



강사 윤영주
yj.youn1103@gmail.com

첫째마당

인공지능 머신러닝 딥러닝

3장 딥러닝의 핵심 미리 보기

- 1** 미지의 일을 예측하는 원리
- 2** 딥러닝 코드 실행해 보기
- 3** 딥러닝 개괄하기
- 4** 이제부터가 진짜 딥러닝?



1 미지의 일을 예측하는 원리



1 미지의 일을 예측하는 원리

- 미지의 일을 예측하는 원리



기존 환자의
데이터를 이용해
새로운 환자의
수술 결과를 예측하는
프로그램을 짜 보세요!



1 미지의 일을 예측하는 원리

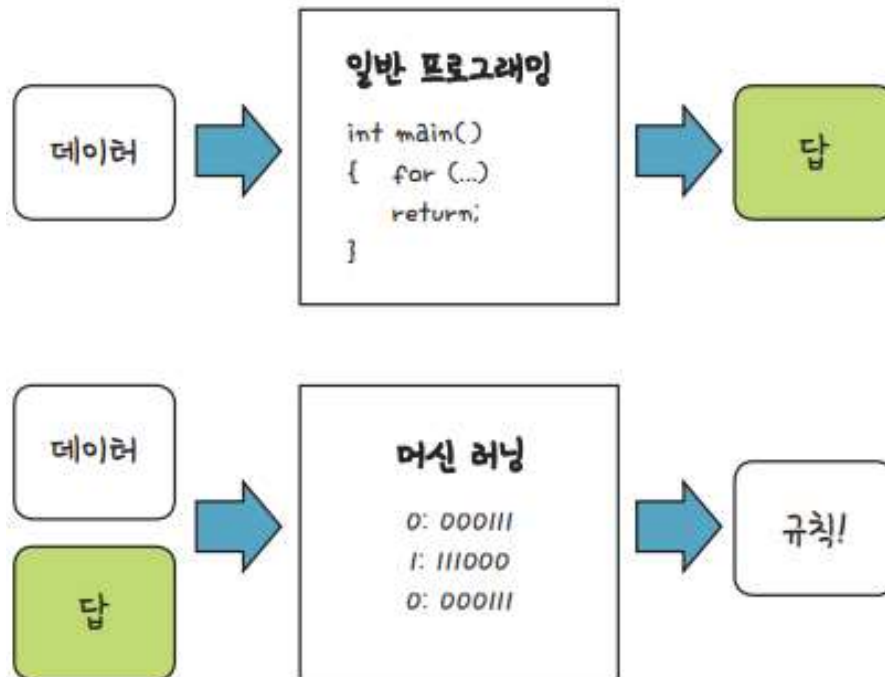
● 미지의 일을 예측하는 원리

- 이러한 과제를 받았다고 해 보자
- 기존 프로그래밍 기법으로 이러한 프로그램을 만들려면 쉽지 않음
- 머신 러닝은 이를 매우 쉽게 해결
- 기존에 우리가 했던 프로그래밍이 데이터를 입력해서 답을 구하는 데 초점이 맞추어 있었음
- 머신 러닝은 데이터 안에서 규칙을 발견하고 그 규칙을 새로운 데이터에 적용해서 새로운 결과를 도출하는 데 초점이 맞추어 있기 때문임
- 머신 러닝은 기존 데이터를 이용해 아직 일어나지 않은 미지의 일을 예측하기 위해 만들어진 기법



1 미지의 일을 예측하는 원리

▼ 그림 2-1 | 머신 러닝과 일반 프로그래밍 비교





1 미지의 일을 예측하는 원리

● 미지의 일을 예측하는 원리

- 실제 예를 들어 머신 러닝을 활용하는 방법에 대해 살펴보자
- 중환자를 전문으로 수술하는 어느 병원의 의사가 수많은 환자를 수술해 오던 중 다음과 같은 질문을 던져 보았음
- “혹시 수술하기 전에 수술 후의 생존율을 수치로 예측할 수 있는 방법이 있을까?” 방법이 있음
- 자신이 그동안 집도한 수술 환자의 수술 전 상태와 수술 후 생존율을 정리해 놓은 **데이터**를 머신 러닝 알고리즘에 넣는 것
- 머신 러닝은 데이터가 가진 패턴과 규칙을 분석해서 저장해 두자
- 이후 새로운 환자가 오면 저장된 분석 결과와 비교해 생존 가능성을 예측하게 되는 것
- 이것이 바로 머신 러닝이 하는 일



1 미지의 일을 예측하는 원리

● 미지의 일을 예측하는 원리

- 여기서 데이터가 입력되고 패턴이 분석되는 과정을 **학습(training)**이라고 함
- 다시 말해 학습 과정은 깨끗한 좌표 평면에 기존 환자들을 하나씩 배치하는 과정이라고 할 수 있음
- 예를 들어 환자들의 분포를 그래프 위에 펼쳐 놓고 이 분포도 위에 수술 성공과 실패 여부를 구분짓는 경계를 그려 넣음
- 이를 잘 저장해 놓았다가 새로운 환자가 오면 분포도를 다시 꺼냄
- 새 환자가 분포도의 어디쯤 위치하는지 정하고는 아까 그려 둔 경계선을 기준으로 이 환자의 수술 결과를 예측하는 것



1 미지의 일을 예측하는 원리

▼ 그림 2-2 | 머신 러닝의 학습 및 예측 과정



기존 환자 데이터 입력



머신 러닝으로 학습



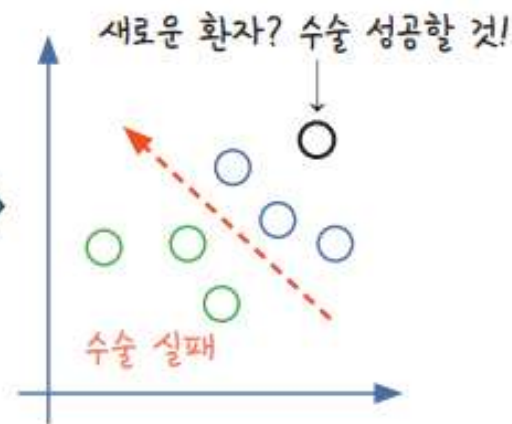
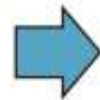
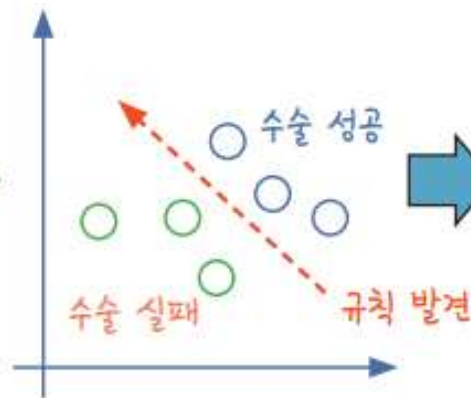
새로운 환자 예측

진료 기록 1: 수술 후 사망

진료 기록 2: 수술 후 생존

진료 기록 3: 수술 후 사망

⋮





1 미지의 일을 예측하는 원리

- 미지의 일을 예측하는 원리

- 우리가 지금 배우려는 것이 바로 이러한 학습과 예측의 구체적인 과정
- 머신 러닝의 예측 성공률은 결국 얼마나 정확한 경계선을 긋느냐에 달려 있음
- 더 정확한 선을 긋기 위한 여러 가지 노력이 계속되어 왔고, 그 결과 퍼셉트론(perceptron), 선형 회귀(linear regression) 등을 지나 오늘날 딥러닝이 탄생



2 딥러닝 코드 실행해 보기

2 딥러닝 코드 실행해 보기



▼ 그림 2-9 | 구글 코랩 실행 결과

1 실행 시간

▼ 나의 첫 딥러닝: '10장 폐암 수술 환자의 생존을 예측' 코드 미리보기

▼ 1. 환경 준비

```
[1] from tensorflow.keras.models import Sequential # 텐서플로의 케라스 API에서 필요한 함수들을 불러옵니다.
    from tensorflow.keras.layers import Dense # 데이터를 다루는데 필요한 라이브러리를 불러옵니다.
    import numpy as np
```

▼ 2. 데이터 준비

```
[2] !git clone https://github.com/taehojo/data.git # 깃허브에 준비된 데이터를 가져옵니다.
```

```
Data_set = np.loadtxt("./data/ThoracicSurgery3.csv", delimiter=",") # 수술 환자 데이터를 불러옵니다.
X = Data_set[:,0:16] # 환자의 진찰 기록을 X로 지정합니다.
y = Data_set[:,16] # 수술 후 사망/생존 여부를 y로 지정합니다.
```

fatal: destination path 'data' already exists and is not an empty directory.

▼ 3. 구조 결정

```
[3] model = Sequential() # 딥러닝 모델의 구조를 결정합니다.
    model.add(Dense(30, input_dim=16, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
```

▼ 4. 모델 실행

```
[4] model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']) # 딥러닝 모델을 실행합니다.
    history=model.fit(X, y, epochs=5, batch_size=16)
```

```
Epoch 1/5
30/30 [=====] - 1s 2ms/step - loss: 1.5790 - accuracy: 0.4957
Epoch 2/5
30/30 [=====] - 0s 2ms/step - loss: 0.4954 - accuracy: 0.8511
Epoch 3/5
30/30 [=====] - 0s 2ms/step - loss: 0.4470 - accuracy: 0.8511
Epoch 4/5
30/30 [=====] - 0s 2ms/step - loss: 0.4369 - accuracy: 0.8511
Epoch 5/5
30/30 [=====] - 0s 2ms/step - loss: 0.4368 - accuracy: 0.8511
```

2 실행 결과

2 딥러닝 코드 실행해 보기



- 딥러닝 코드 실행해 보기

- 실행 결과는 매번 실행할 때마다 미세하게 달라짐
- 이것은 첫 가중치를 랜덤하게 정하고 실행을 반복하며, 조금씩 가중치를 수정해 가는 딥러닝의 특성 때문임
- 딥러닝의 동작 원리에 대해서 앞으로 차차 배워 나갈 것



3 딥러닝 개괄하기



3 딥러닝 개괄하기

● 딥러닝 개괄하기

- 지금 불러와 실행한 코드는 10장에서 상세히 다루게 될 폐암 수술 환자의 수술 1년 후 생존율을 예측한 모델
- 먼저 코드를 개괄적으로 살펴보며 딥러닝을 프로그래밍하는 과정에 대한 감을 잡아 보자

● 1. 환경 준비

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
```

딥러닝을 구동하거나 데이터를 다루는 데 필요한 라이브러리들을 불러옵니다.

2. 데이터 준비

```
!git clone https://github.com/taehojo/data.git

Data_set = np.loadtxt("./data/ThoraricSurgery3.csv",
delimiter=",")
X = Data_set[:,0:16]
y = Data_set[:,16]
```

준비된 수술 환자 정보 데이터를 나의 구글 코랩 계정에 저장합니다. 해당 파일을 불러와 환자 상태의 기록에 해당하는 부분을 X로, 수술 1년 후 사망/생존 여부를 y로 지정합니다.

있음



3 딥러닝 개괄하기

- 딥러닝 개괄하기

3. 구조 결정

```
model = Sequential()  
model.add(Dense(30, input_dim=16, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

딥러닝 모델의 구조를 결정합니다.
여기에 설정된 대로 딥러닝을 수행
합니다.

4. 모델 실행

```
model.compile(loss='binary_crossentropy',  
optimizer='adam', metrics=['accuracy'])  
history = model.fit(X, y, epochs=5, batch_size=16)
```

딥러닝 모델을 실행합니다. 앞서 설
정된 구조대로 실행하고 결과를 출
력합니다.

3 딥러닝 개괄하기



● 딥러닝 개괄하기

1. 환경 준비 딥러닝을 구동하는 데 필요한 라이브러리 호출

- 이 책의 모든 코드는 파이썬으로 되어 있음
- 파이썬은 초보자부터 전문가까지 모두에게 애용되는 프로그래밍 언어로, 특히 다양한 플랫폼에서 데이터를 분석하고 딥러닝, 머신 러닝을 구현하는 데 사용
- 파이썬은 풍부한 라이브러리를 가지고 있다는 것이 장점인데, 라이브러리란 특정한 기능을 담은 작은 프로그램들(module, API)을 모아 놓은 것을 의미
- 목적에 따라 라이브러리를 불러오면 다양한 작업을 간단히 진행할 수 있음
- 라이브러리를 불러올 때 사용하는 명령어가 import

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 코드의 처음이 다음과 같이 시작

```
from tensorflow.keras.models import Sequential ..... ①  
from tensorflow.keras.layers import Dense ..... ②  
import numpy as np ..... ③
```

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 라이브러리에 포함된 모듈이 너무 많을 때, 그중 지금 필요한 일부 모듈만 다음과 같이 불러올 수 있음

```
from (라이브러리명) import (함수명)
```

3 딥러닝 개괄하기



● 딥러닝 개괄하기

- 예를 들어 ❶ **from** tensorflow.keras.models **import** Sequential은 텐서플로(tensorflow)의 케라스(keras)라는 API에 있는 모델(model) 클래스로부터 Sequential() 함수를 불러오라는 의미
- 마찬가지로 ❷ **from** tensorflow.keras.layers **import** Dense는 케라스 API의 레이어(layers) 클래스에서 Dense()라는 함수를 불러오라는 의미
- 불러온 라이브러리명이 길거나 같은 이름이 이미 있을 경우 다음과 같이 짧게 줄일

```
import (라이브러리명) as (새로운 이름)
```

- 예를 들어 ❸ **import** numpy **as** np 명령은 아나콘다에 이미 포함되어 있는 넘파이(numpy) 라이브러리를 np라는 짧은 이름으로 불러와 사용할 수 있게 해 줌



3 딥러닝 개괄하기

● 딥러닝 개괄하기

2. 데이터 준비 데이터를 불러와 사용할 수 있도록 준비

- 이제 데이터를 불러와 구글 코랩에서 사용할 수 있도록 준비할 차례
- 데이터는 직접 업로드하는 방법과 깃허브에서 불러오는 방법이 있음
- 우리는 이 책을 위해 깃허브에 준비된 데이터를 내 계정으로 불러오도록 하겠음
- 데이터를 가져오기 위해 실행하는 코드는 다음과 같음

```
!git clone https://github.com/taehojo/data.git
```

- data라는 폴더가 새로 생기는 것을 확인할 수 있음
- ❶ 폴더 모양의 아이콘을 클릭한 후 ❷ data 폴더를 클릭하면 ❸ 준비된 데이터를 확인할 수 있음

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- data 폴더 안에 있는 데이터들은 **./data/데이터명** 형식으로 불러올 수 있음
- 넘파이 라이브러리를 이용해 data 폴더에 있는 csv 파일을 불러오는 부분은 다음과 같음

```
data_set = np.loadtxt("./data/ThoraricSurgery3.csv", delimiter=",")
```

3 딥러닝 개괄하기



● 딥러닝 개괄하기

- 넘파이 라이브러리의 loadtxt() 함수를 사용해 'ThoraricSurgery3.csv'라는 외부 데이터셋을 불러왔음
- 머신 러닝에서 알고리즘이나 좋은 컴퓨터 환경만큼 중요한 것이 바로 좋은 데이터를 준비하는 일
- 데이터를 면밀히 관찰하고 효율적으로 다루는 연습을 하는 것이 중요
- 우선은 지금 불러온 ThoraricSurgery3.csv 파일에 관해 좀 더 살펴보자
- ① 먼저 그림 2-12와 같이 data 폴더의 ThoraricSurgery3.csv 파일을 더블클릭
- ② 웹 브라우저 우측에 새로운 공간이 생기며 해당 데이터를 미리 볼 수 있음



3 딥러닝 개괄하기

▼ 그림 2-13 | 폐암 수술 환자의 의료 기록과 1년 후 사망 여부 데이터

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	2.88	2.16	1	0	0	0	1	1	3	0	0	0	1	0	60	0
2	2	3.4	1.88	0	0	0	0	0	0	1	0	0	0	1	0	51	0
3	2	2.76	2.08	1	0	0	0	1	0	0	0	0	0	1	0	59	0
4	2	3.68	3.04	0	0	0	0	0	0	0	0	0	0	0	0	54	0
5	2	2.44	0.96	2	0	1	0	1	1	0	0	0	0	1	0	73	1
...
470	2	4.72	3.56	0	0	0	0	0	0	1	0	0	0	1	0	51	0

↑
샘플 수
(환자 수: 470명)

↑
속성
(의료 기록: 16가지)

↑
클래스
(사망: 0/생존: 1)

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 가로줄 한 행이 한 사람의 환자로부터 기록된 정보를 의미
- 총 470행이므로 환자 470명에 대한 정보
- 한 행에는 17개의 숫자가 들어 있음
- 이는 환자마다 17개의 정보를 순서에 맞추어 정리했다는 의미
- 앞의 정보 16개는 종양의 유형, 폐활량, 호흡 곤란 여부, 고통 정도, 기침, 흡연, 천식 여부 등 16가지 환자 상태를 조사해서 기록해 놓은 것
- 마지막 17번째 정보는 수술 1년 후의 생존 결과
- 1은 수술 후 생존했음을, 0은 수술 후 사망했음을 의미

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 이번 프로젝트의 목적은 1번째 항목부터 16번째 항목까지 이용해서 17번째 항목, 즉 수술 1년 후의 생존 또는 사망을 맞히는 것
- 1번째 항목부터 16번째 항목까지 **속성(attribute)**이라 하고, 정답에 해당하는 17번째 항목을 **클래스(class)**라고 함
- 클래스는 앞서 이야기한 '이름표'에 해당
- 딥러닝을 위해서는 속성과 클래스를 서로 다른 데이터셋으로 지정해 주어야 함

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 먼저 속성으로 이루어진 데이터셋을 X라는 이름으로 만들어 줌

```
X = Data_set[:,0:16]
```

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 파이썬은 숫자를 1부터 세지 않고 0부터 셈
- 범위를 정할 경우 콜론(:) 앞의 숫자는 범위의 맨 처음을 의미하고, 콜론(:) 뒤의 숫자는 이 숫자가 가리키는 위치 '바로 앞'이 범위의 마지막이라는 의미
- 쉽표(.)를 기준으로 앞은 행(샘플), 뒤는 열(속성)의 범위가 입력
- 예를 들어[:,0:16]은 모든 행의 1번째 열부터 16번째 열까지 가져오라는 의미

3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 다음으로 17번째 줄에 위치한 클래스를 따로 모아 데이터셋 y 로 지정

```
y = Data_set[:,16]
```

- 보통 집합은 대문자로, 원소는 소문자로 표시
- X 에는 여러 개의 속성이 담기기 때문에 대문자 X 로, y 는 클래스 하나의 원소만 담기기 때문에 소문자 y 로 썼음



3 딥러닝 개괄하기

● 딥러닝 개괄하기

3. 구조 결정 어떤 딥러닝 구조를 만들 것인가

- 앞서 우리는 딥러닝을 실행시키기 위해 텐서플로를 불러왔음
- 텐서플로는 구글에서 만든 딥러닝 전용 라이브러리
- 텐서플로를 이용하면 여러 가지 알고리즘을 활용해 다양한 딥러닝 작업을 할 수 있지만, 사용법이 쉽지 않다는 단점이 있음

▼ 그림 2-14 | 텐서플로(<https://www.tensorflow.org>)



3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 이를 해결해 주기 위해 개발된 것이 케라스(Keras)

▼ 그림 2-15 | 케라스(<https://keras.io>)



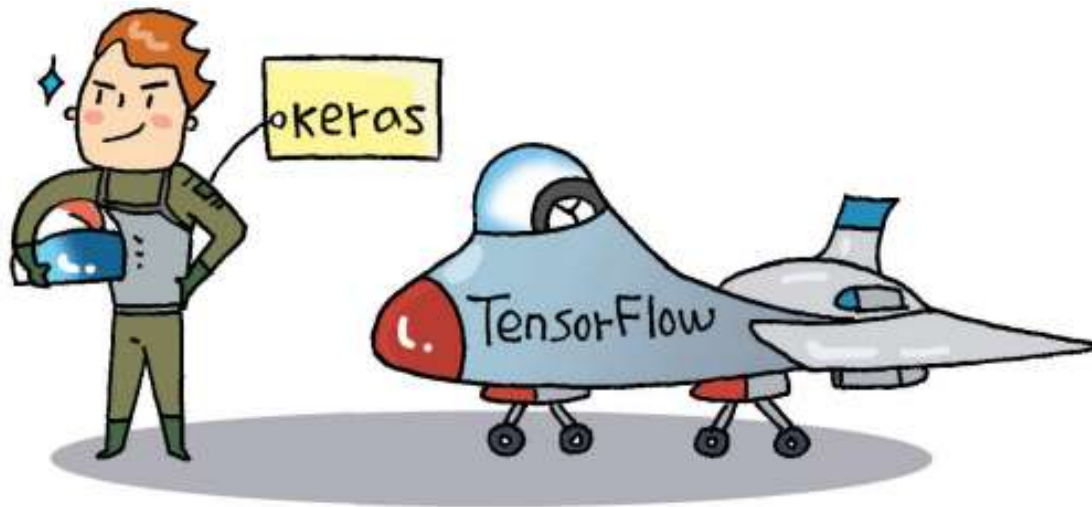


3 딥러닝 개괄하기

- 딥러닝 개괄하기

- 텐서플로가 목적지까지 이동시켜 주는 비행기라면 케라스는 조종사에 해당
- 케라스를 활용하면 딥러닝의 거의 모든 작업을 쉽게 처리할 수 있음

▼ 그림 2-16 | 텐서플로와 케라스의 관계



3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 불러온 예제에서 케라스를 어떻게 활용했는지 알아보자

```
model = Sequential() ..... ❶  
model.add(Dense(30, input_dim=16, activation='relu')) ..... ❷  
model.add(Dense(1, activation='sigmoid')) ..... ❸
```

3 딥러닝 개괄하기



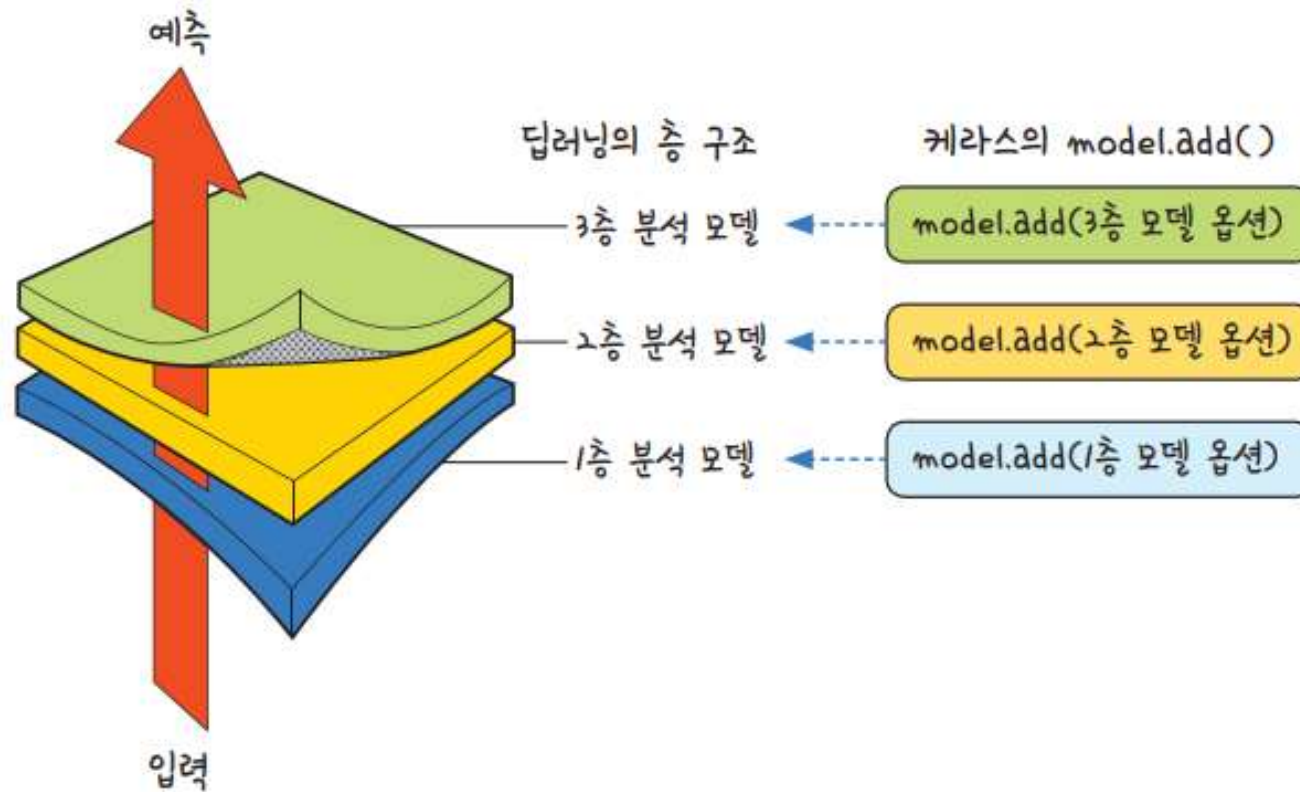
● 딥러닝 개괄하기

- ❶ 먼저 앞서 불러온 Sequential() 함수를 model로 선언
- 앞으로 상세히 다루겠지만, 딥러닝은 그림 2-17과 같이 여러 층이 쌓여 있는 구조
- 준비된 데이터가 입력되는 입력층에 이어 첫 번째 작업을 진행하는 1층, 두 번째 작업을 하는 2층... 이런 식으로 출력 결과가 나오는 출력층까지 여러 개의 층이 각자 자신이 맡은 일을 하면서 앞뒤로 정보를 주고받음
- 케라스의 Sequential() 함수는 딥러닝의 한 층 한 층을 ❷ model.add()라는 함수를 사용해 간단히 추가시켜 줌
- 여기서는 ❷ 와 ❸, 두 개의 층을 쌓았음
- Model.add() 함수를 한 줄 추가하는 것으로 필요한 만큼 내부의 층을 만들 수 있음

3 딥러닝 개괄하기



▼ 그림 2-17 | 딥러닝의 층 구조와 케라스



3 딥러닝 개괄하기



- 딥러닝 개괄하기

- 각 `model.add()` 함수 안에는 케라스 API의 `layers` 클래스에서 불러온 `Dense()` 함수가 포함되어 있음
- `Dense`는 '밀집한, 뻑뻑한'이란 뜻으로, 여기서는 각 층의 입력과 출력을 촘촘하게 모두 연결하라는 것

3 딥러닝 개괄하기



● 딥러닝 개괄하기

- 이제 두 가지를 더 알면 됨
- 첫째, 좋은 딥러닝 모델을 만들려면 몇 개의 층으로 쌓아 올려야 하는가?
- 둘째, Dense 함수 안에 있는 숫자와 설정의 의미는 무엇이며, 어떻게 정해야 하는가?
- 딥러닝을 설계한다는 것은 결국 몇 개의 층을 어떻게 쌓을지, Dense 외에 어떤 층을 사용할지, 내부의 변수들을 어떻게 정해야 하는지 등에 대해 고민하는 것
- 대개 어떤 데이터를 가지고 무엇을 할 것인지에 따라 딥러닝의 설계가 결정
- 각 설정과 변수의 의미를 알고 이것을 자유롭게 구성할 수 있는지가 딥러닝을 잘 다루는지 여부를 결정하는 것
- 이 책에서 배울 내용도 결국 이것
- Dense() 함수의 내부에 쓰인 각 설정의 의미들은 책의 진도가 나감에 따라 앞으로 하나씩 배우게 될 것



3 딥러닝 개괄하기

- 딥러닝 개괄하기

4. 모델 실행 만든 딥러닝을 실행시키고 결과 확인

- 만들어 놓은 모델을 실행시키는 부분

```
model.compile(loss=binary_crossentropy, optimizer='adam',  
metrics=['accuracy']) ..... ❶  
history = model.fit(X, y, epochs=5, batch_size=16) ..... ❷
```

3 딥러닝 개괄하기



● 딥러닝 개괄하기

- `model.compile()` 함수는 앞서 만든 `model`의 설정을 그대로 실행하라는 의미
- 함수 내부에 `loss`, `optimizer`, `metrics` 등 키워드들이 들어 있음
- 이것은 앞 단계에서 만들어진 딥러닝 구조를 어떤 방식으로 구동시키고 어떻게 마무리할 것인지와 관련된 옵션들인데, 둘째 마당과 셋째 마당에서 자세히 배울 것
- 딥러닝은 여러 층이 쌓여 만들어진다는 설명을 이미 한 바 있음
- 딥러닝의 기본 방식은 이 층들을 한 번만 통과하는 것이 아니라 위아래로 여러 차례 오가며 최적의 모델을 찾는 것
- 몇 번을 오갈 것인지, 그리고 한 번 오갈 때 몇 개의 데이터를 사용할 것인지정하는 함수가 `model.fit()` 함수



4 이제부터가 진짜 딥러닝?

4 이제부터가 진짜 딥러닝?



● 이제부터가 진짜 딥러닝?

- 지금까지 딥러닝을 위한 작업 환경을 만들고, 딥러닝 모델을 실행해 보면서 학습 목표를 파악했음
- 딥러닝을 위한 학습에는 단순한 파이썬 프로그래밍뿐 아니라 선형 회귀, 로지스틱 회귀 등 기초 통계학 개념들도 필요함
- 이러한 설명에는 필연적으로 수학 개념이 따라오게 되어 있음
- 예전에 배웠지만 잠시 잊고 지냈던 분들을 위해 '3장. 딥러닝을 위한 기초 수학'을 다음 장에 준비했음
- 물론, 수학에 자신이 있다면 둘째 마당으로 직행해도 됨
- 만일 예전에 배웠던 것들을 한 번 더 확인하고 싶다면 다음 장에 이어지는 딥러닝을 위한 기초 수학 편을 통해 필요한 개념들을 정리하고 넘어가길 권함

3장 딥러닝을 위한 기초 수학



- 1 일차 함수, 기울기와 y 절편
- 2 이차 함수와 최솟값
- 3 미분, 순간 변화율과 기울기
- 4 편미분
- 5 지수와 지수 함수
- 6 시그모이드 함수
- 7 로그와 로그 함수

딥러닝을 위한 기초 수학



- 딥러닝을 위한 기초 수학

- '딥러닝을 배운다'는 말에는 딥러닝의 실행법을 익히는 것뿐 아니라, 딥러닝의 수학 원리를 공부한다는 의미도 담겨 있음
- 원리를 알아야 정확히 실행할 수 있기 때문에 딥러닝의 원리를 이해하는 것은 좋은 코드를 만드는 것 이상으로 중요함
- 딥러닝의 수학 원리를 이해하기 위해서는 당연히 기본적인 수학 지식이 필요함
- 어떤 원리로 입력 값의 패턴을 분석하고 학습하는지 이해하려면 그 배경이 되는 수학 연산을 살펴보아야 하고, 여기에 사용되는 함수들을 알아야 하기 때문임

딥러닝을 위한 기초 수학



- 딥러닝을 위한 기초 수학

- 좋은 소식은 딥러닝 뒤에 있는 수학적 배경이 다른 머신 러닝과 비교했을 때 그다지 어렵지 않다는 것
- 딥러닝은 고등학교 수준의 수학만으로도 원리와 배경을 파악할 수 있음
- 조금 더 깊이 공부하더라도 대학교 교양 강좌 수준을 넘지 않는 범위에서 딥러닝의 원리를 이해할 수 있음
- 이 장에서는 딥러닝을 이해하는 데 꼭 필요한 기초 수학을 먼저 공부하겠음
- 각 수학 공식이 딥러닝의 어느 부분에 활용되는지 참고하면서, 수학에 대한 두려움을 없애고 딥러닝 공부를 시작할 수 있길 바람



1 일차 함수, 기울기와 y 절편



1 일차 함수, 기울기와 y 절편

- 일차 함수, 기울기와 y 절편

- 함수란 두 집합 사이의 관계를 설명하는 수학 개념
- 변수 x 와 y 가 있을 때, x 가 변하면 이에 따라 y 는 어떤 규칙으로 변하는지 나타냄
- 보통 함수를 나타낼 때는 function의 f 와 변수 x 를 사용해 $y = f(x)$ 라고 표시



1 일차 함수, 기울기와 y 절편

- 일차 함수, 기울기와 y 절편

- 일차 함수는 y 가 x 에 관한 일차식으로 표현된 경우를 의미
- 예를 들어 다음과 같은 함수식으로 나타낼 수 있음

$$y = ax + b \quad (a \neq 0)$$

- x 가 일차인 형태이며 x 가 일차로 남으려면 a 는 0이 아니어야 함



1 일차 함수, 기울기와 y 절편

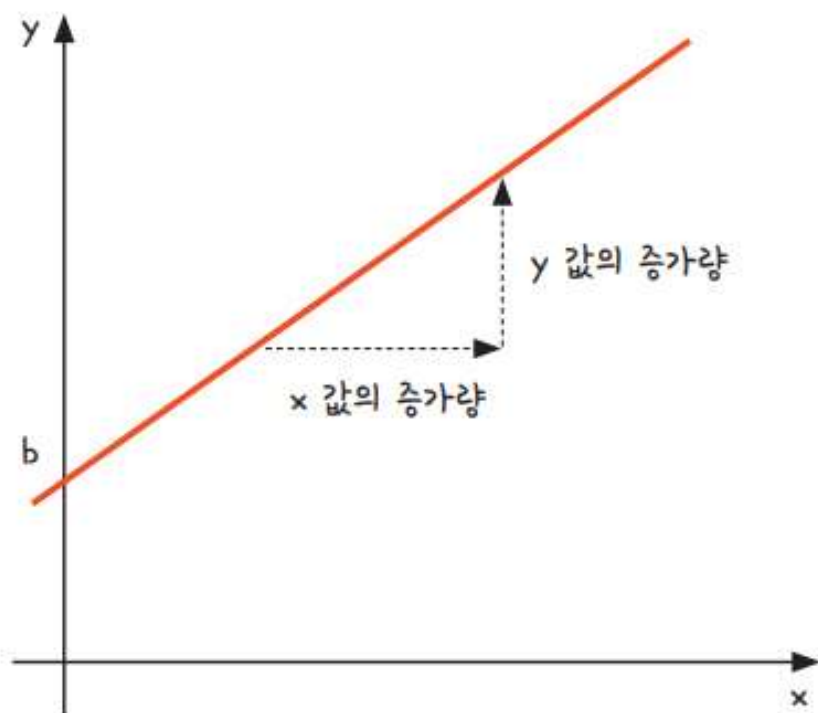
- 일차 함수, 기울기와 y 절편

- 일차 함수식 $y = ax + b$ 에서 a 는 기울기, b 는 절편이라고 함
- 기울기는 기울어진 정도를 의미하는데, 그림 3-1에서 x 값이 증가할 때 y 값이 어느 정도 증가하는지에 따라 그래프의 기울기 a 가 정해짐
- 절편은 그래프가 축과 만나는 지점을 의미
- 그림 3-1에서 y 축과 만나는 y 절편이 바로 b

1 일차 함수, 기울기와 y 절편



▼ 그림 3-1 | 일차 함수 그래프





1 일차 함수, 기울기와 y 절편

- 일차 함수, 기울기와 y 절편
 - 딥러닝의 수학 원리를 배울 때 초반부터 이 식이 등장
 - x 가 주어지고 원하는 y 값이 있을 때 적절한 a 와 b 를 찾는 것, 이것이 바로 딥러닝을 설명하는 가장 간단한 표현



2 이차 함수와 최솟값

2 이차 함수와 최솟값



- 이차 함수와 최솟값

- 이차 함수란 y 가 x 에 관한 이차식으로 표현되는 경우를 의미
- 다음과 같은 함수식으로 표현할 수 있음

$$y = ax^2 (a \neq 0)$$

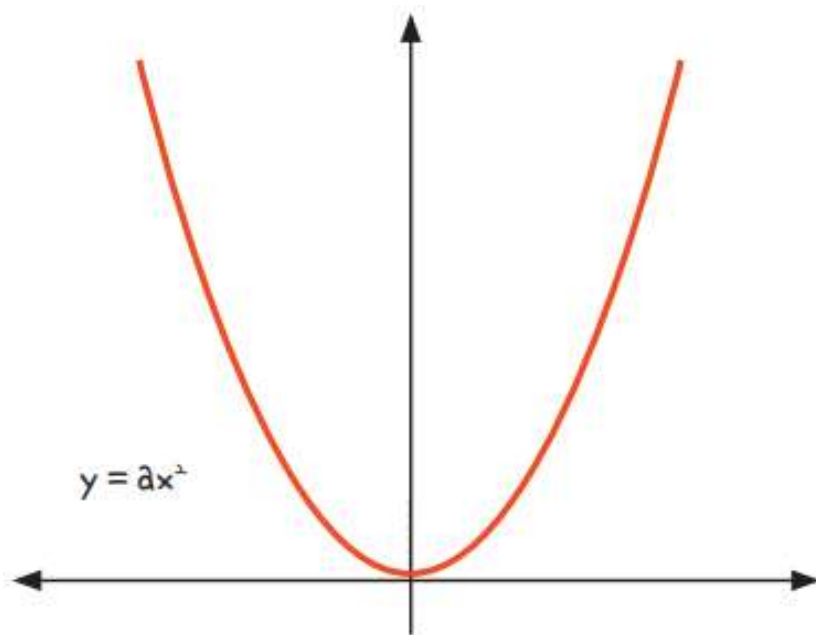
2 이차 함수와 최솟값



- 이차 함수와 최솟값

- 이차 함수의 그래프는 그림 3-2와 같이 포물선 모양
- $a > 0$ 이면 아래로 볼록한 그래프가 됨

▼ 그림 3-2 | 이차 함수 그래프



2 이차 함수와 최솟값



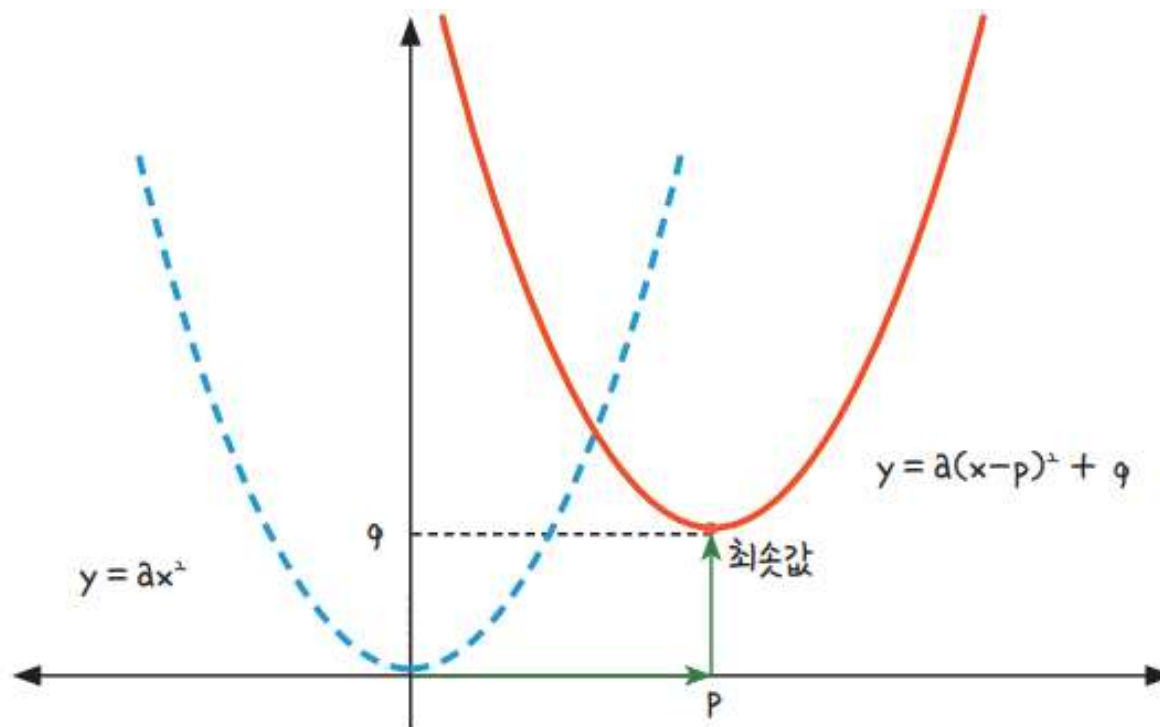
- 이차 함수와 최솟값

- $y = ax^2$ 의 그래프를 x축 방향으로 p만큼, y축 방향으로 q만큼 평행 이동시키면 그림 3-3과 같이 움직임
- 점 p와 q를 꼭짓점으로 하는 포물선이 됨
- 이때 포물선의 맨 아래에 위치한 지점이 **최솟값**이 되는데, 딥러닝을 실행할 때는 이 최솟값을 찾아내는 과정이 매우 중요함

2 이차 함수와 최솟값



▼ 그림 3-3 | 이차 함수 그래프의 평행 이동과 최솟값



2 이차 함수와 최솟값



- 이차 함수와 최솟값

- 이 최솟값은 4장에 소개할 '최소 제곱법' 공식으로 쉽게 알아낼 수 있음
- 딥러닝을 실제로 실행할 때 만나는 문제에서는 대부분 최소 제곱법을 활용할 수가 없음
- 그 이유는 최소 제곱법을 계산하기 위해 꼭 필요한 조건들을 알 수 없기 때문임
- 미분과 기울기를 이용해야 함



3 미분, 순간 변화율과 기울기

3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 딥러닝을 이해하는 데 가장 중요한 수학 원리는 **미분**이라고 할 수 있음
- 조금 전 딥러닝은 결국 일차 함수의 a 와 b 값을 구하는 것인데, a 와 b 값은 이차 함수 포물선의 최솟값을 구하는 것
- 이 최솟값을 미분으로 구하기 때문에 미분이 딥러닝에서 중요한 것
- 미분과 기울기의 개념을 먼저 알아보자

3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 그림 3-4와 같이 $y = x^2$ 이라는 그래프가 있다고 해 보자
- x 축에 있는 한 점 a 에 대응하는 y 의 값은 a^2
- 이때 a 가 오른쪽이나 왼쪽으로 조금씩 이동한다고 상상해 보자
- 이에 따라 y 도 조금씩 변화할 것

3 미분, 순간 변화율과 기울기



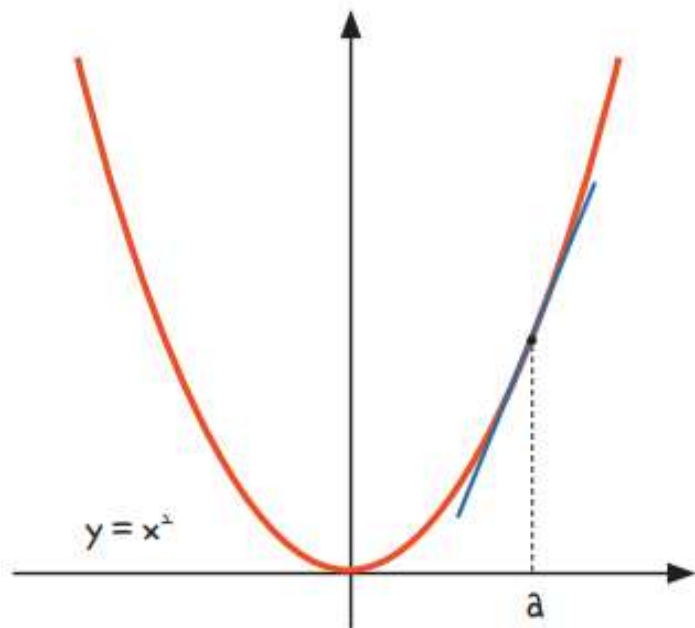
- 미분, 순간 변화율과 기울기

- 상상력을 조금 더 발휘해 이번에는 a 가 미세하게 '0에 가까울 만큼' 움직였다고 하자
- y 값 역시 매우 미세하게 변화를 할 텐데, 이번에는 너무 미세해서 실제로 움직이는 것이 아니라 방향만 드러내는 정도의 순간적인 변화만 있을 것
- 이 순간의 변화를 놓고 **순간 변화율**이라는 이름을 붙였음
- 순간 변화율은 어느 쪽을 향하는 방향성을 지니고 있으므로, 이 방향을 따라 직선을 길게 그려 주면 그래프와 맞닿는 접선이 그려짐
- 이 선이 바로 이 점에서의 **기울기**가 됨

3 미분, 순간 변화율과 기울기



▼ 그림 3-4 | a 에서의 순간 변화율은 곧 기울기다!



3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 미분을 한다는 것은 쉽게 말해 이 '순간 변화율'을 구한다는 것
- 어느 순간에 어떤 변화가 일어나고 있는지 숫자로 나타낸 것을 **미분 계수**라고 하며, 이 미분 계수는 곧 그래프에서의 기울기를 의미
- 이 기울기가 중요한 것은 기울기가 0일 때, 즉 x축과 평행한 직선으로 그어질 때가 바로 그래프에서 최솟값인 지점이 되기 때문임

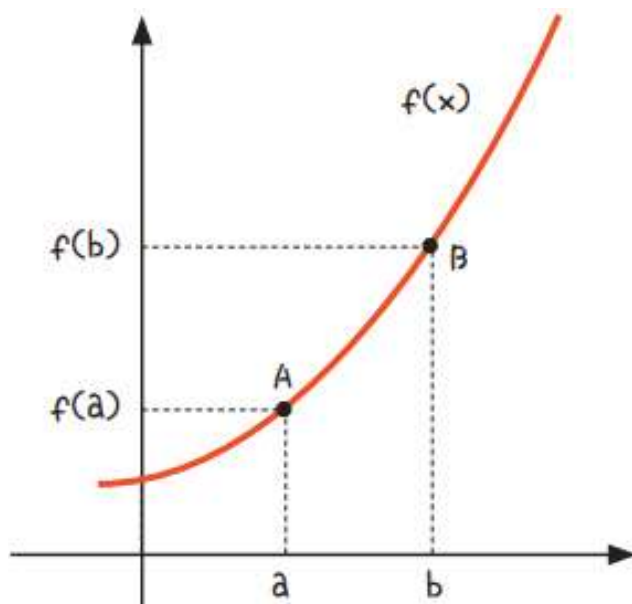
3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 이제 순간 변화율을 구하는 방법을 알아보자
- 어떤 함수 $f(x)$ 가 그림 3-5와 같이 주어졌다고 하자
- 이 함수에 x 축 위의 두 실수 a 와 b 를 대입하면 두 점 A, B 는 그림과 같이 각각 $A(a, f(a))$, $B(b, f(b))$ 에 해당하는 곳에 표시

▼ 그림 3-5 | 함수 $f(x)$ 의 x 축 위에 두 실수 a 와 b 를 대입



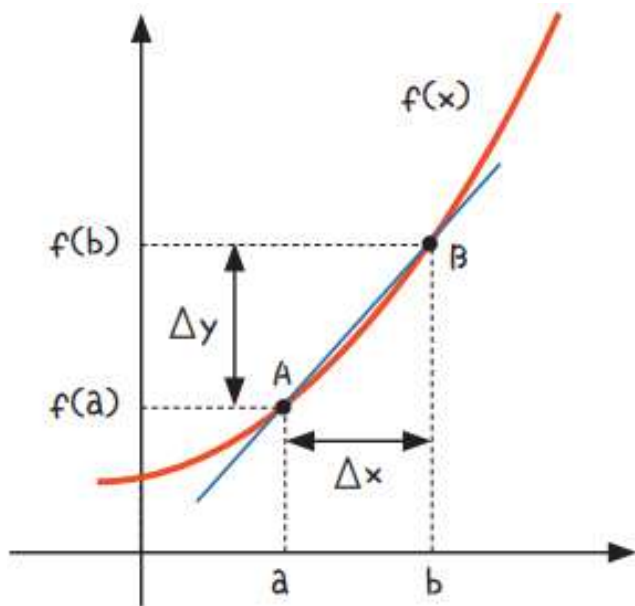
3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 이때 두 점 A와 B를 이어 직선을 만들면 그림 3-6과 같이 두 점 A와 B를 지나는 직선의 기울기가 그려짐
- 여기서 Δ (델타)는 변화량을 나타내는 기호

▼ 그림 3-6 | A와 B를 지나는 직선은 이 두 점 간의 기울기, 곧 평균 변화율을 의미





3 미분, 순간 변화율과 기울기

- 미분, 순간 변화율과 기울기

- 이 그래프에서 x 값의 증가량은 $b - a$ 이고, y 값의 증가량은 $f(b) - f(a)$
- 이를 Δ 를 써서 표현하면 x 값의 증가량은 Δx 로, y 값의 증가량은 $f(a + \Delta x) - f(a)$ 로 나타낼 수 있음
- 직선의 기울기 $\frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}}$ 이라고 했음
- A와 B를 지나는 직선의 기울기는 다음과 같이 표현할 수 있음

$$\text{직선 AB의 기울기} = \frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}} = \frac{f(b) - f(a)}{b - a} = \frac{f(a + \Delta x) - f(a)}{\Delta x}$$



3 미분, 순간 변화율과 기울기

- 미분, 순간 변화율과 기울기

- 이때 직선 AB의 기울기를 A와 B 사이의 '평균 변화율'이라고도 함
- 미분을 배우고 있는 우리에게 필요한 것은 **순간 변화율**
- 순간 변화율은 x의 증가량(Δx)이 0에 가까울 만큼 아주 작을 때의 순간적인 기울기를 의미하므로, 극한(limit) 기호를 사용해 다음과 같이 나타냄

$$\lim_{\Delta x \rightarrow 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$



3 미분, 순간 변화율과 기울기

- 미분, 순간 변화율과 기울기

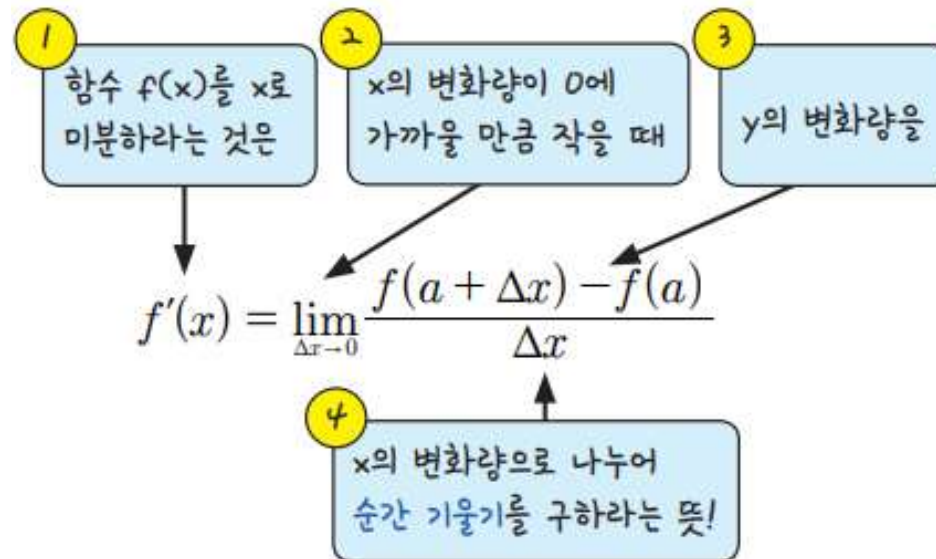
- 여기서 $\lim_{\Delta x \rightarrow 0}$ 는 'x의 증가량이 0에 가까울 만큼 작을 때'라는 뜻
- 기울기는 $\frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}}$ 이므로 순간 기울기는 $\lim_{\Delta x \rightarrow 0} \frac{y \text{ 값의 증가량}}{x \text{ 값의 증가량}}$ 으로 표현되며,
이것은 $\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx}$
라고도 쓸 수 있음



3 미분, 순간 변화율과 기울기

- 미분, 순간 변화율과 기울기

- "함수 $f(x)$ 를 미분하라"는 것을 $f'(x)$ 또는 $\frac{d}{dx}f(x)$ 로 표기하는데, 함수 $f(x)$ 를 미분하는 공식을 알기 쉽게 정리하면 다음과 같음



3 미분, 순간 변화율과 기울기



- 미분, 순간 변화율과 기울기

- 다음은 딥러닝을 공부하는 과정 중에 자주 만나게 되는 중요한 다섯 가지 미분의 기본 공식

미분의 기본 공식

1 | $f(x) = x$ 일 때 $f'(x) = 1$

2 | $f(x) = a$ 에서 a 가 상수일 때 $f'(x) = 0$

3 | $f(x) = ax$ 에서 a 가 상수일 때 $f'(x) = a$

4 | $f(x) = x^a$ 에서 a 가 자연수일 때 $f'(x) = ax^{a-1}$

5 | $f(g(x))$ 에서 $f(x)$ 와 $g(x)$ 가 미분 가능할 때 $\{f(g(x))\}' = f'(g(x)) \times g'(x)$



4 편미분

4 편미분



● 편미분

- 미분과 더불어 딥러닝을 공부할 때 가장 자주 접하게 되는 또 다른 수학 개념은 바로 편미분
- 미분과 편미분 모두 '미분하라'는 의미에서는 다를 바가 없음
- 여러 가지 변수가 식 안에 있을 때, 모든 변수를 미분하는 것이 아니라 우리가 원하는 한 가지 변수만 미분하고 그 외에는 모두 상수로 취급하는 것이 바로 편미분
- 예를 들어 $f(x) = x$ 와 같은 식을 미분할 때는 변수가 x 하나뿐이어서 미분하라는 의미에 혼란이 없음
- 다음 식을 보자

$$f(x, y) = x^2 + yx + a \text{ (} a \text{는 상수)}$$

4 편미분



- 편미분

- 여기에는 변수가 x 와 y , 이렇게 두 개 있음
- 이 중 어떤 변수로 미분해야 하는지 정해야 하므로 편미분을 사용하는 것
- 만일 이 식처럼 여러 변수 중에서 x 에 관해서만 미분하고 싶다면, 함수 f 를 ' x 에 관해 편미분하라'고 하며 다음과 같이 식을 씀

$$\frac{\partial f}{\partial x}$$

4 편미분



● 편미분

- 앞에 나온 함수 $f(x, y) = x^2 + yx + a$ 를 x 에 관해 편미분하는 과정은 어떻게 될까?
- 먼저 바로 앞에서 배운 미분의 성질 4에 따라 x^2 항은 $2x$ 가 됨
- 미분법의 기본 공식 3에 따라 yx 는 y 가 됨
- 마지막 항 a 는 미분의 성질 1에 따라 0이 됨
- 이를 정리하면 다음과 같이 나타낼 수 있음

$f(x, y) = x^2 + yx + a$ 일 때

$$\frac{\partial f}{\partial x} = 2x + y$$



5 지수와 지수 함수

5 지수와 지수 함수



- 지수와 지수 함수
 - 지수란 다음과 같은 형태를 의미

$$a^{\square}$$

5 지수와 지수 함수



- 지수와 지수 함수

- 여기서 a 를 '밑'이라 하고 x 를 '지수'라고 함
- a 를 x 만큼 반복해서 곱한다는 뜻
- 지수 함수란 변수 x 가 지수 자리에 있는 경우를 의미
- 식으로 나타내면 다음과 같은 형태

$$y = a^x \quad (a \neq 1, a > 0)$$

5 지수와 지수 함수



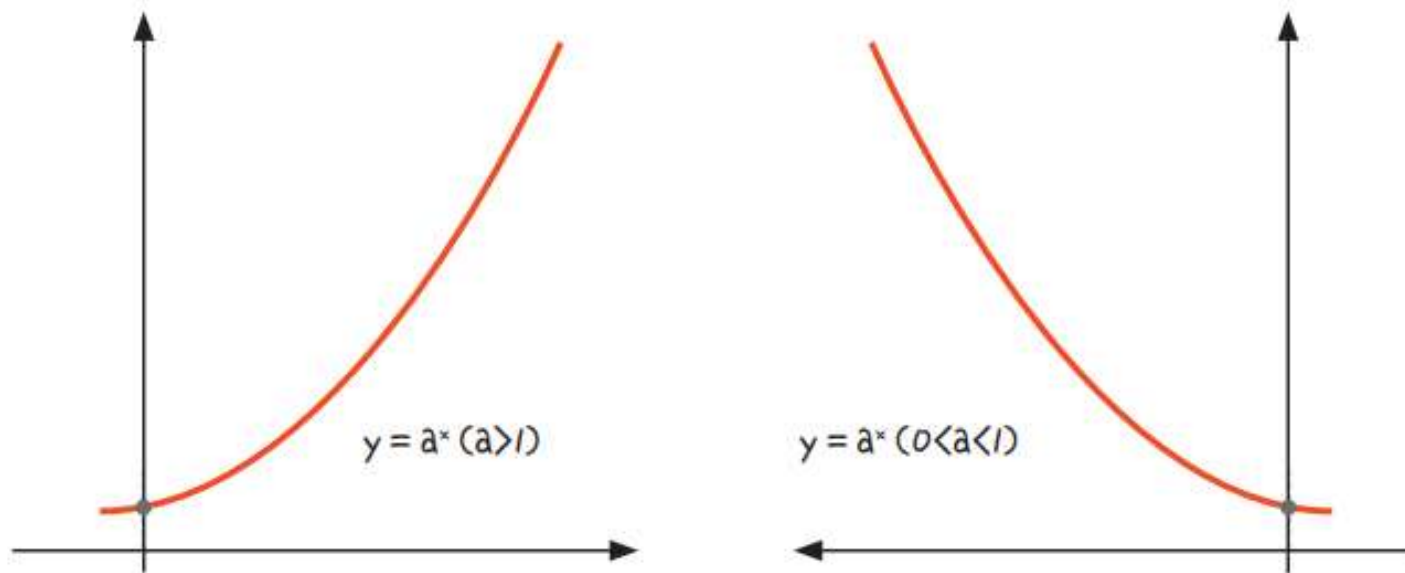
- 지수와 지수 함수

- 지수 함수에서는 밑(a) 값이 무엇인지가 중요함
- 이 값이 1이면 함수가 아님
- 또 0보다 작으면 허수를 포함하게 되므로 안 됨
- 밑의 값은 $a > 1$ 이거나 $0 < a < 1$, 둘 중 하나가 되어야 함
- 이 두 가지 경우의 그래프는 각각 그림 3-7과 같음

5 지수와 지수 함수



▼ 그림 3-7 | $a > 1$ 일 때와 $0 < a < 1$ 일 때의 지수 함수





6 시그모이드 함수

6 시그모이드 함수



● 시그모이드 함수

- 딥러닝의 내부를 보면 입력받은 신호를 얼마나 출력할지를 계산하는 과정이 무수히 반복
- 이때 출력 값으로 얼마나 내보낼지를 계산하는 함수를 활성화 함수라고 함
- 활성화 함수는 딥러닝이 발전함에 따라 여러 가지 형태로 개발되어 왔는데, 그중 가장 먼저 배우는 중요한 함수가 바로 시그모이드 함수
- 시그모이드 함수는 지수 함수에서 밑 값이 자연 상수 e 인 함수를 의미
- 자연 상수 e 는 '자연 로그의 밑', '오일러의 수' 등 여러 이름으로 불리는데, 파이(π)처럼 수학에서 중요하게 사용되는 무리수이며 그 값은 대략 2.718281828...

6 시그모이드 함수



- 시그모이드 함수

- 자연 상수 e 가 지수 함수에 포함되어 분모에 들어가면 시그모이드 함수가 되는데, 이를 식으로 나타내면 다음과 같음

$$f(x) = \frac{1}{1 + e^{-x}}$$

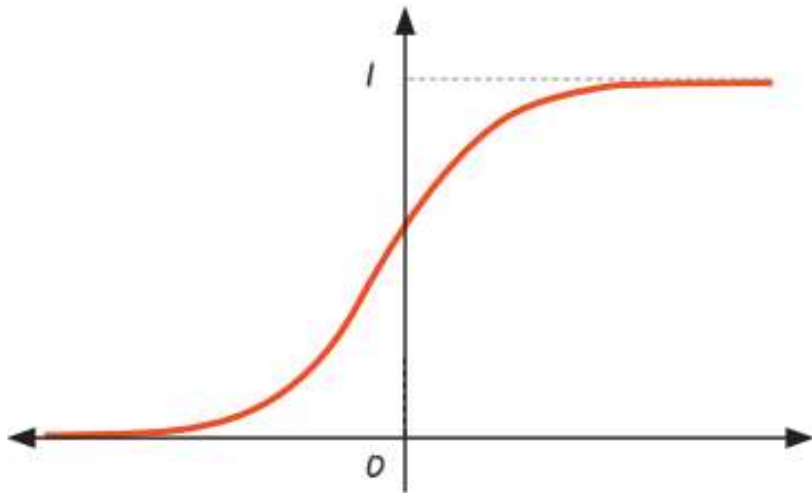
6 시그모이드 함수



- 시그모이드 함수

- 시그모이드 함수를 그래프로 그려 보면 그림 3-8과 같이 S자 형태로 나타남

- ▼ 그림 3-8 | 시그모이드 함수의 그래프



6 시그모이드 함수



- 시그모이드 함수

- x 가 큰 값을 가지면 $f(x)$ 는 1에 가까워지고, x 가 작은 값을 가지면 $f(x)$ 는 0에 가까워짐
- S자 형태로 그려지는 이 함수의 속성은 0 또는 1, 두 개의 값 중 하나를 고를 때 유용하게 쓰임



7 로그와 로그 함수

7 로그와 로그 함수



- 로그와 로그 함수

- 로그를 이해하려면 먼저 지수부터 이해해야 함
- a를 x만큼 거듭제곱한 값이 b라고 할 때, 이를 식으로 나타내면 다음과 같음

$$a^x = b$$

7 로그와 로그 함수



● 로그와 로그 함수

- 이때 a 와 b 를 알고 있는데 x 를 모른다고 해 보자
- x 는 과연 어떻게 구할 수 있을까?
- 이 x 를 구하기 위해 사용하는 방법이 로그
- 영어로 Logarithm이라고 하는데 앞 세 글자 \log 를 사용해서 표시하며, 지수식에서 a 와 b 의 위치를 다음과 같이 바꾸어 쓰면 됨

$$\begin{array}{c} a^x = b \\ \swarrow \quad \searrow \\ \log_a b = x \end{array}$$

- 로그가 지수와 이렇게 밀접한 관계가 있듯이 로그 함수 역시 지수 함수와 밀접한 관계에 있음
- 바로 역함수의 관계
- 역함수는 x 와 y 를 서로 바꾸어 가지는 함수

7 로그와 로그 함수



- 로그와 로그 함수

- 지수 함수 $y = a^x (a \neq 1, a > 0)$ 는 로그 정의를 따라 $x = \log_a y$ 로 바꿀 수 있음
- 역함수를 만들기 위해 x 와 y 를 서로 바꾸어 주면 됨
- 다음 식이 바로 **로그 함수**의 형태

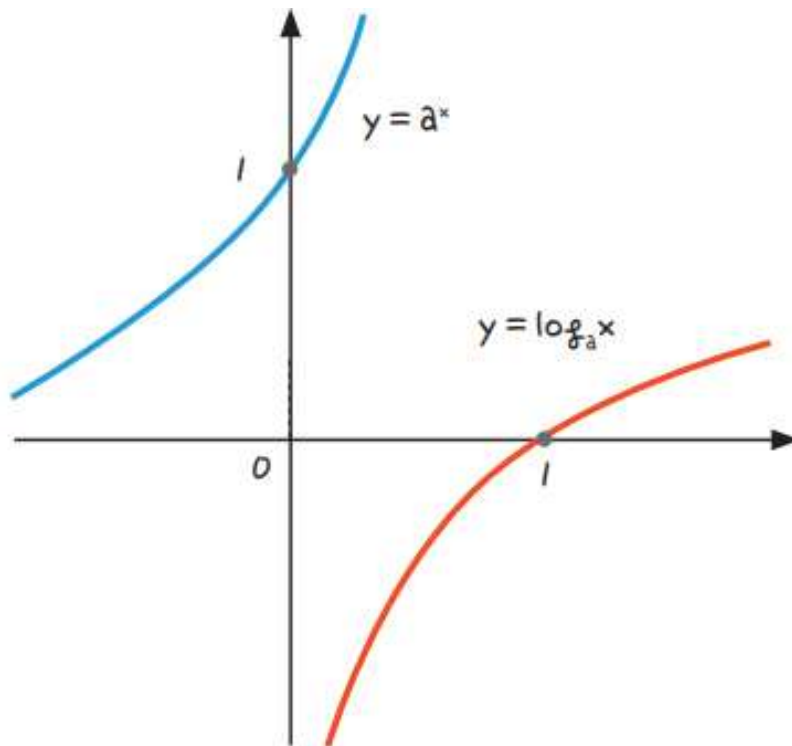
7 로그와 로그 함수



● 로그와 로그 함수

- 역함수의 그래프는 $y = x$ 에 대해 대칭인 선으로 나타남
- 그림 3-9는 지수 함수 $y = a^x$ 의 그래프를 $y = x$ 에 대칭으로 이동시킨 로그 함수 $y = \log_a x$ 의 그래프를 보여 줌

▼ 그림 3-9 | 지수 함수 $y = a^x$ 와 로그 함수 $y = \log_a x$



7 로그와 로그 함수



- 로그와 로그 함수

- 6장에서 로지스틱 회귀를 배울 때, 우리는 x 가 1에 가까워지거나 0에 가까워질수록 오차가 커지는 그래프가 필요함
- 이러한 그래프를 만들기 위해 $y = \log_a x$ 를 x 축 또는 y 축으로 대칭 이동하거나 알맞게 평행 이동하면 다음과 같음

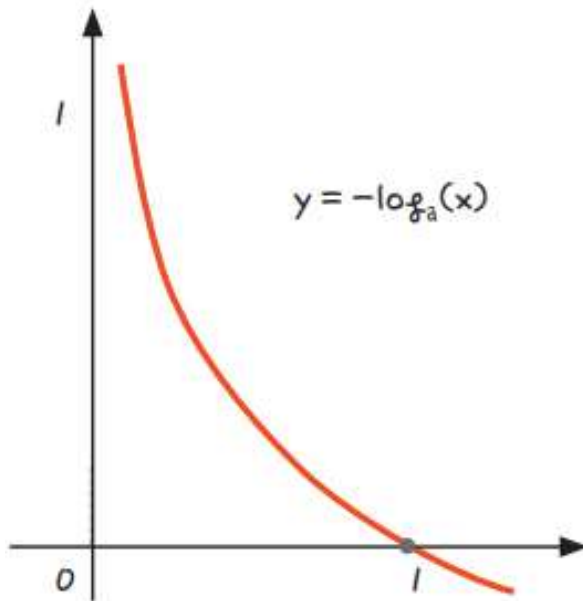
7 로그와 로그 함수



- 로그와 로그 함수

1 | x축에 대해 대칭 이동

▼ 그림 3-10 | $y = -\log_a(x)$ 그래프



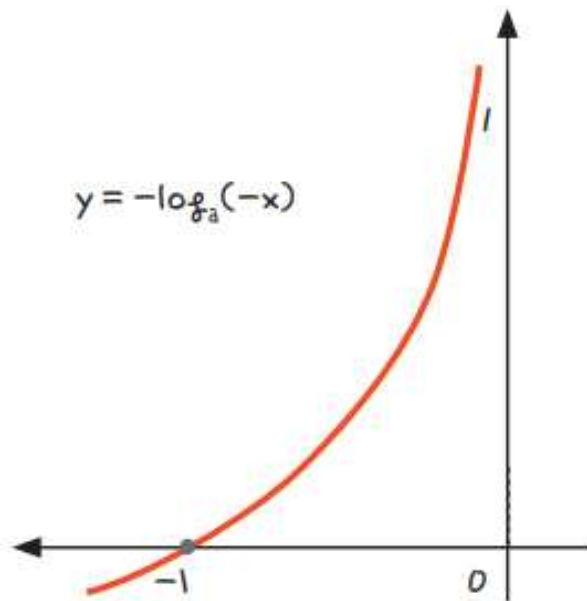
7 로그와 로그 함수



- 로그와 로그 함수

2 | x축과 y축에 대해 대칭 이동

▼ 그림 3-11 | $y = -\log_a(-x)$ 그래프



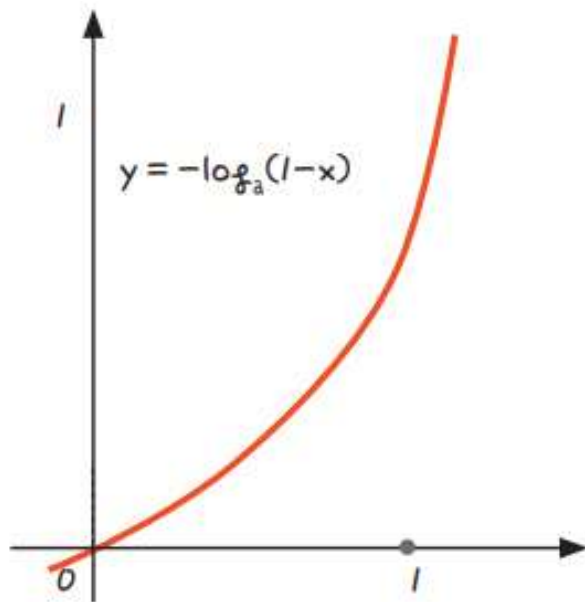
7 로그와 로그 함수



- 로그와 로그 함수

3 | 2의 그래프를 x축 오른쪽 방향으로 1만큼 평행 이동

▼ 그림 3-12 | $y = -\log_a(1-x)$ 그래프



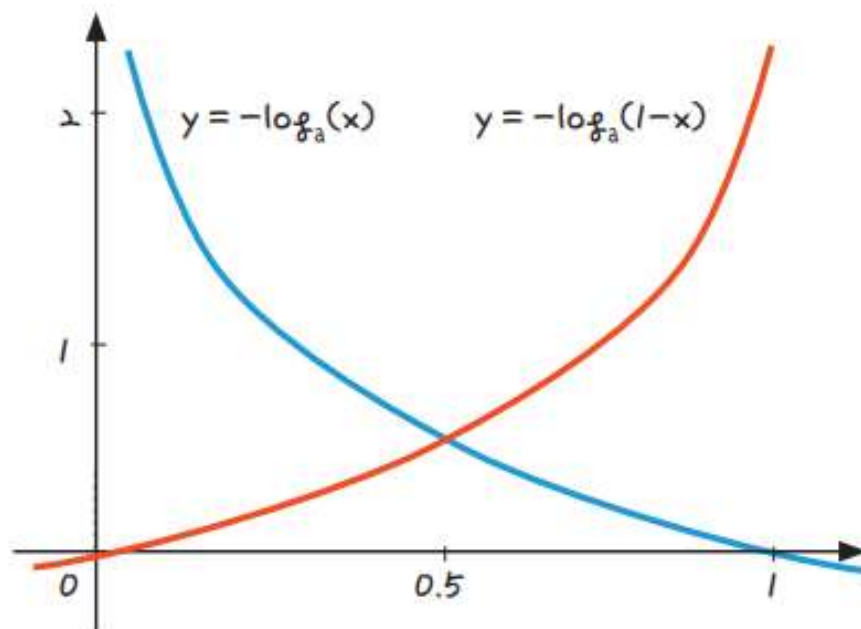
7 로그와 로그 함수



- 로그와 로그 함수

4 | 1과 3을 함께 나타낸 그래프

▼ 그림 3-13 | $y = -\log_a(x)$ 와 $y = -\log_a(1-x)$ 그래프



7 로그와 로그 함수



● 로그와 로그 함수

- 지금까지 설명한 일차 함수, 이차 함수, 미분, 편미분, 지수 함수, 시그모이드 함수 그리고 로그 함수 이렇게 일곱 가지를 알고 있으면 4~22장 내용을 모두 이해할 수 있음
- 여기에 합을 표현하기 위해 만들어진 \sum (시그마) 기호가 종종 나옴
- $\sum_{i=1}^n F(i)$ 라고 하면 i 를 1부터 n 까지 $F(i)$ 에 대입해 더하라는 뜻
- 즉, $F(1) + F(2) + F(3) + \dots + F(n)$ 이 됨
- 나머지 어려운 증명이나 체인 룰이 등장하는 수식의 계산은 딥러닝 활용 편을 모두 마치고 나서 이어지는 심화 학습 편에서 다룸
- 심화 학습 편을 공부하지 않아도 이 책에 나오는 모든 딥러닝 예제를 이해하고 실행하는 데는 문제없음