생의 퀴즈 점수는 얼마가 될 것인가 예측할 수 있는 최적의 **함수**를 찾는 것이 선형 회귀입니다.

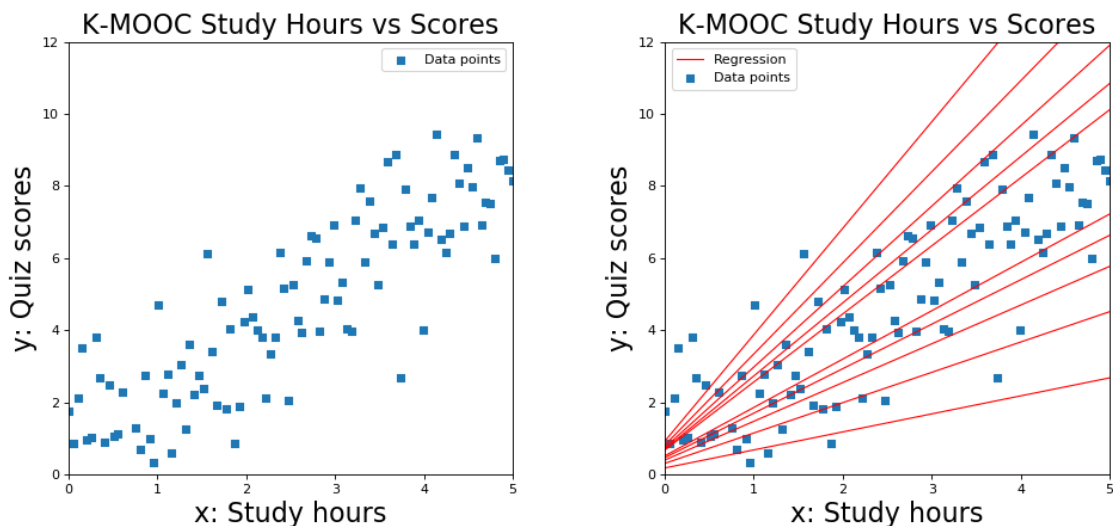


그림 6: 선형회귀(공부시간 vs 퀴즈점수)

여기서 연속적인 숫자 즉 실수를 예측한다는 말은 퀴즈 점수가 5.3이든 5.6이든 큰 문제가 되지 않습니다.

분류와 회귀를 함께 사용해야 하는 경우도 있습니다. 예를 들어 날씨가 좋은지, 나쁜지를 이모티콘으로 나타내고 싶다고 하면, 온도와 구름, 바람 등의 요소로 날씨를 회귀분석하여 0.8 이상이면

웃는 이모티콘, 그 이하이면 울상 이모티콘을 보여줄 수 있습니다. 두 가지의 값(좋음, 나쁨)을 나타내므로 분류 문제이지만, 회귀를 이용하여 분류를 구현하게 됩니다.

회귀가 활용되는 분야는 광고 인기 예측, 날씨 예측, 시장 예측, 기대 수명 예측, 인구 성장 예측에 사용됩니다.

비지도학습

비지도학습은 컴퓨터가 스스로 자료를 모델링하는 학습 방법입니다.

대표적으로 군집화(clustering), 확률밀도추정, 차원축소 등의 다양한 기법을 사용하여 주어진 자료에 숨겨진 구조나 특성을 찾아냅니다. 예측한 결과로부터 피드백도 없습니다. 예를 들면, 비지도학습으로 파티 장소의 소음 속에 사람의 음성을 분리하거나 혹은 아무런 단서없이 주어진 수많은 영상을 여러 장르로 자동적으로 분류하도록 하는 학습 방법입니다.

또 다른 예를 들면, 다음과 같이 레이블이 없이 주어진 자료(검은 색)에 대해 스스로 학습하여 자료들을 여러 개의 군집을 유추해내는 학습방법도 비지도학습이라고 합니다.

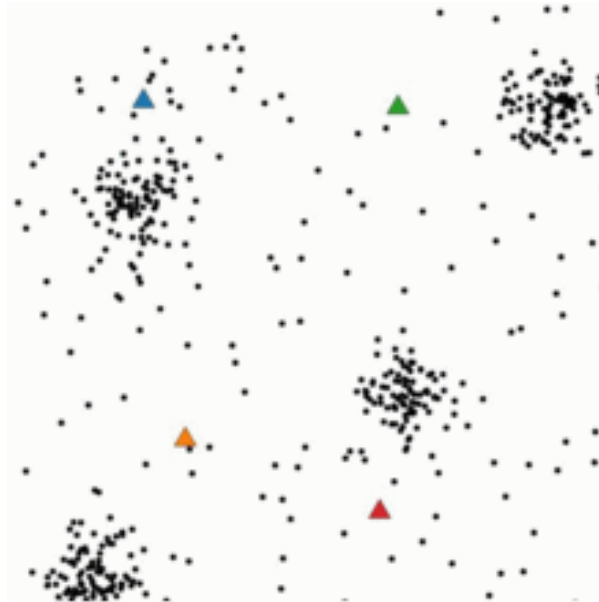


그림 5: 비지도학습-군집화

출처: https://ko.wikipedia.org/wiki/K-평균_알고리즘

이러한 군집화(Clustering)은 추천 시스템, 타겟 마케팅, 고객 세분화에 활용되고 있습니다. 비지도 학습 중에 대표적인 "차원 감소"는 빅데이터 시각화(feature elicitation), 특성 추출, 구조 탐색(structure discovery), 압축(meaningful compression)에 활용 됩니다.

강화학습

주어진 환경(state)에서 어떤 행동(action)을 할 때 얻는 보상(reward)을 극대화함으로 학습하는 방법입니다. 강화학습에서는 지도학습과 달리 학습자료가 제공되지 않으며, 보상이 주어집니다. 행동을 하는 주체(agent)는 환경을 관찰하고, 이를 기반으로 의사결정을 내려 행동합니다. 이때 환경이 변화하면서 행동의 주체가 어떤 보상을 받습니다. 이를 기반으로 행동의 주체는 더 많은 보상을 얻을 수 있는 방향으로 행동을 학습하게 됩니다.

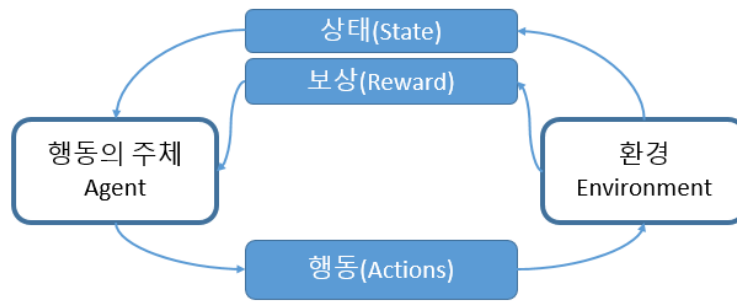


그림 6: 강화학습

예를 들면, 행동의 주체를 "강아지" 그리고, 환경을 "먹이 기계"가 있다고 가정합니다. 강아지는 먹이 기계에서 냄새가 난다는 것을 알게 됩니다. 그러나 어떻게 먹어야 하는지는 모릅니다. 강아지는 기계에 대해 다양한 행동을 시도하다가 어떤 버튼을 눌러 먹이가 조금 나오는 것을 발견하고, 여러 버튼들 중에서 어떤 것을 누르면 물을 마실 수 있게 됩니다. 강아지가 자신의 어떤 행동에 대해 보상을 받게 된 것입니다. 강아지가 자기 행동에 대한 보상을 받으면서 학습하고, 자기가 해야 할 최선을 행동(정책)을 찾아가는 학습 방법이 강화학습입니다.

강화학습의 대표적인 예로 2016년에 이세돌 프로 바둑기사와 대국을 펼친 알파고가 있습니다. 체스는 이미 지난 1997년 인간이 컴퓨터에 정복당한 영역 중 하나이지만, 바둑은 복잡성으로 인해 아직까지 정복되지 않고 있는 분야였습니다. 바둑 경기의 경우의 수는 10의 170 제곱에 이릅니다. 이를 숫자로 풀면 우주에 있는 원자의 수보다 더 큰 숫자이지요. 체스와 비교해서도 경우의 수가 10의 100제곱 이상 많은 것이기도 합니다. 알파고는 강화학습을 통해 좋은 수를 두었을 때 그에 대한 보상을 받으면서 학습하여, 자신이 어떤 수를 두어야 하는지 스스로 판단했습니다.

강화학습의 또 다른 예로, 영국의 딥마인드(Deep Mind)라는 벤처 회사가 벽돌부수기(아타리 2600 비디오)게임을 강화학습으로 학습하는 과정을 2014년에 발표하였습니다. 컴퓨터에 주어진 입력 정보는 점수와 픽셀 디스플레이뿐이었다고 합니다. 게임의 규칙도 모른 채 게임을 시작했습니다. 컴퓨터가 처음에는 무엇을 하는지 잘 몰랐지만, 벽돌을 깰 때마다 점수(reward)를 주는 방식으로 학습 과정을 거치며, 일정한 시간이 지난 후에는 점점 실력이 좋아지고, 6시간 후에 굉장한 속도로 벽돌을 깨는 것을 볼 수 있습니다. 이러한 과정이 논문으로 네이처(2015년 2월, <https://www.nature.com/articles/nature14236>)에 실렸습니다.

그 즈음에 딥마인드는 구글의 한 회사가 됩니다. 그리고 2년 후, 딥마인드는 알파고를 만들어서 이세돌 기사와 대국을 하여 4:1로 이기는 일이 일어난 것입니다. 딥마인드 알파고가 최근에 은퇴를 했다는 소식을 들었습니다.

강화학습은 능력 습득, 로봇 항법, 실시간 결정, 게임에 많이 사용됩니다.

본 강좌에서는 딥러닝의 핵심적인 알고리즘에 해당하는 다층적 인공신경망과 지도학습의 원리를 이해하고 실제로 구현할 것입니다.

연습 문제

1. 다음과 같은 손글씨는 인공신경망을 학습(훈련)시키기 위한 데이터의 일부분입니다. 지도학습 알고리즘을 사용하려면, 각 이미지에 대한 실제 값들(5, 0, 4, 1, 9)도 추가적으로 제공해야 합니다. 이러한 값들을 무엇이라고 합니까?



1. 입력자료
2. 학습자료
3. 레이블
4. 예측값

2. 손글씨로 작성한 편지가 있는 사진을 타이핑한 문서로 변환할 때 적합한 학습 방법은 무엇입니까?

1. 지도 학습
2. 비지도 학습
3. 강화학습

3. 명시적인 답이 없는 데이터를 사용하여 기계학습 모델을 훈련하는 방법이기 때문에 모델의 훈련 결과를 평가하기 어려운 것이 특징이기도 한 학습법은 무엇입니까?

1. 지도 학습
2. 비지도 학습
3. 강화학습

4. 드론이 날면서 내외부적으로 발생하는 다양한 흔들림을 제어하기 위해 적용할만한 학습 방법은 무엇입니까?

1. 지도 학습
2. 비지도 학습
3. 강화학습

5. 기계학습이나 심화학습에서 '학습'은 컴퓨터가 스스로 데이터로부터 `__`을 찾아내는 것입니다. 전통적인 프로그램은 사람이 이것을 찾아서 만든대로 동작했기 때문에, 이러한 새로운 '학습' 방법을 경험하게 되면 참 놀라운 일로 보입니다. 밑줄친 부분에 알맞은 단어는 무엇입니까?__

1. 코드
2. 규칙
3. 패턴
4. 코어
5. 레이블

6. ChatGPT는 미국의 OpenAI가 개발한 인공지능 챗봇입니다. 빅데이터 분석을 바탕으로 사람과 대화하듯 답을 해주는 시스템입니다. 위에서 언급한 다양한 기술과 알고리즘을 사용하는데 다음 중에 어떤 모델에 해당합니까?

1. CNN(Convolution Neural Network, 합성곱신경망)
2. RNN(Recurrent Neural Network, 순환신경망)
3. NLP(Natural Language Processing, 자연어처리)
4. GAN(Generative Adversarial Nets, 적대적 생성망)

학습 정리

1. 기계학습의 배경과 개념
2. 기계학습의 정의와 소개
 - 인공지능, 강 인공지능, 약 인공지능
 - 기계학습(Machine Learning)
 - 심화학습(Deep Learning)
3. 기계학습의 종류
 - 지도학습
 - 비지도학습
 - 강화학습

KMOOC 위대한 수업: 요슈아 벤지오 교수 강의

요슈아 벤지오(Yoshua Benjio) 교수는 제프리 힌튼(Geoffrey Hinton), 얀 르쿤(Yann LeCun)과 함께 2018년 튜링상을 공동 수상했습니다. 그는 딥러닝 개념을 창시했을 뿐만 아니라 그의 제자들(Xavier Glorot, Ian Goodfellow)과 함께 ReLU 함수, GAN(Generative Adversarial Nets, 적대적 생성망)을 창시했습니다. 아래 강의는 www.kmooc.kr 웹사이트에 학습자 등록 이후에 시청 가능합니다. 각 강의는 15분 내외입니다.

1. 딥 러닝의 탄생
2. 추론하는 기계
3. 인공신경망 최적화하기
4. 창작하는 기계
5. 착한 인공지능

참고 자료

- 스탠포드 대학 앤드루 응 교수의 기계학습 강의
 - <https://www.coursera.org/specializations/deep-learning>
 - <https://www.coursera.org/learn/machine-learning/home/welcome>
- 홍콩과학기술대학 김성훈 교수의 심화학습 강의
- 스탠포드 대학 Sebastian Thrun 교수와 Katie Malone의 기계학습 강의
- 스탠포드 대학 비지도학습 & 딥러닝 튜토리얼 웹사이트
- ChatGPT

너는 청년의 때에 너의 창조주를 기억하라 곧 곤고한 날이 이르기 전에, 나는 아무 낙이 없다고 할 해들이 가깝기 전에 (전12:1)