

프로그래밍_인공지능2

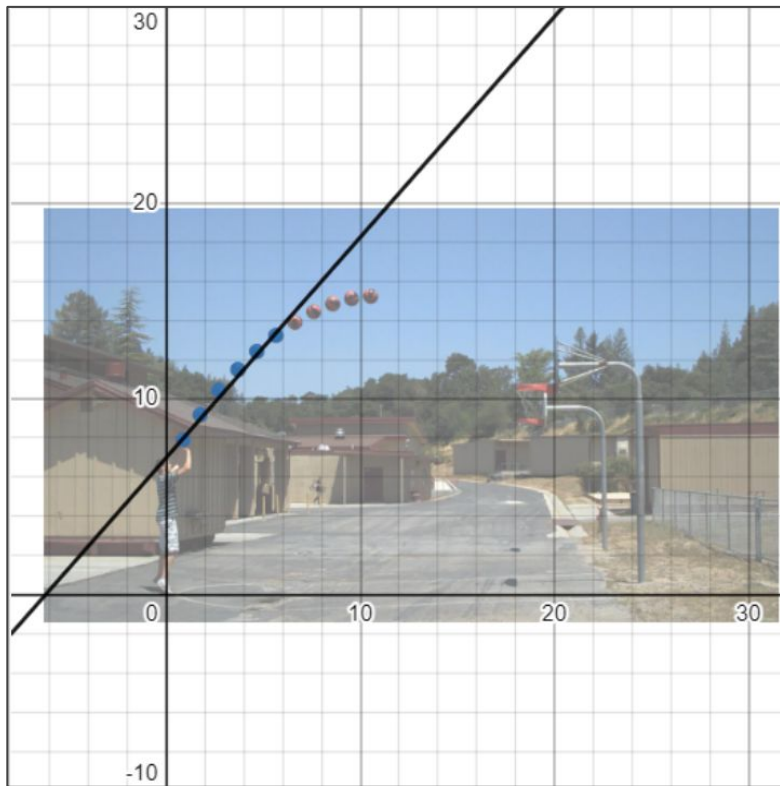
2022학년도 2학기

내가 던진 자유투가
성공할까?

선형(직선)으로 예측이 가능할까?

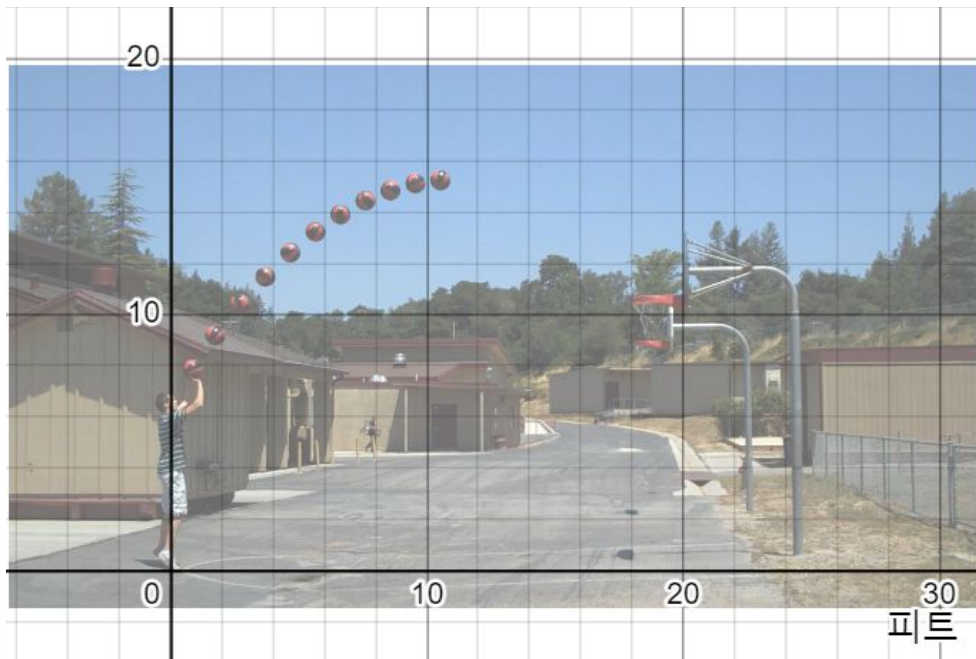
- 같이 해봅시다!

> Will It Hit the Hoop?



TF basic1_실습 에서 만든 모델에 적용해보기

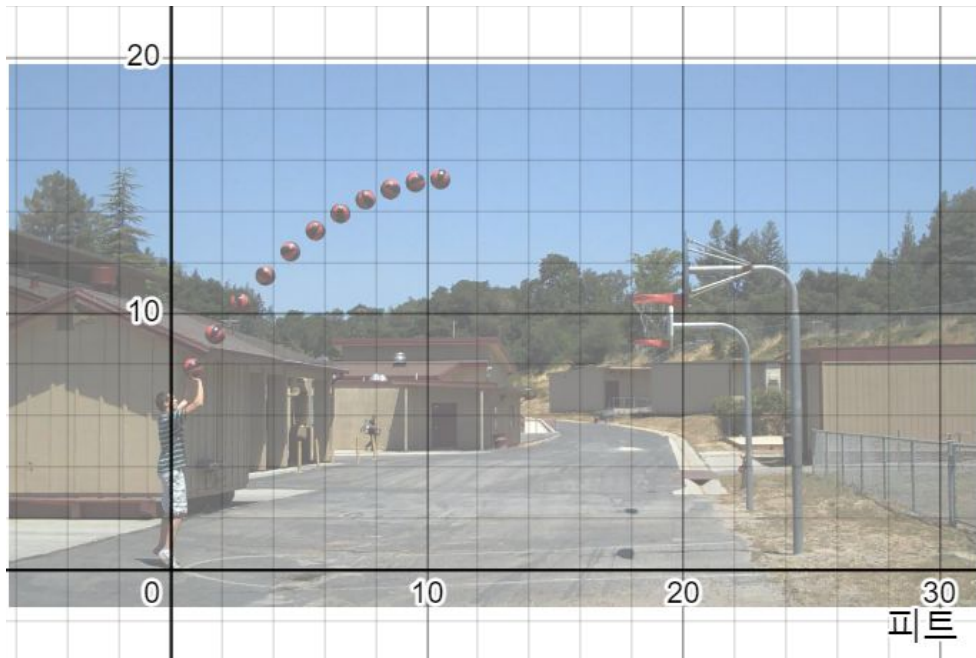
- 옆에 그림을 참고하여 x s와 y s의 값을 입력해보세요!



TF basic1_실습 에서 만든 모델에 적용해보기

- 옆에 그림을 참고하여 x s와 y s의 값을 입력해보세요!

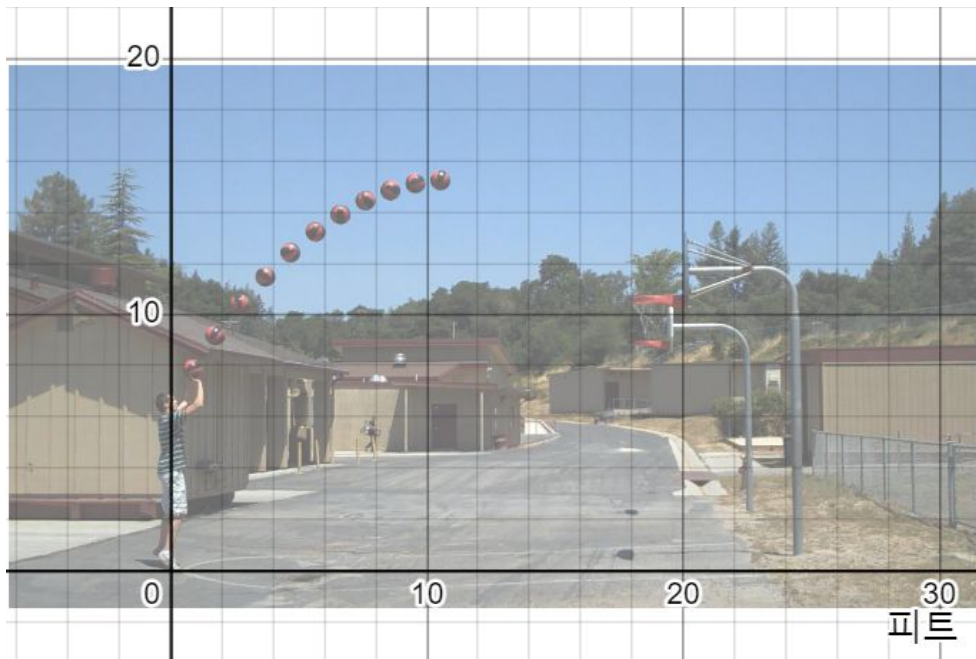
? x 가 15일 때, 올바른 값을 예측하나요?



TF basic1_실습 에서 만든 모델에 적용해보기

? x 가 15일 때, 올바른 값을 예측하나요?

- 예측하지 못한다면, 어떻게 바꿔볼 수 있을까요?



**농구의 자유투에 성공하기 위한 요소를 분석하여
이를 해결하기 위한 방법을 찾고,
알고리즘을 설계할 수 있다.**

[자료수집1] 자유투에 성공하기 위한 여러 요소 조사하기



예시: <https://www.youtube.com/watch?v=tkD1fDZrWzg&t=140s>

- 작성 시, 요소+이유로 정리하기

[설계1] 자유투에 성공하기 위한 요소에 따른 가설 설정하기

1. 종속변수: 자유투 성공 여부
2. 독립변수: 자유투에 성공하기 위한 요소

3. 조건: 각자 상황에 맞도록 설정하기

- 2차원 예) [Will It Hit the Hoop?](#)
- 3차원 예) <https://vlabon.re.kr/contents/basket4>

* 위의 두 사이트에서 자료수집하기

* 자유투 성공 여부는 실패 0, 성공 1로 하거나 성공 확률을 알 수 있다면 확률로 표시해도 된다!

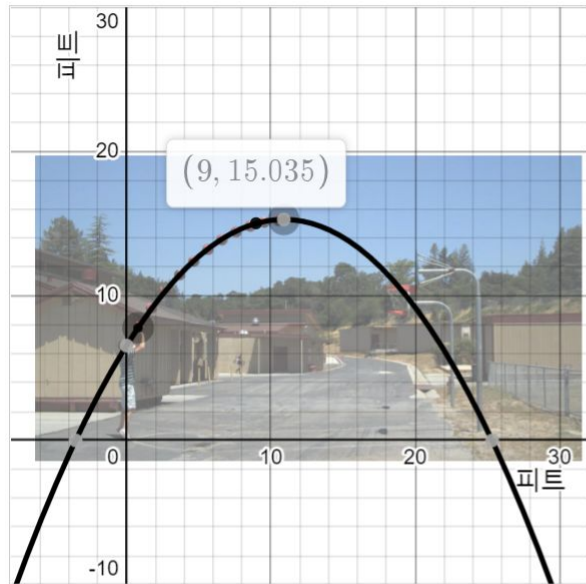
[자료수집2] 데이터 수집하기

2차원의 경우, 마우스로 꾸욱 누르면 좌표를 알려준다.

7번의 시도를 데이터로 삼아,

어떤 독립변인들이 골 성공여부를 결정했는지 생각해보고 데이터를 스프레드시트에 정리한다.

1번째 샷

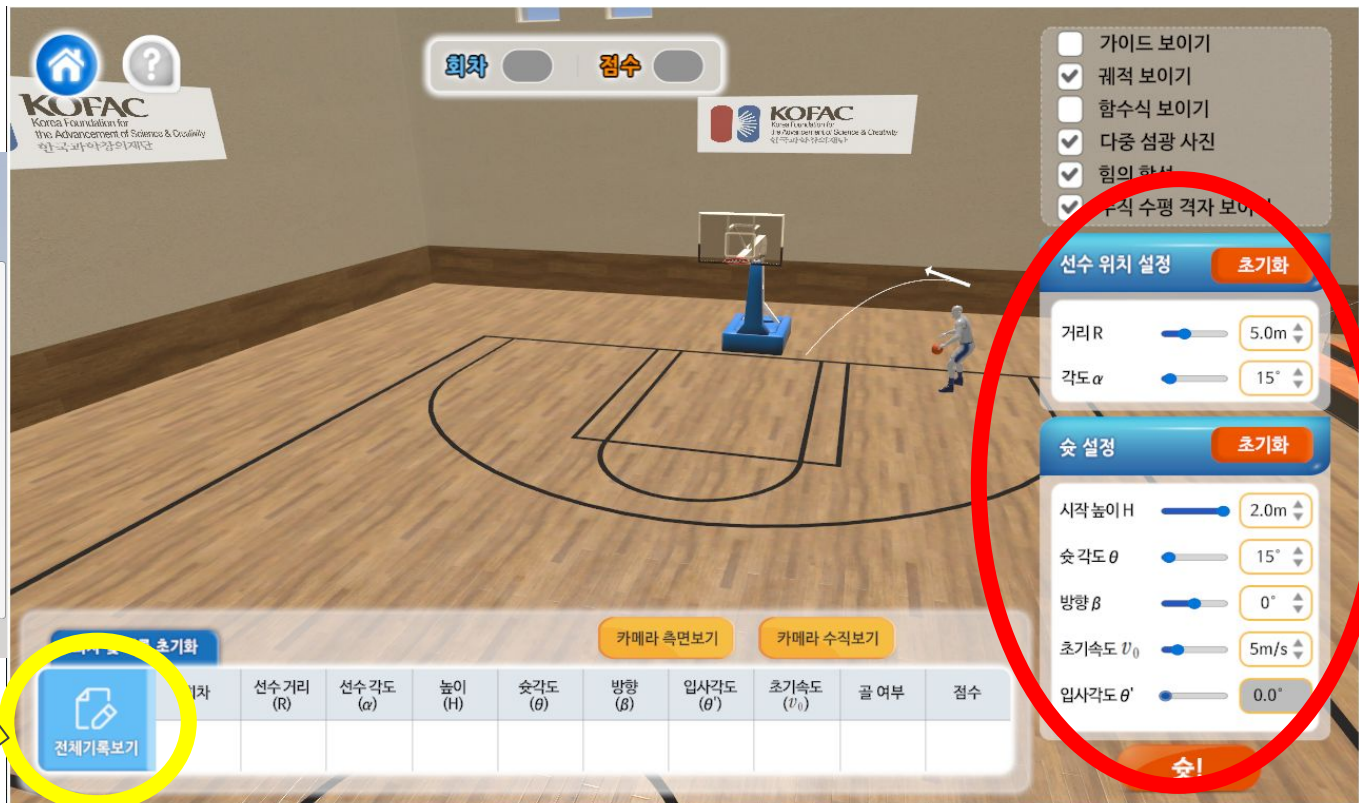


[자료수집2] 데이터 수집하기

3차원의 경우,

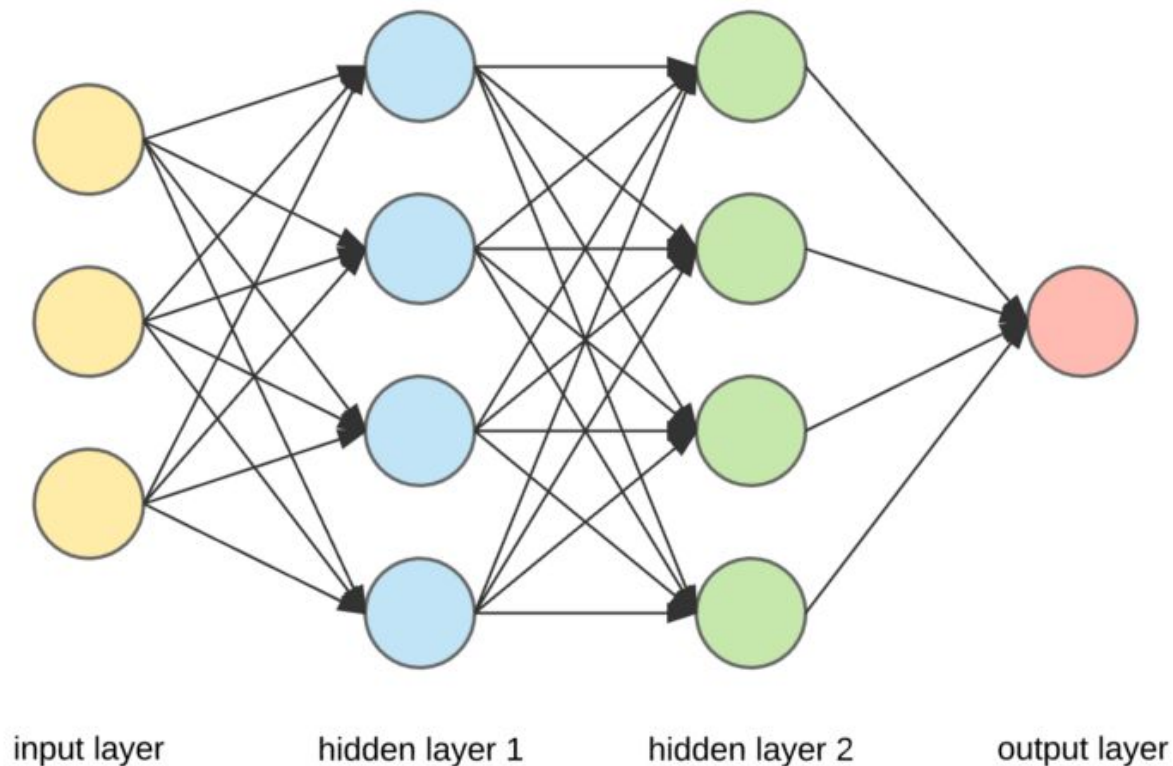
<input checked="" type="checkbox"/> 높이 (H)	<input checked="" type="checkbox"/> 슛각도 (θ)	<input checked="" type="checkbox"/> 방향 (β)	<input checked="" type="checkbox"/> 입사각도 (θ')	<input checked="" type="checkbox"/> 초기속도 (v_0)		
스�각도 (θ)	방향 (β)	입사각도 (θ')	초기속도 (v_0)	골 여부	점수	누적 점수

선택 삭제 초기화 기록 다운로드



인공지능 모델 설계하기

고차원적인 생각 필요 = 모델의 층이 깊어야 함



1. 라이브러리 가지고 오기

```
import tensorflow as tf
import numpy as np
import pandas as pd #자료를 표의 형태(csv, 엑셀파일)로 가져올 수도
있어서 추가함
from tensorflow import keras
from tensorflow import layers #층을 여러개 쌓으려고 가지고 올
from pandas import read_csv #csv 파일 처리를 위함(API나 다운로드
받는 자료들이 대부분 csv 파일로 되어있을 것)
```

2. 독립변수와 종속변수 반영_ 방법1

- 지난 시간에 했던 것처럼 직접 입력하기

```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype = float)
```

```
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype = float)
```

2. 독립변수와 종속변수 반영_ 방법2

- csv 파일을 드라이브에 업로드하고, 파일에 있는 내용을 가져와 사용하기

1. 구글 드라이브 연결하기

```
from google.colab import drive
drive.mount('/content/drive')
```

2. csv 파일 읽어오기

```
자료 = pd.read_csv('/content/drive/ #파일경로')
```

```
열_이름_독립변수 = 자료[ [#파일 안을 확인해서 독립변수에 해당하는 열이름 적어주기]]
```

```
열_이름_종속변수 = 자료[ [#파일 안을 확인해서 종속변수에 해당하는 열이름 적어주기]]
```

예) 독립변수 = 자료['나이', '성별']

종속변수 = 자료['키']

3. 확인하기

```
print(열_이름_독립변수.shape, 열_이름_종속변수.shape)
```


2. 독립변수와 종속변수 반영_ 방법3

- github 파일 이용하기

파일경로 =

'https://raw.githubusercontent.com/blackdew/tensorflow1/master/csv/boston.csv'

보스턴 = pd.read_csv(파일경로)

3. 모델 생성하기_방법1

```
model = tf.keras.models.Sequential([  
    tf.keras.layers.Flatten(input_shape=(28, 28)),  
    tf.keras.layers.Dense(128, activation='relu'),  
    tf.keras.layers.Dropout(0.2),  
    tf.keras.layers.Dense(10)  
])
```

3. 모델 생성하기_방법2

```
model = tf.keras.Sequential([  
# 64개의 유닛을 가진 완전 연결 층을 모델에 추가합니다:  
layers.Dense(64, activation='relu', input_shape=(32,)),  
# 또 하나를 추가합니다:  
layers.Dense(64, activation='relu'),  
# 10개의 출력 유닛을 가진 소프트맥스 층을 추가합니다:  
layers.Dense(10, activation='softmax')])
```

3. 모델 생성하기_ 방법3

```
X = tf.keras.layers.Input(shape=[13])
H = tf.keras.layers.Dense(8, activation='swish')(X)
H = tf.keras.layers.Dense(8, activation='swish')(H)
H = tf.keras.layers.Dense(8, activation='swish')(H)
Y = tf.keras.layers.Dense(1)(H)
model = tf.keras.models.Model(X, Y)
```

4. 모델 컴파일러 정해주기

1. `model.compile(optimizer='sgd', loss='mse')`
2. `model.compile(loss='categorical_crossentropy',
metrics='accuracy')`
3. `model.compile(loss='mse')`
4. `model.compile(optimizer='adam',
loss='mse',
metrics=['accuracy'])`
5. `model.compile(optimizer=tf.keras.optimizers.Adam(),
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])`