

인공지능수학

1주차: 인공지능과 수학

1차시: 인공지능의 발전

인공지능의 정의

- 반드시 최신의 생성형 인공지능처럼 행동해야만 하는 것을 가리키는 것이 아님
- 인간이 수행하는 절차를 기계가 대신 할 수 있도록 만들면 그것이 인공지능

인공지능의 발전

- 논리/규칙 기반의 전문가 시스템
- 연결 기반의 인공신경망
- 통계 기반의 기계학습
- 인공신경망 기반의 딥러닝

이차방정식의 풀이법으로 알아보는 인공지능

- 논리/규칙 기반으로 해결하는 이차방정식: 우리가 잘 알고 있는 근의 공식입니다
- 인공신경망 기반의 딥러닝으로 해결하는 이차방정식도 가능합니다

사람이 만드는 함수와 AI가 만드는 모델

- 수학시간에 이차방정식의 계수는 사람이 정의했습니다
- 이 방식은 함수만 한번 정의하면 모든 x 에 대해 정확한 y 가 산출 됩니다

$$y = 2x^2 + 3x + 4$$

- AI에서는 x 와 y 를 제공하고 이차방정식의 계수를 컴퓨터가 만듭니다
- 이 방식은 하나의 x, y 쌍만 가지고는 모든 x 에 대한 y 의 관계를 서술할 수 없습니다

$$y = 2x^2 + 3x + 4$$

사람이 만드는 함수와 AI가 만드는 모델

- $(x, y) = (0, 4)$ 으로 아래의 이차방정식 계수를 만들어봅시다

$$y = ?x^2 + ?x + ?$$

- $a = 0, b = 0, c = 4$ 이면 되겠습니다

$$y = 0x^2 + 0x + 4$$

- 그러나 이 계수로는 $(x, y) = (1, 9)$ 를 충족할 수 없습니다

$$y = 2x^2 + 3x + 4$$

$$y = 2 + 3 + 4(x = 1)$$

사람이 만드는 함수와 AI가 만드는 모델

- $(x, y) = (0, 4)$ 과 $(x, y) = (1, 9)$ 2개 데이터로 계수를 만들어봅시다
- 연립방정식 해법을 고려하지 않고 생각합니다

$$y = 3x^2 + 2x + 4$$

- 그러나 이 함수는 $(x, y) = (-1, 3)$ 은 만족하지 않습니다
- 우리가 원했던 계수는 아래와 같았습니다
- 그러나 주어진 데이터만으로는 아래 계수를 도출하기 어렵습니다

$$y = 2x^2 + 3x + 4$$

사람이 만드는 함수와 AI가 만드는 모델

- 하나의 데이터만 더 있으면 연립방정식 해법으로 3개 계수를 모두 맞출 수 있지 않았을까요?
- 그러나 AI가 만드는 모델이라는 것은 연립방정식 해법을 사용하지 않습니다
- 임의의 계수를 설정하고 그 계수로 연산을 해본 후 오차를 보정하는 방식으로 계수를 맞추어 나갑니다
- 우리는 이것을 학습이라고 합니다
- 이 방식을 오늘 3차시에 Google Colab 환경에서 코드를 실행하며 알아볼 것입니다

전통적 인공지능과 현대 인공지능의 장단점

- 데이터는 조금밖에 없지만 규칙이 확실하면 전통적 인공지능이 유리
- 규칙을 우리가 찾기는 힘들지만 데이터가 많이 있다면 현대 인공지능(딥러닝)이 유리

지도학습, 비지도학습, 강화학습

- 지도학습: 정답을 알려주고 학습
- 비지도학습: 정답을 알려주지 않고 학습
- 강화학습: 행동에 따른 보상을 통해 학습

딥러닝 학습에서의 가중치와 편향

- 신경세포에서 입력이 전기 신호라면 일차방정식에서 x 라고 볼 수 있습니다
- 신경세포에서 입력에 따라 전달되는 전기 신호는 출력이므로 일차방정식에서 y 라고 볼 수 있습니다
- 이 때 동일한 입력에 대해서 기울기 a 에 따라 y 값이 좌우됩니다
- 그래서 기울기를 가중치라고 하고 weight의 w 를 따서 a 대신 사용하기도 합니다
- 이 때 y 절편은 입력과 무관하게 작용합니다
- 입력과 무관하게 역치에 영향을 줄 수 있습니다
- 그래서 b 를 편향이라고 하고 bias의 b 를 따서 그대로 b 로 사용합니다
- 인공지능의 각 신경세포는 이렇게 일차방정식들의 결합으로 이루어져서 입력신호를 전달하고 역치에 따라 반응한다고 보면 됩니다

$$y = wx + b$$

지뢰찾기 인공지능

- 깃발이 있는 곳은 명확하게 지뢰가 있는 곳입니다
- 나머지는 경우의 수를 확인하고 확률을 기반으로 찍어야 합니다
- 경우의 수는 어떻게 확인할까요?
- 아래의 경우는 전통적 인공지능을 만드는 알고리즘의 일부입니다

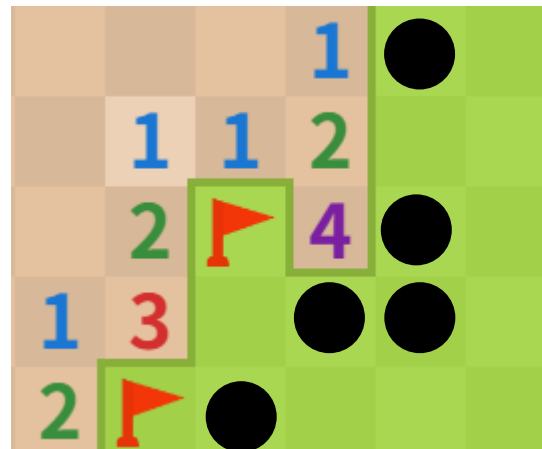
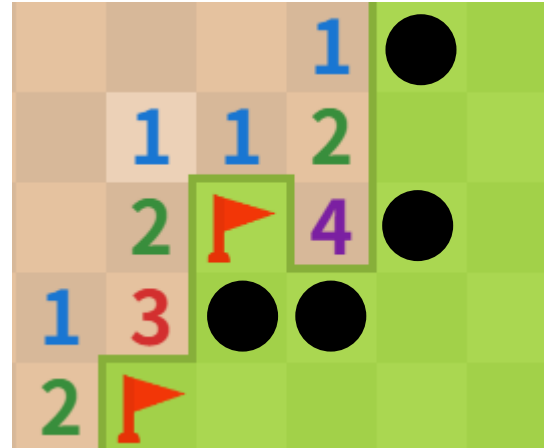


이곳에 지뢰가 있다고 가정하고 나머지 경우를 확인



이곳에 지뢰가 있다고 가정하고 나머지 경우를 확인

지뢰찾기 인공지능



지뢰찾기 인공지능

- 논리를 요약해 봅시다
- 왼쪽 위부터 미지의 공간에 지뢰가 있다고 가정 → 경우의 수 구하기
- 왼쪽 위부터 미지의 공간에 지뢰가 없다고 가정 → 경우의 수 구하기

- 각 경우의 수는 어떻게 구했나요?
- 왼쪽 위를 제외하고 그 다음 미지의 공간에 지뢰가 있다고 가정 → 경우의 수 구하기
- 왼쪽 위를 제외하고 그 다음 미지의 공간에 지뢰가 없다고 가정 → 경우의 수 구하기

- 이것도 인공지능 알고리즘입니다
- 전통적 방식으로 알고리즘을 만드는 것입니다

지뢰찾기 인공지능을 지도학습으로 만든다면



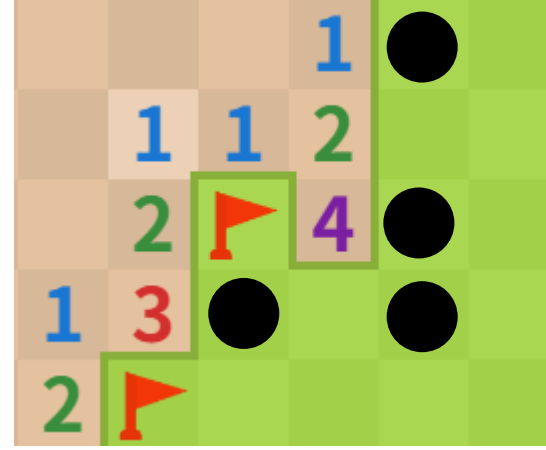
입력값1



결괏값1



입력값2



결괏값2



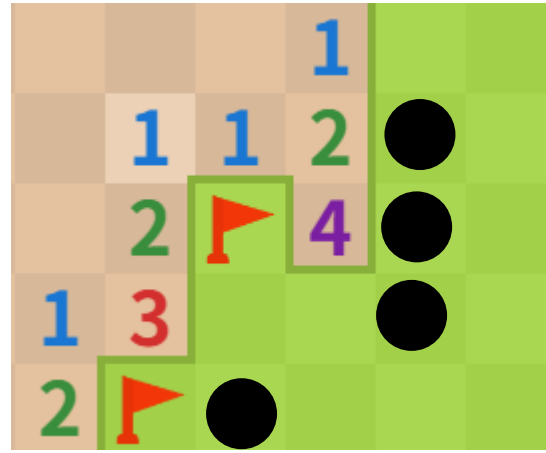
입력값3



결괏값3



입력값4



결괏값4

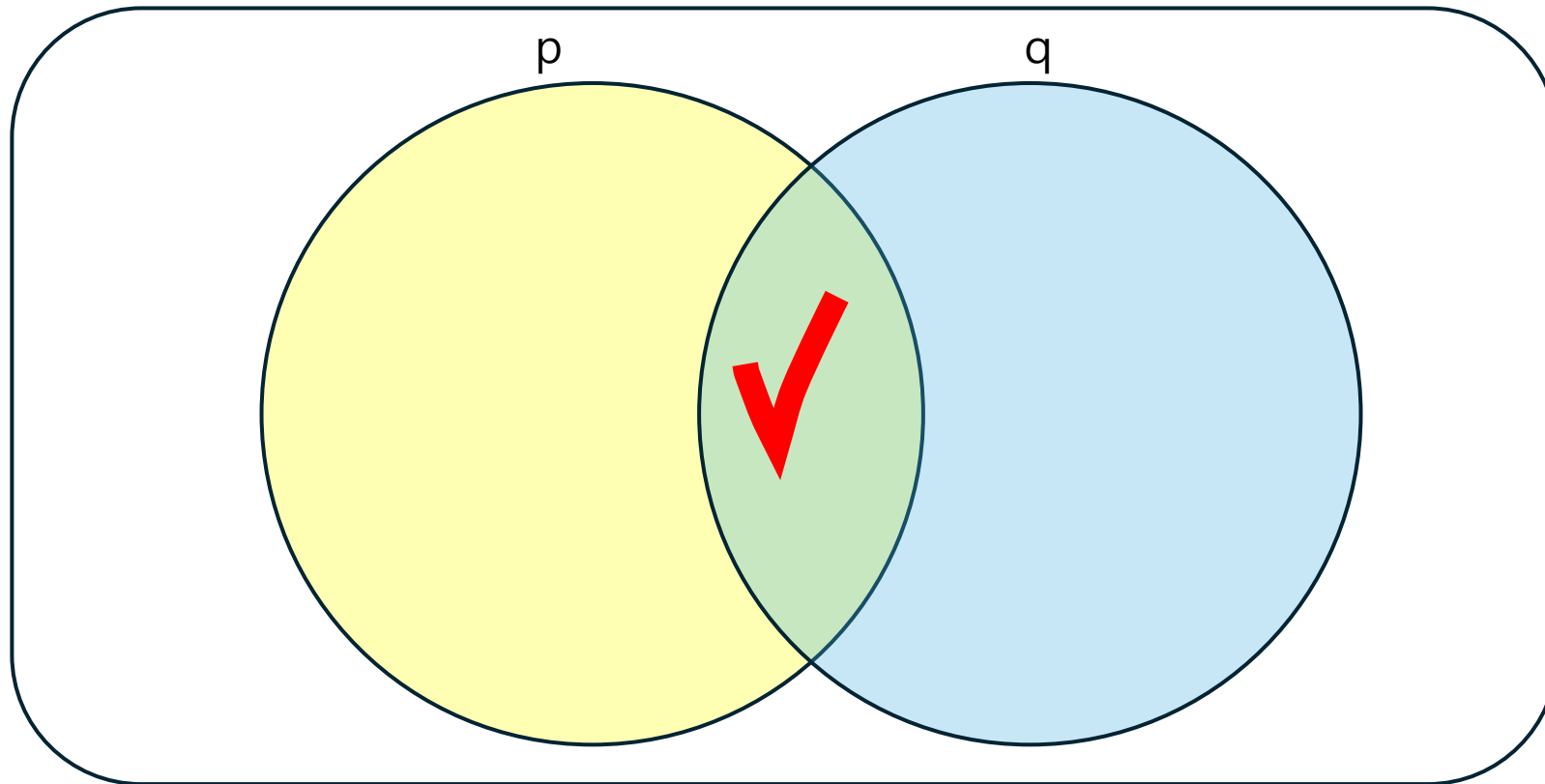
지뢰찾기 인공지능을 강화학습으로 만든다면



2차시: 인공지능과 수학

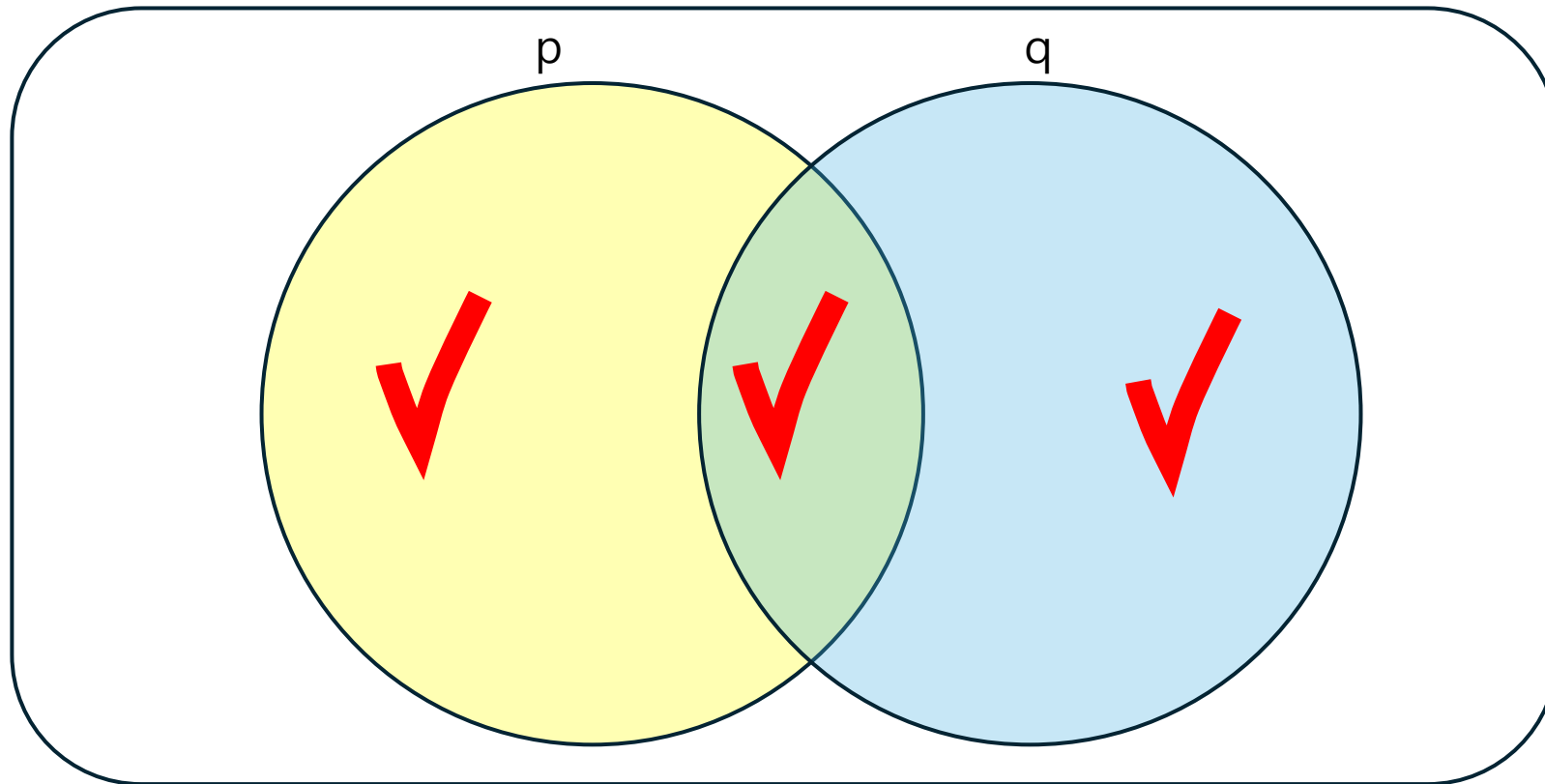
알고보면 집합과 같은 논리연산자

- 명제를 집합으로 놓고 생각해봅시다
- 명제 p 가 참이고 동시에 명제 q 가 참이라면 $p \cap q$
- 이것을 논리연산자에서는 AND라고 합니다($p \text{ AND } q$)



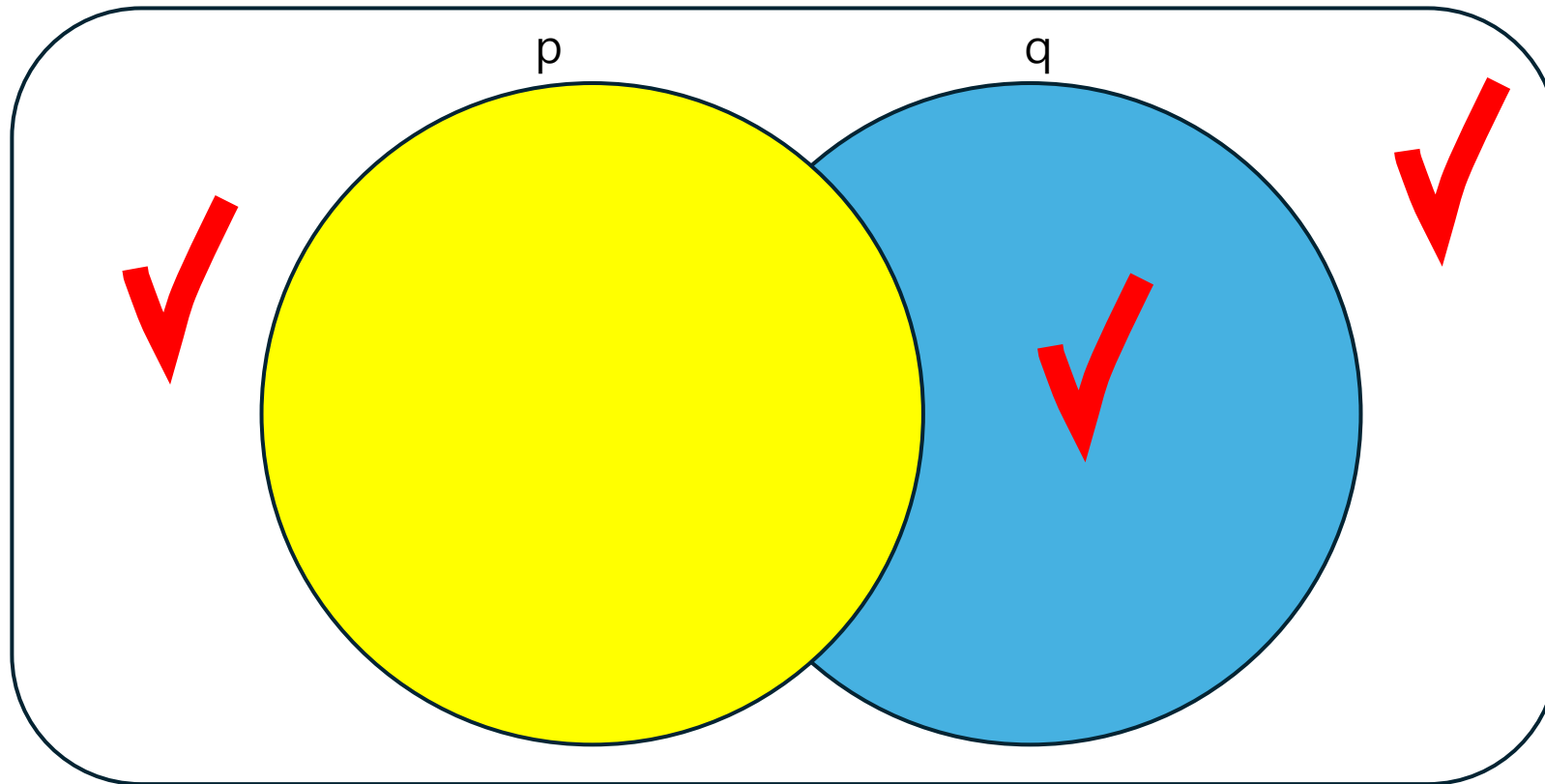
알고보면 집합과 같은 논리연산자

- 명제를 집합으로 놓고 생각해봅시다
- 명제 p 가 참이거나 명제 q 가 참이라면 $p \cup q$
- 이것을 논리연산자에서는 OR라고 합니다 ($p \text{ OR } q$)



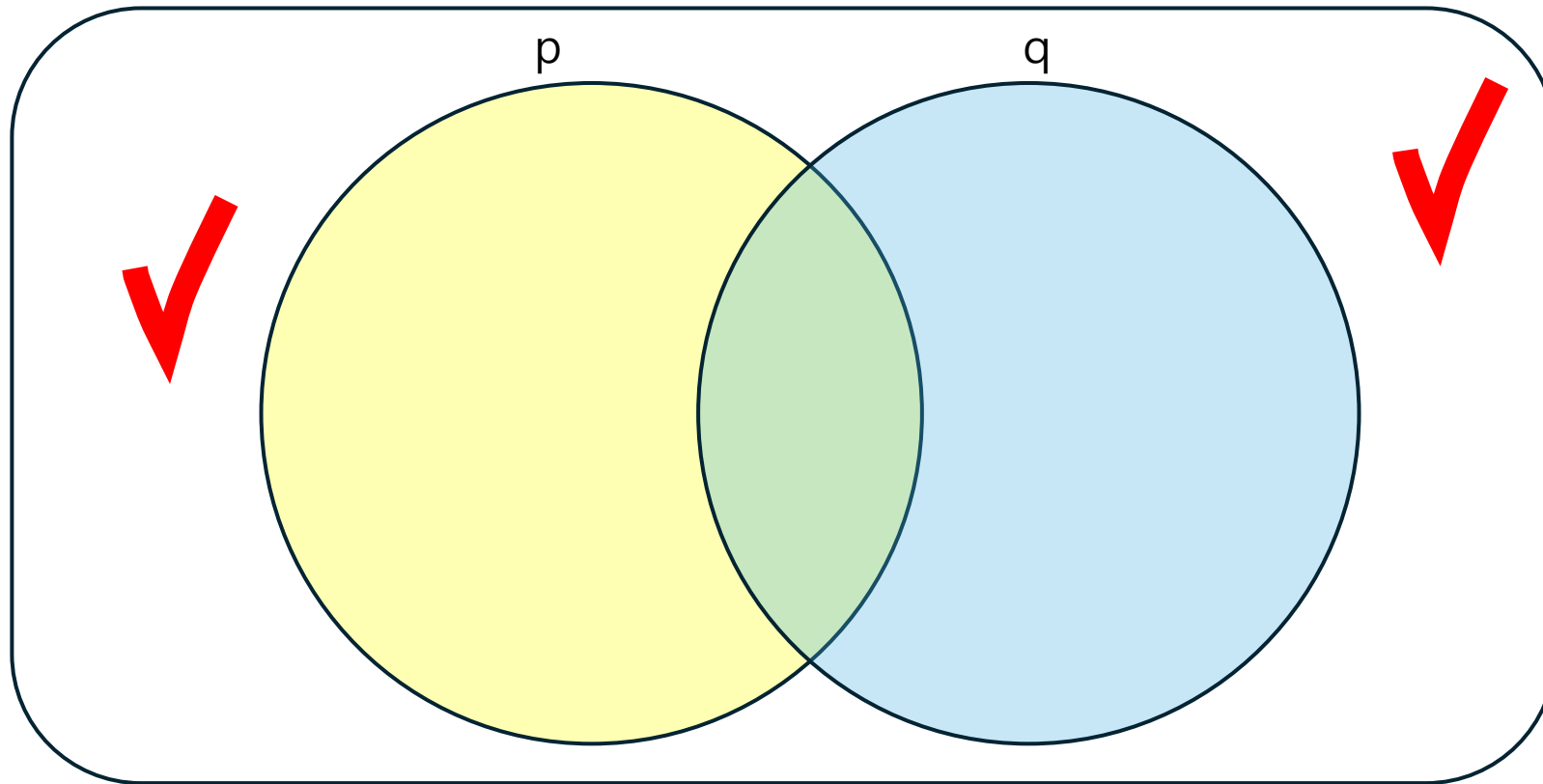
알고보면 집합과 같은 논리연산자

- 명제를 집합으로 놓고 생각해봅시다
- 명제 p 가 거짓이면 p^c
- 이것을 논리연산자에서는 NOT이라고 합니다 (NOT p)



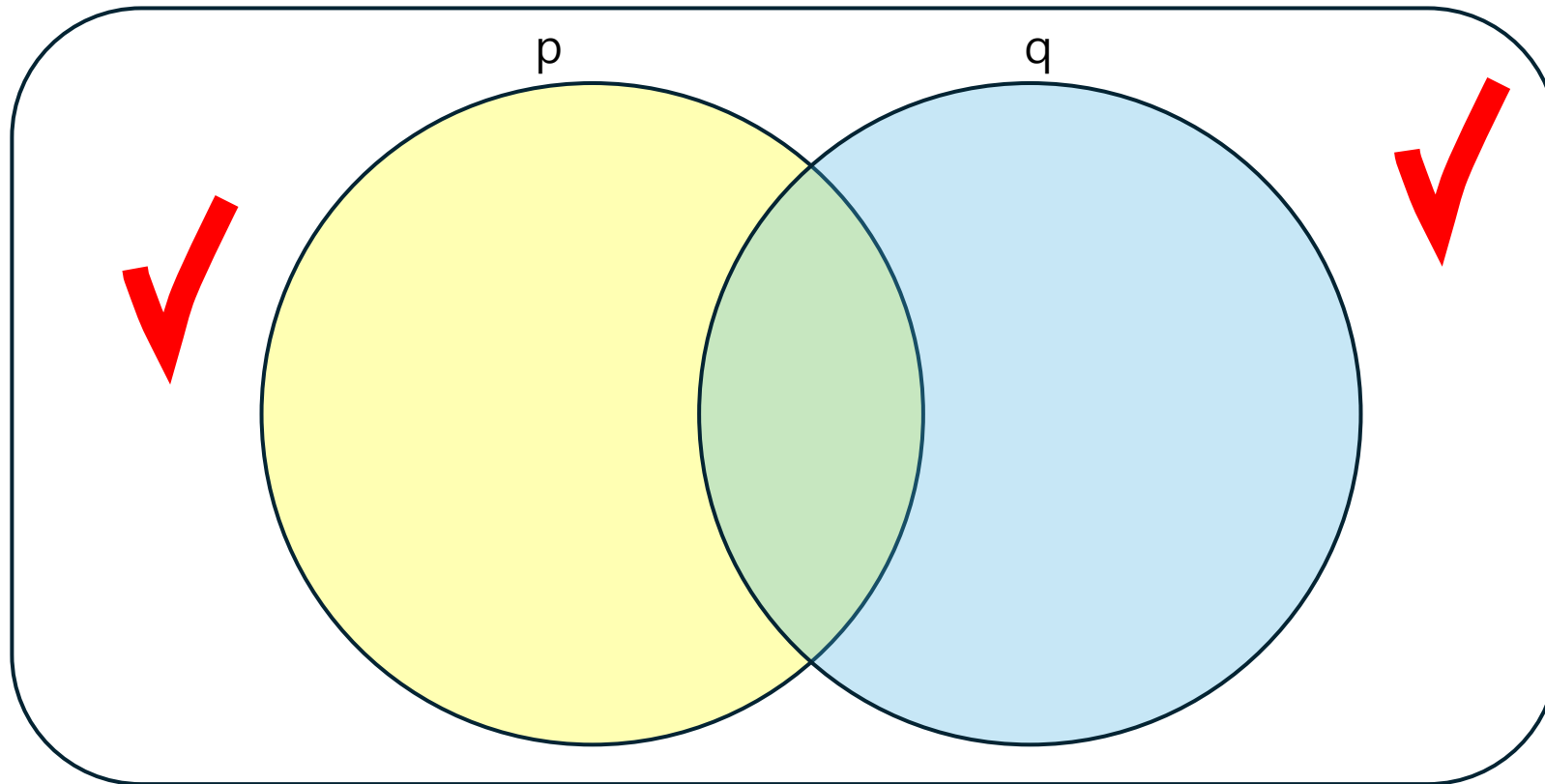
알고보면 집합과 같은 논리연산자

- 명제를 집합으로 놓고 생각해봅시다
- 명제 p 도 거짓이고 명제 q 도 거짓이라면 $(p \cup q)^c$
- 논리 연산자로는 $\text{NOT}(p \text{ AND } q)$, 기호로는 $\sim(p \wedge q)$ 가 됩니다



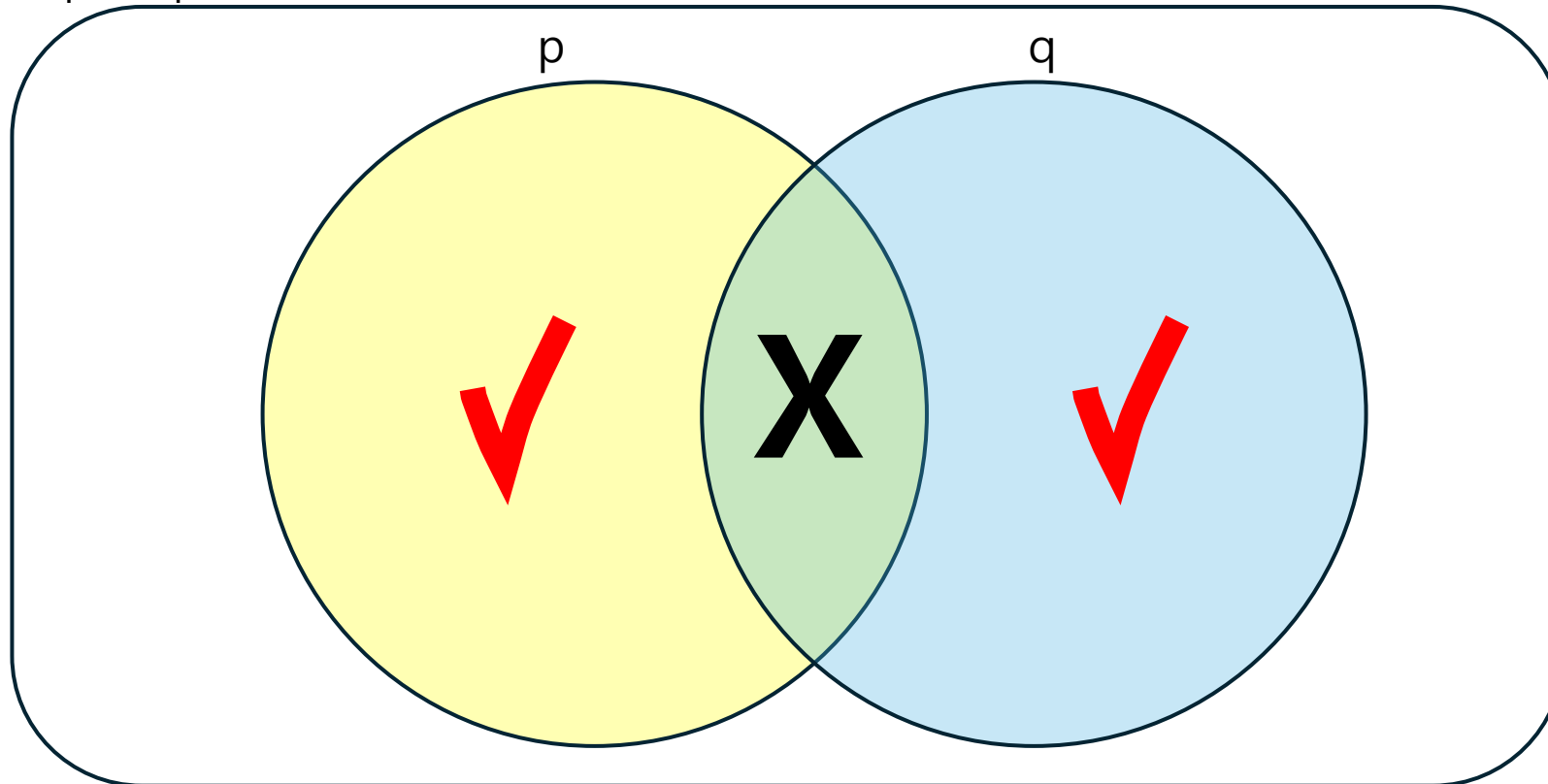
알고보면 집합과 같은 논리연산자

- 드모르간의 법칙도 적용됩니다
- 명제 p 도 거짓이고 명제 q 도 거짓이라면 $(p \cup q)^c = (p^c \cap q^c)$
- 논리 연산자로 변환하면 (NOT p) AND (NOT q) 이므로 p 도 거짓이고 q 도 거짓입니다



XOR연산자

- Exclusive OR 라고 합니다
- 배타적 OR, 즉 p만 참이거나 q만 참인 것입니다
- 집합으로 변환하면 p에만 속해있거나 q에만 속해있는 것이 되고 번역하면 $(p - q) \cup (q - p)$ 이 됩니다
- 이는 $(p \cup q) - (p \cap q)$ 와 같습니다



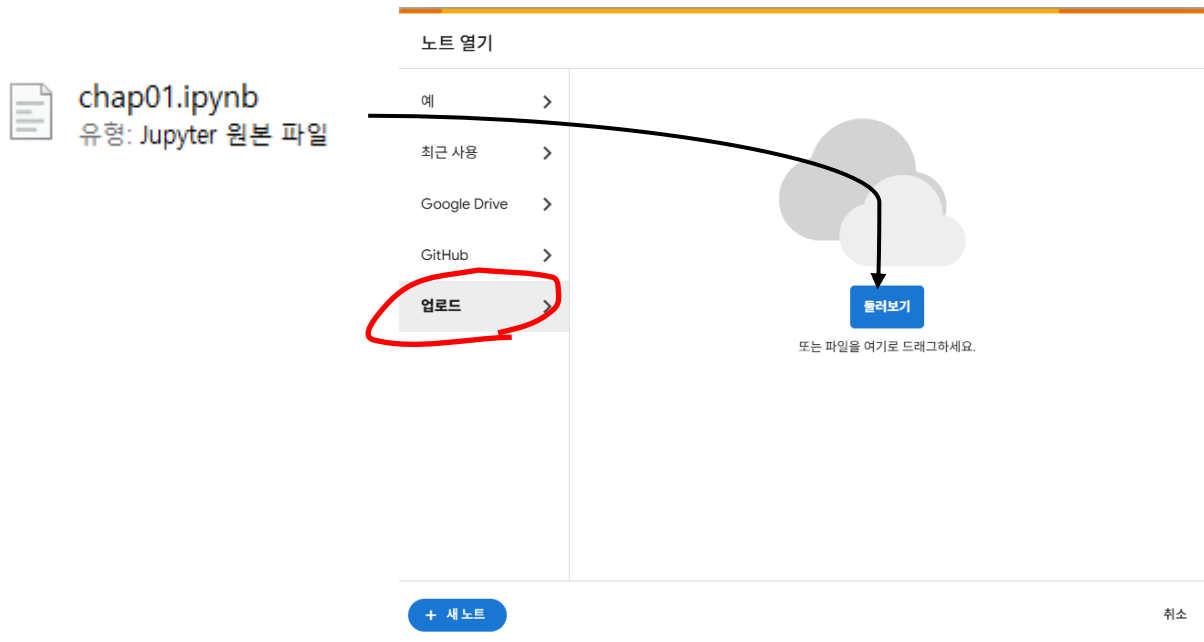
순서도

- 문제의 해결절차를 기호화 한 것입니다
 - 오늘날 프로그래밍 에서는 순서도 외에도 다양한 방법을 많이 사용합니다
 - 순서도를 그릴 때 장점과 단점은 무엇일까요?
-
- 장점: 몇가지 기호만 알면 순서도를 그릴 수 있습니다
 - 단점: 순서도로 모든 논리를 다 표현하지 못합니다

3차시: Google colab 사용해보기

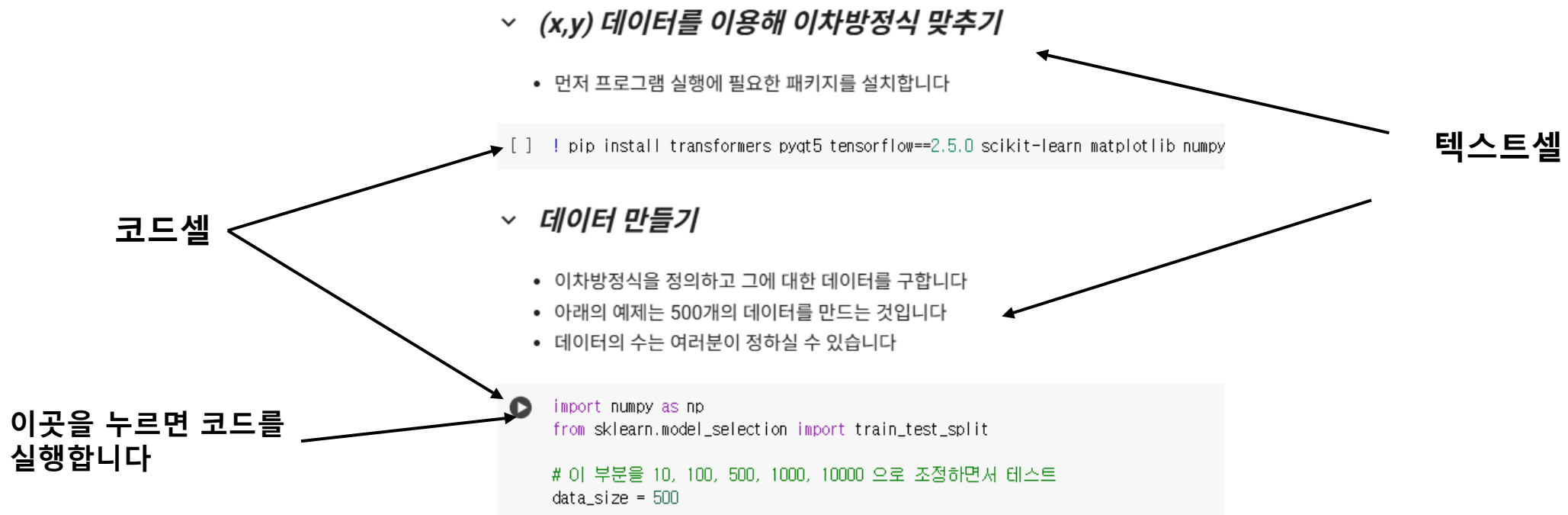
무료 클라우드에서 배우는 인공지능 수학

- 우리 수업은 앞으로 Google colab 이라는 무료 클라우드 환경을 병행하여 진행합니다
- 우선 https://github.com/suriseven/mathematics_for_AI/blob/main/chap01/chap01.ipynb 을 다운로드 받습니다
- Google 로그인 된 상태에서 <https://colab.research.google.com/> 으로 이동합니다
- 다운로드한 파일을 드래그 하여 아래와 같이 드롭합니다



코드셀과 텍스트셀

- 실습환경은 아래와 같이 코드셀과 텍스트셀로 나누어져 있습니다
- 텍스트셀은 설명을 위한 공간입니다
- 코드셀이 실제로 코드를 실행하는 공간입니다
- 코드 실행 버튼을 누르면 결과를 확인할 수 있습니다



실행결과의 확인

- 코드의 실행 결과는 코드셀의 아래에서 확인할 수 있습니다
- 코드가 변경되면 실행결과도 바뀔 수 있습니다
- 그러면 코드를 변경해봅시다

이곳을 변경하면서 결과가 바뀌는 것을 확인해봅시다

코드셀

실행결과

```
import numpy as np
from sklearn.model_selection import train_test_split

# 이 부분을 10, 100, 500, 1000, 10000 으로 조정하면서 테스트
data_size = 500

# 데이터 생성:  $y = 2x^2 + 3x + 4$ 
np.random.seed(100)
x = np.random.uniform(-10, 10, data_size).reshape(-1, 1)
y = 2 * x**2 + 3 * x + 4

# 학습 데이터와 테스트 데이터를 8:2 비율로 분리
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random

# 이 데이터로 학습을 할 것입니다
print('학습데이터 5개 샘플')
print(x_train[0], y_train[0])
print(x_train[1], y_train[1])
print(x_train[2], y_train[2])
print(x_train[3], y_train[0])
print(x_train[0], y_train[0])

print()

# 학습된 모델을 이용해 이 데이터를 맞출 것입니다
print('테스트데이터 5개 샘플')
print(x_test[0], y_test[0])
print(x_test[1], y_test[1])
print(x_test[2], y_test[2])
print(x_test[3], y_test[3])
print(x_test[4], y_test[4])
```

↕

학습데이터 5개 샘플
[4.08840817] [49.69538716]
[9.03546292] [194.38556923]
[-4.51852506] [31.27856224]
[3.65412913] [49.69538716]
[4.08840817] [49.69538716]

테스트데이터 5개 샘플
[5.56073105] [82.52565269]
[9.33219389] [206.17626726]

오늘 실습할 샘플 코드 소개

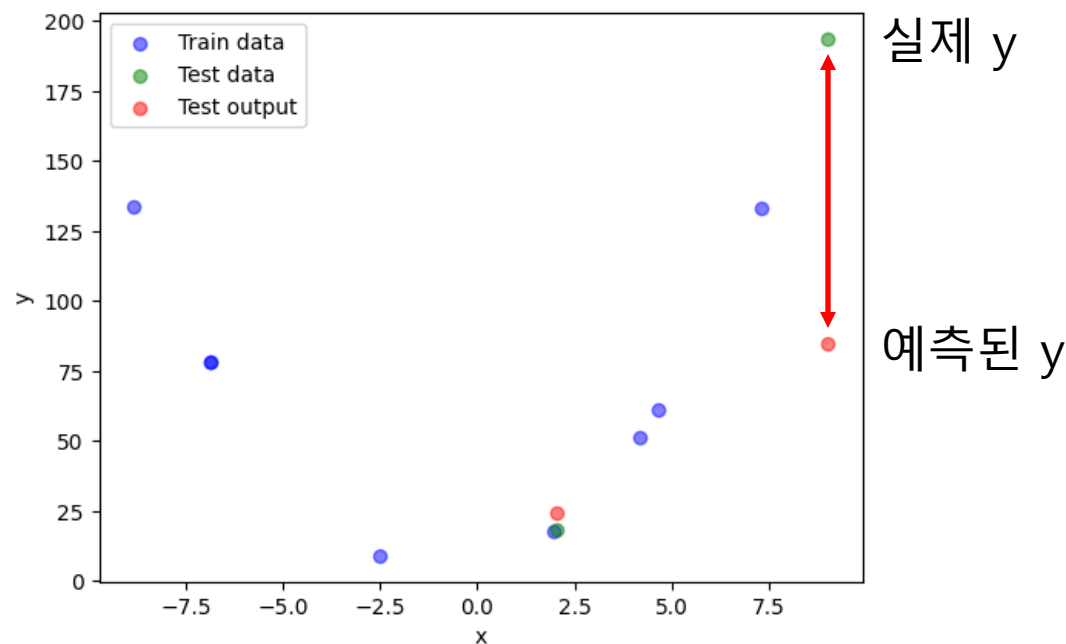
- 아래 식을 만족하는 (x, y) 쌍을 여러 개 만듭니다
- 아래 각 항의 계수는 컴퓨터에게 절대 알려주지 않습니다

$$y = 2x^2 + 3x + 4$$

- (x, y) 쌍의 80%를 이용해서 학습을 시킵니다
- (x, y) 쌍의 20%를 이용해 x 를 대입하여 y 가 맞게 나오는지 확인해봅니다

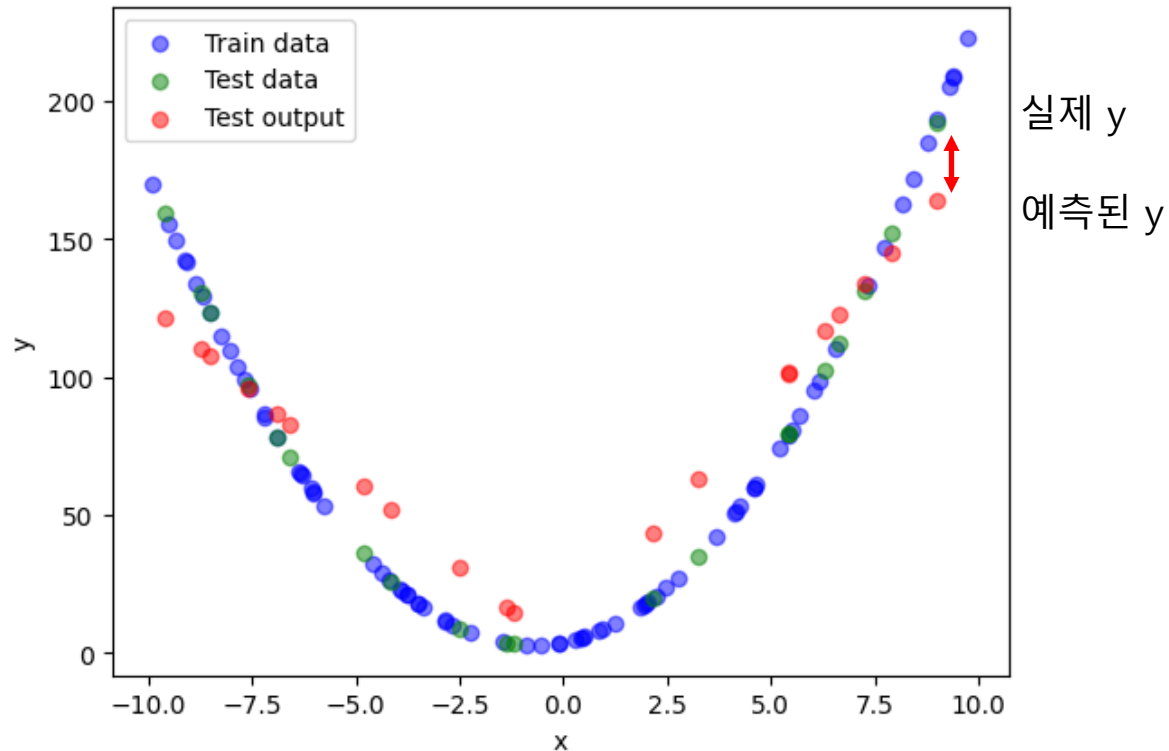
사람이 만드는 함수와 AI가 만드는 모델

- 데이터를 10개 사용했을 때 이차방정식의 계수를 맞추기
- 파란색 점은 학습에 사용된 (x, y) 쌍입니다
- 초록색 점은 테스트에 사용된 (x, y) 쌍입니다
- 붉은색 점은 테스트에 사용된 x 를 모델에 대입했을 때 출력된 y 입니다



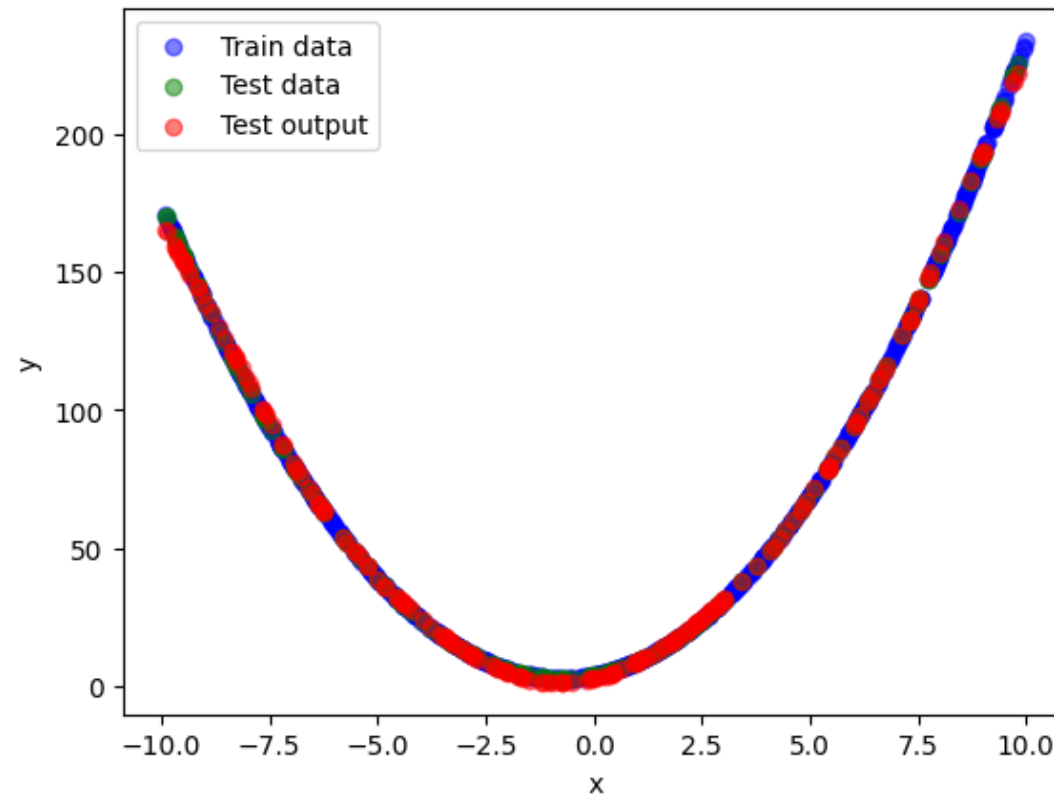
사람이 만드는 함수와 AI가 만드는 모델

- 데이터를 100개 사용했을 때 이차방정식의 계수를 맞추기
- 10개 데이터로 학습했을 때보다 예측 오차가 줄어들었으나 여전히 오차가 있습니다



사람이 만드는 함수와 AI가 만드는 모델

- 데이터를 1000개 사용했을 때 이차방정식의 계수를 맞추기
- 테스트에 사용한 x 로 모델이 연산한 결과와 실제 y 가 거의 일치합니다



AI학습의 효율성

- 우리는 3개의 (x, y) 쌍만 가지고 계수를 정확히 도출할 수 있습니다
- 그러나 AI는 무려 1,000개의 데이터가 필요했습니다
- 연립방정식만 놓고 보면 비효율적입니다
- 단, 연립방정식을 몰라도 계수에 근접한 값을 알아낼 수 있습니다
- 데이터만 충분하면 이차방정식보다 더 복잡한 문제를 근사적으로 풀 수 있다는 것입니다

근사적이라는 말의 의미

- 아래 붉은 점은 이차방정식 계수에 의해 y 를 계산해 낸 것이 아닙니다
- 학습된 데이터를 기반으로 추정했을 때 확률적으로 가장 근사적인 y 가 도출된 것입니다

