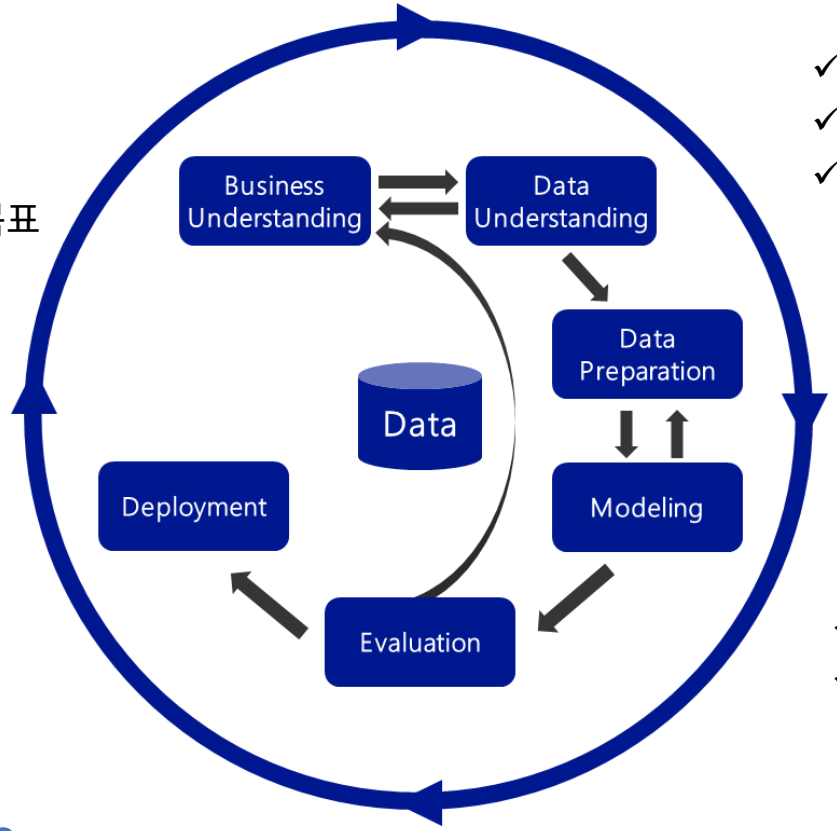
1일차 정리

전체 Process(CRISP-DM)

무엇이 문제인가?

✓ 비즈니스 **문제정의**
✓ 데이터분석 방향, 목표
✓ 초기 **가설 수립**

$$x \rightarrow y$$



✓ 원본식별
✓ **분석을 위한 구조 만들기**
✓ **데이터분석 EDA & CDA**

✓ 모델링을 위한
데이터 구조 만들기
- 모든 셀은 **값**이 있어야 한다.
- 모든 값은 **숫자** 여야 한다.
- (필요 시) 숫자의 **범위**가 일치

✓ 모델을 만들고
✓ 검증하기

✓ 모델 관리
✓ AI 서비스 구축

문제가 해결 되었는가?

✓ 기술적 관점 평가
✓ 비즈니스 관점 평가

알고리즘 한판 정리

	선형회귀	로지스틱회귀	KNN	SVM	Decision Tree	Random Forest	Gradient Boost (GBM, XGB, LGBM)
개념	✓오차를 최소화 하는 직선, 평면	✓오차를 최소화 하는 직선, 평면 ✓직선을 로지스틱 함수로 변환 (0~1 사이 값으로)	✓예측할 데이터와 train set과의 거리 계산 ✓가까운 [k개 이웃의 y]의 평균으로 예측	✓마진을 최대화 하는 초평면 찾기 ✓데이터 커널 변환	✓정보전달량 = 부모 불순도 - 자식 불순도 ✓정보 전달량이 가장 큰 변수를 기준으로 split	✓여러 개의 트리 ✓각각 예측 값의 평균 ✓행과 열에 대한 랜덤 : 조금씩 다른 트리들 생성	✓여러 개의 트리 ✓트리를 더해서 하나의 모델로 생성 ✓더해지는 트리는 오차를 줄이는 모델
전제 조건	✓NaN조치 ✓가변수화 ✓x들 간 독립	✓NaN조치 ✓가변수화 ✓x들 간 독립	✓NaN조치 ✓가변수화 ✓스케일링	✓NaN조치 ✓가변수화 ✓스케일링	✓NaN조치 ✓가변수화	✓NaN조치 ✓가변수화	✓NaN조치 ✓가변수화
성능	✓변수 선택 중요 ✓x가 많을 수록 복잡	✓변수 선택 중요 ✓x가 많을 수록 복잡	✓주요 hyper-parameter - n_neighbors : k 작을수록 복잡 - metric : 거리계산법	✓주요 hyper-parameter - C : 클수록 복잡 - gamma : 클수록 복잡	✓주요 hp - max_depth : 클수록 복잡 - min_samples_leaf : 작을수록 복잡	✓주요 hp 기본값으로도 충분! - n_estimators - max_features ✓기본값으로 생성된 모델 ==> 과적합 회피	✓주요 hp - n_estimators - learning_rate ✓XGB, LGBM : 과적합 회피를 위한 규제

회귀모델 평가

오차의 크기

$\hat{y}$: 예측값

오차 | 제곱 오차 | 절대값 오차 | 오차율

y	$\hat{y}$	$y-\hat{y}$	$(y-\hat{y})^2$	$\lvert y-\hat{y}\rvert$	$\left\lvert\frac{y-\hat{y}}{y}\right\rvert$
6	4				
5	6				
12	9				
2	2				

평균

MSE
RMSE

MAE

MAPE

평균 오차

평균 오차율

딥러닝 개념 - 학습 절차

✓ model.fit(x_train, y_train) 하는 순간…

단계① : 가중치에 (초기)값을 할당한다.
- *초기값은 랜덤으로 지정*

$$medv = 1 \cdot lstat + 3$$

$$\boldsymbol{medv = w_1 \cdot lstat + w_0}$$

단계② : (예측) 결과를 뽑는다.

medv	lstat	$\hat{y}$
20	10	13
10	11	14
8	15	18

단계③ : 오차를 계산한다.

$$mse = \frac{\sum(y - \hat{y})^2}{n} = \frac{7^2 + 6^2 + 8^2}{3}$$

forward propagation

단계④ : 오차를 줄이는 방향으로 가중치를 조정
- *Optimizer : GD, Adam…*

$$\boldsymbol{w_1 : 1 \rightarrow 0.8}$$
$$\boldsymbol{w_0 : 3 \rightarrow 3.3}$$

back propagation

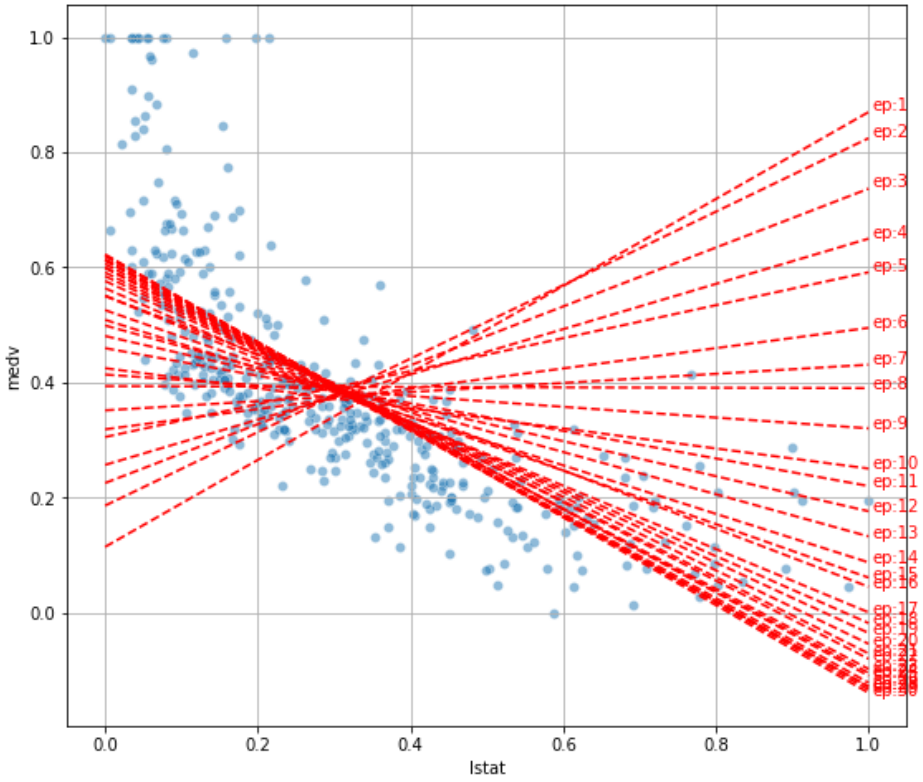
단계⑤ : 다시 단계①로 올라가 반복한다.
- *max iteration에 도달.(오차의 변동이 (거의) 없으면 끝.)*

- 가중치(weight)의 다른 용어 **파라미터(parameter)**

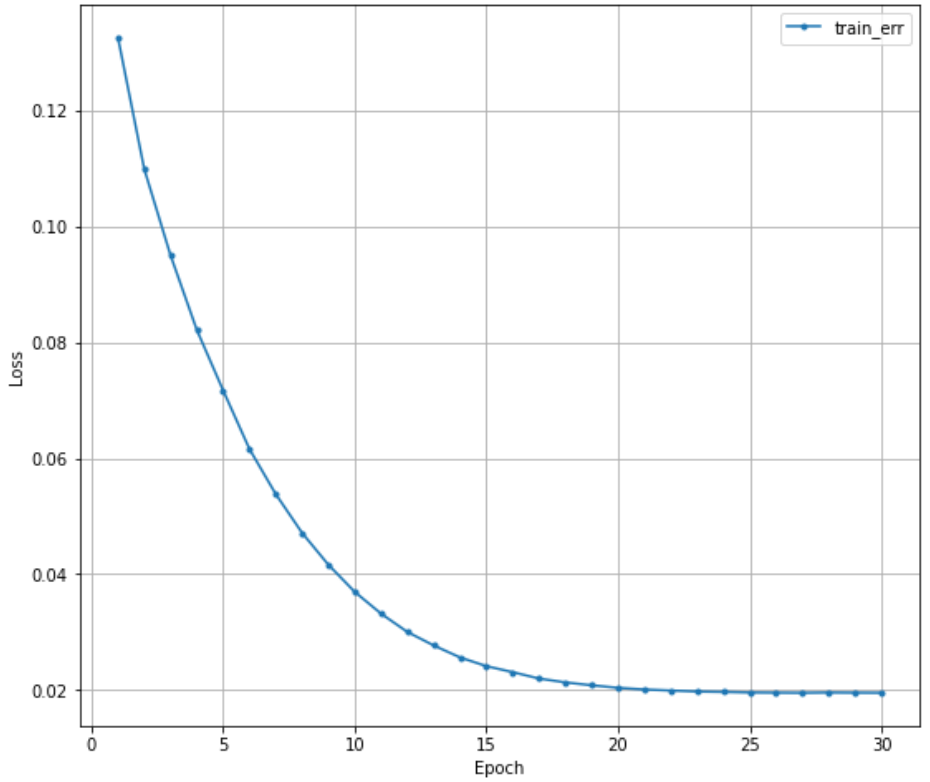
딥러닝 개념 - 학습 절차

✓30번 조정하며 최적의 Weight를 찾아가는 과정

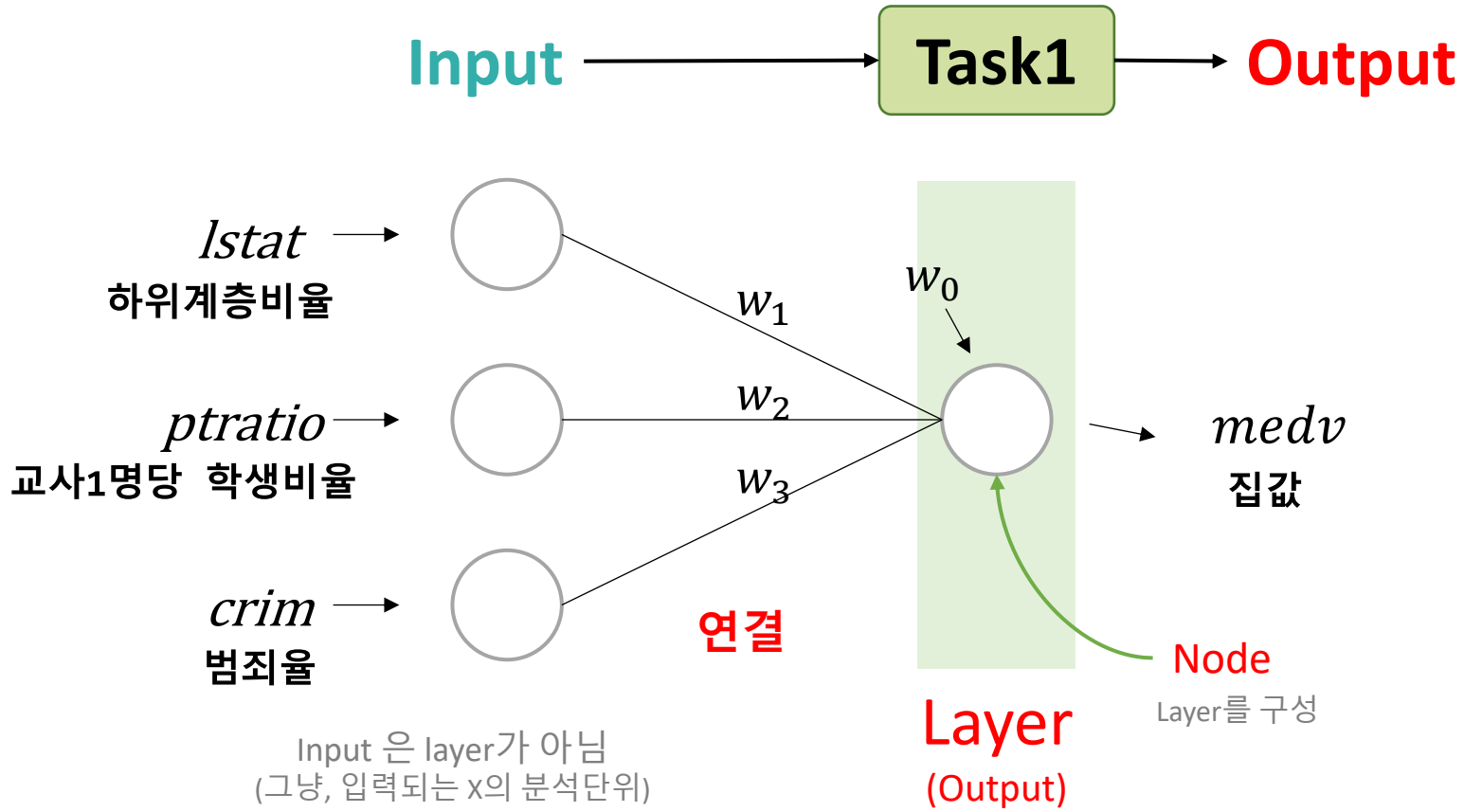
모델의 가중치가 업데이트되는 과정



모델의 오차가 줄어드는 과정



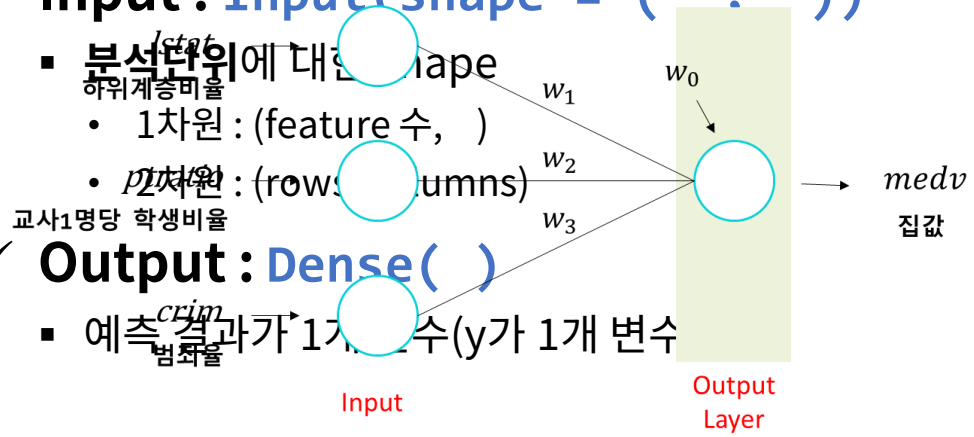
딥러닝 구조



$$medv = w_1 \cdot lstat + w_2 \cdot ptratio + w_3 \cdot crim + w_0$$

딥러닝 코드 - Dense

- ✓ **Input : Input(shape = (,))**
  - **분석단위**에 대한 shape
    - 1차원 : (feature 수,)
    - 2차원 : (rows, columns)
- ✓ **Output : Dense( )**
  - 예측 결과가 1개 수(y가 1개 변수




```
# 메모리 정리
clear_session()

# Sequential 타입                                    3
model = Sequential([Input(shape = (nfeatures,)),
                    Dense(1) ])
# 모델요약                                          input
model.summary()                          output
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1)	4

Total params: 4 (16.00 B)
Trainable params: 4 (16.00 B)
Non-trainable params: 0 (0.00 B)

잠깐만요

위 모델 코드의 Input 함수를 첫 번째 Layer(Dense) 안에 옵션으로 포함 가능

```
Model = Sequential([Dense(1, input_shape=(3,)) ])
```

Compile

✓컴파일(Compile)

- 선언된 모델에 대해 몇 가지 설정을 한 후
- 컴퓨터가 이해할 수 있는 형태로 변환하는 작업

```
model.compile( optimizer = Adam(learning_rate = 0.1)
             , loss='mse')
```

✓loss function(오차함수)

- Cost Function, Objective Function 과 같은(유사한) 의미
- 오차 계산을 무엇으로 할지 결정
- mse : mean squared error
  - 회귀모델 : mse
  - 분류모델 : cross entropy

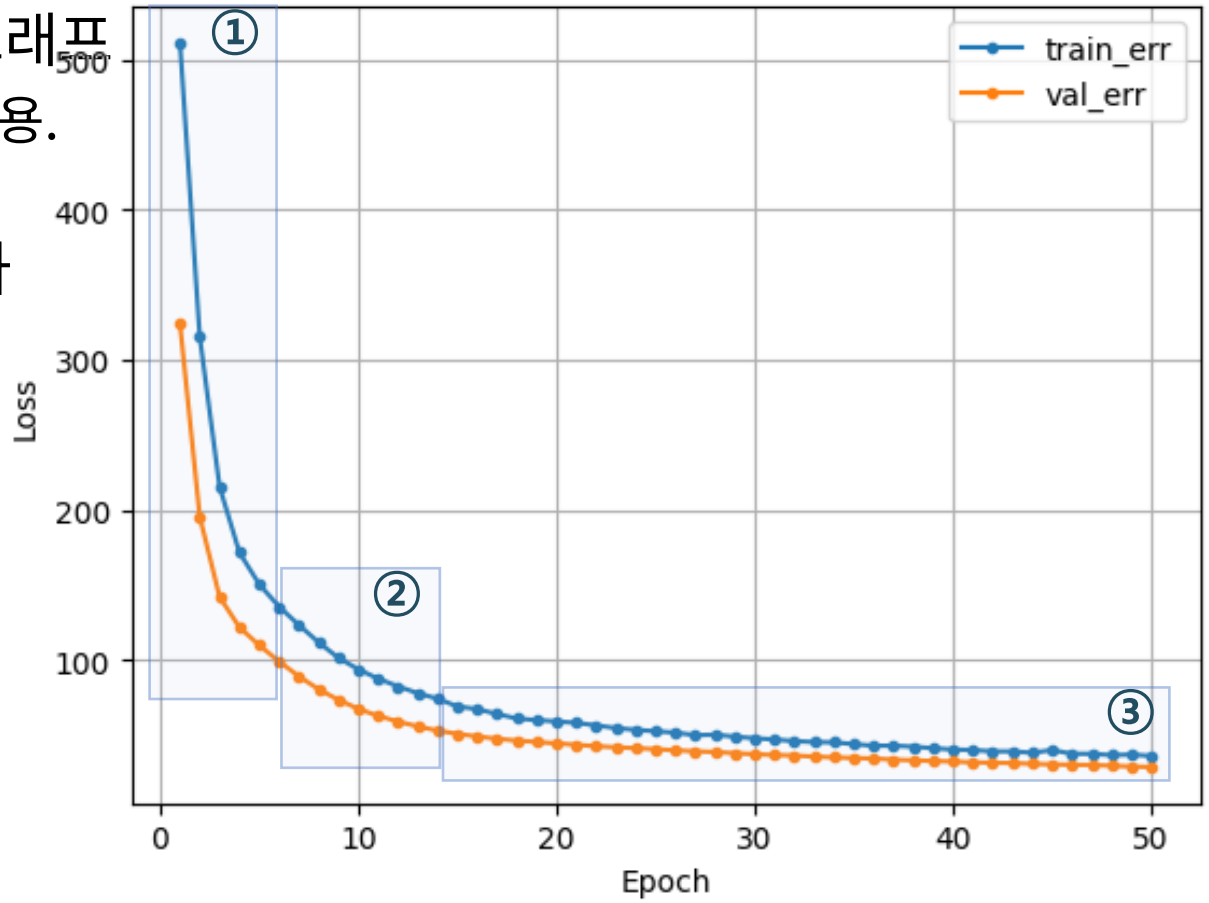
학습 곡선

✓학습 곡선이란

- **모델 학습이 잘 되었는지 파악**하기 위한 그래프
  - 정답은 아니지만, **학습 경향을 파악**하는데 유용.

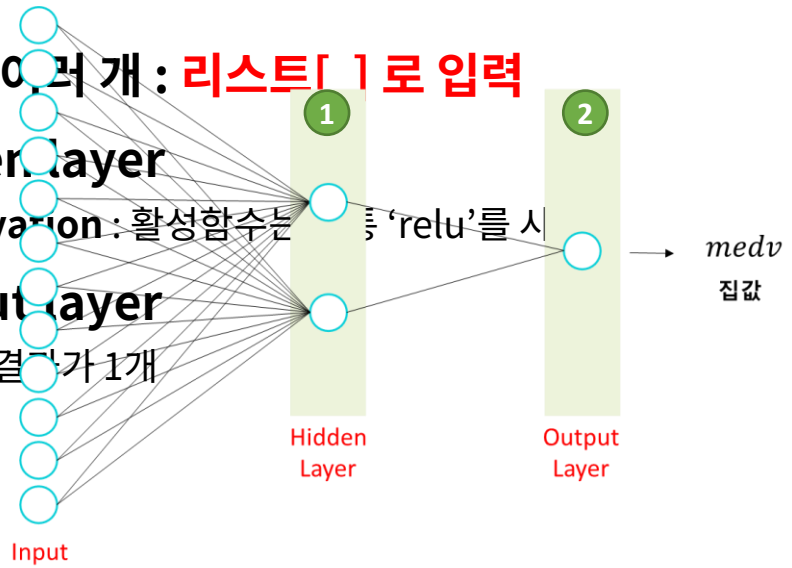
- 각 Epoch 마다 train error와 val error가 어떻게 줄어들고 있는지 확인
  - Epoch = 10 : train data를 10번 반복 학습

- 바람직한 학습 곡선
  - ① 초기 epoch에서는 오차가 크게 줄고
  - ② 오차 하락이 꺾이면서
  - ③ 점차 완만해짐
  - 그러나 학습곡선의 모양새는 다양함



딥러닝 구조 - Hidden Layer

- ✓ layer 여러 개 : 리스트[ ] 로 입력
- ✓ hidden layer
  - Activation : 활성함수는 'relu'를 사
- ✓ output layer
  - 예측 결과가 1개




```
# Sequential 타입 모델 선언(입력은 리스트로!)
model3 = Sequential([Input(shape = (nfeatures,)),
                     Dense(2, activation = 'relu'),
                     Dense(1)   ])
```

1

2

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 2)                 26

 dense_1 (Dense)             (None, 1)                 3

=================================================================
Total params: 29
Trainable params: 29
Non-trainable params: 0
_________________________________________________________________
```

활성화 함수 Activation Function

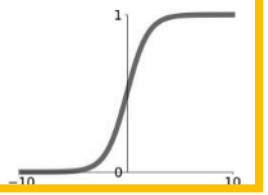
✓그래서 활성화 함수는…

- Hidden Layer에서는 : 선형함수를 비선형 함수로 변환
- Output Layer에서는 : 결과값을 다른 값으로 변환해 주는 역할
  - 주로 분류Classification 모델에서 필요

Hidden layer
국룰

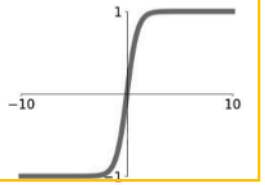
Sigmoid

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



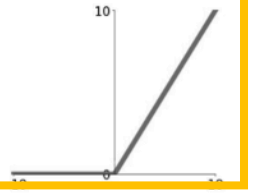
tanh

$$\tanh(x)$$



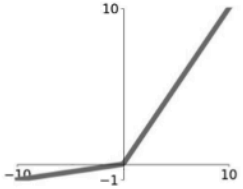
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

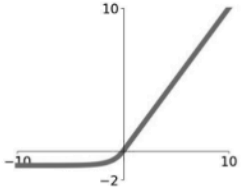


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



요약 : 회귀 모델링

✓ 딥러닝 전처리

- NaN 조치, 가변수화, 스케일링

✓ Layer

- 첫번째 Layer는 input_shape를 받는다.(분석단위의 shape)
  - 2차원 데이터셋의 분석단위 1차원 → shape는 (feature수,)
- Output layer의 node 수 : 1
- Activation Function
  - Hidden layer에 필요 :
    - 비선형 모델로 만들려고 ➔ hidden layer를 여럿 쌓아서 성능을 높이려고.
  - 회귀 모델링에서 Output Layer에는 활성화 함수 필요하지 않음!

구분	Hidden Layer	Output Layer		Compile	
	Activation	Activation	Node수	optimizer	loss
Regression	relu	X	1	adam	mse