

Category 2

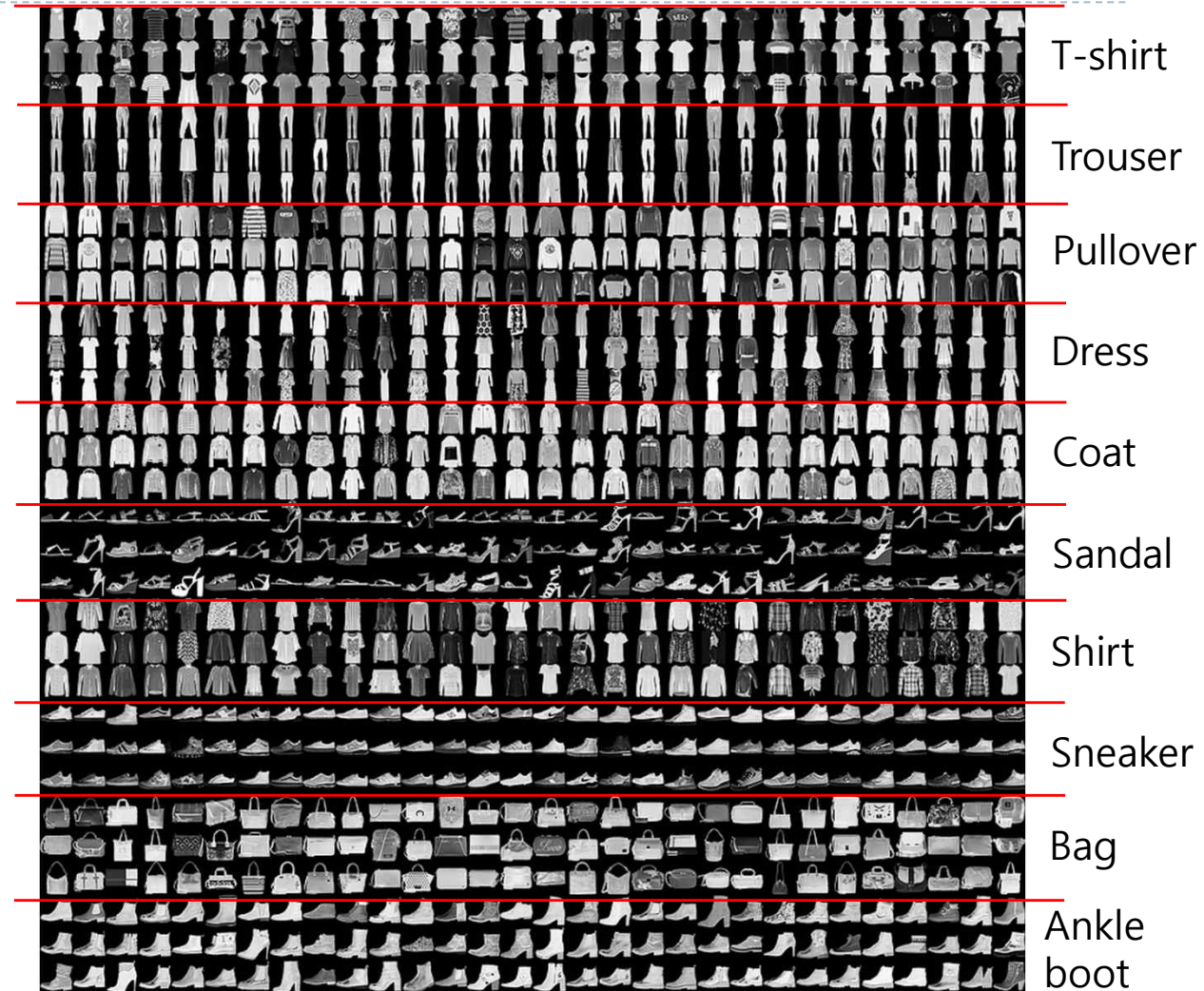
DNN(Deep Neural Net)



Fashion MNIST 데이터셋 분석

Fashion MNIST

“ Fashion-MNIST는 60,000 개의 예제로 구성된 training set 와 10,000 개의 예제로 구성된 test set 로 구성된 Zalando의 article images 데이터 세트, 10 개 클래스의 레이블과 연결된 28x28 grayscale 이미지이다



Fashion MNIST

“ 10 개 클래스 레이블

<https://www.tensorflow.org/tutorials/keras/classification?hl=ko>

Labels

Each training and test example is assigned to one of the following labels:

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot



기본 분류모델

Fashion MNIST

정규화(Normalization)

“ 전처리(Preprocessing)

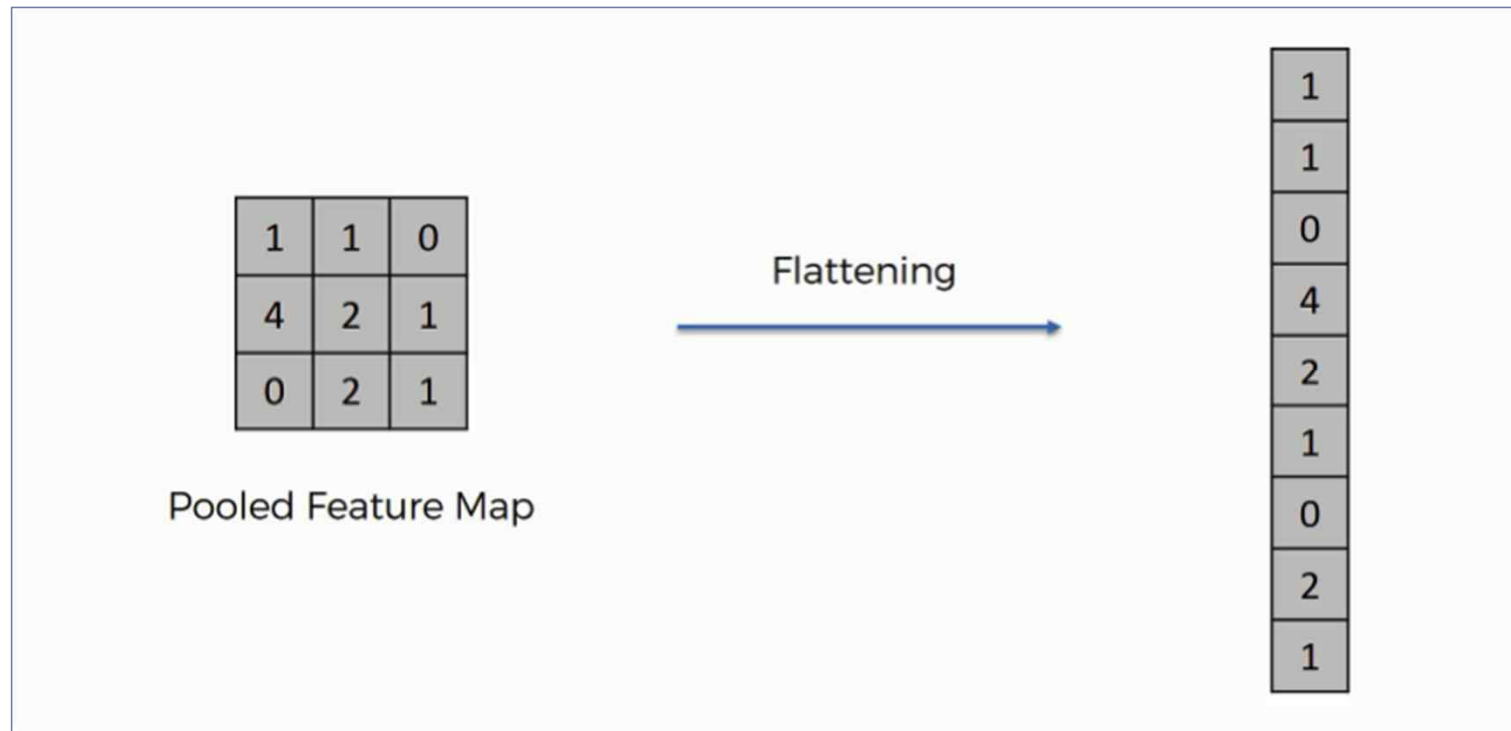
원본 raw 데이터의 품질이 좋지 않아서 가공하여 데이터의 품질을 향상시키는 과정

“ 피처 스케일링

- (1) **표준화(Standardization)** : 데이터가 평균으로 부터 얼마나 떨어져 있는지를 나타내는 값으로, 특정 범위를 벗어난 데이터를 이상치(outlier)로 간주하여 제거 한다, StandardScaler 사용, $Z = (X - \text{mean}()) / \text{std}()$, 정규분포, 가우시안 분포(평균=0, 분산=1)
- (2) **정규화(Normalization)** : 데이터의 범위를 일치시키거나 분포를 유사하게 만들어 주는 기능, 데이터의 범위를 0 ~ 1사이의 값으로 변환, MinMaxScaler 사용, $Z = (X - \text{min}()) / (\text{max}() - \text{min}())$

Flatten Layer

“ Flatten Layer는 2차원 데이터를 1차원으로 변경한다



Flatten Layer

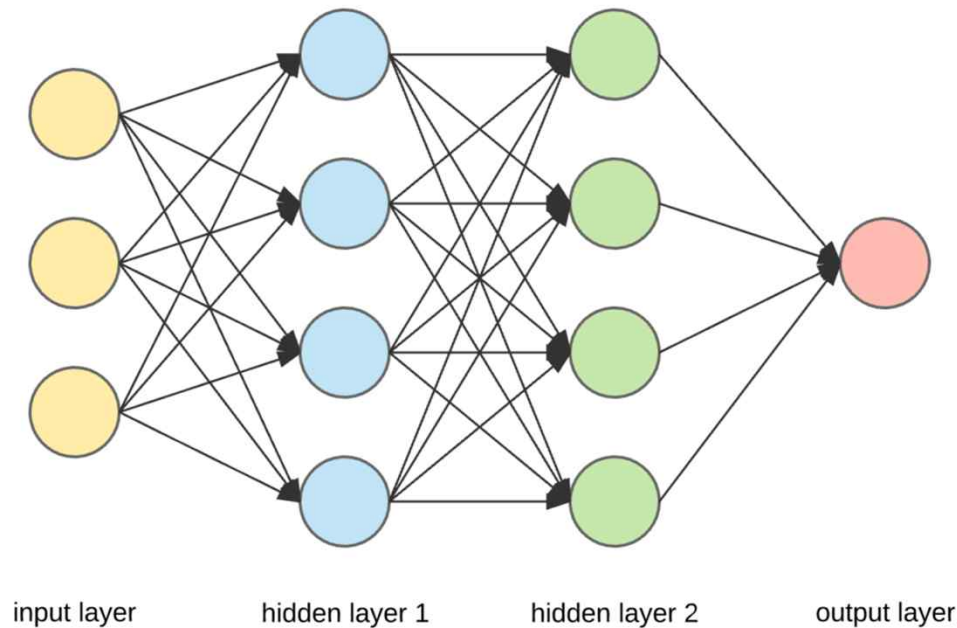
“ 실제 텐서플로 소스 코드에서 입력데이터는 batch size가 N개인 데이터를 입력으로 사용하기 때문에 Flatten Layer는 다차원 배열을 2차원으로 변환해주는 결과가 된다

(28,28) → (28,28) [2차원은 그대로 2차원으로]
(None,28,28) → (None,28*28) [3차원은 2차원으로 출력]
(None,28,28,10) → (None,28*28*10) [4차원은 2차원으로 출력]

None은 batch size, 입력 데이터의 인스턴스 갯수

Dense Layer

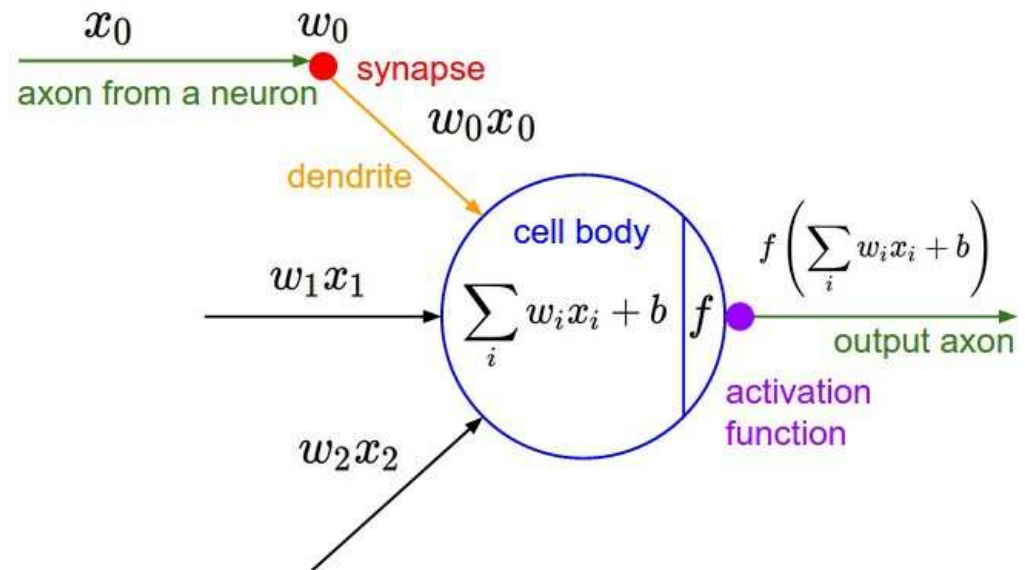
“ FC(Fully Connected) Layer 또는 Dense Layer는 input과 Output을 모두 연결해주는 완전 연결 계층으로, 모든 뉴런이 빠짐없이 입력과 출력에 연결된다



Activation Functions

- “ 활성화 함수란 어떠한 신호를 입력받아 이를 적절한 처리를 하여 출력해주는 함수이다. 이를 통해 출력된 신호가 다음 단계에서 활성화 되는지를 결정한다

input data -> {activation function} -> output data

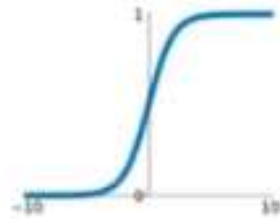


Activation Functions

Activation Functions

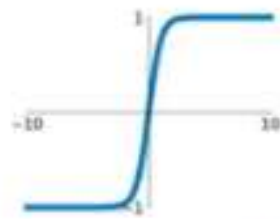
Sigmoid

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



tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$



sigmoid함수는 $0 < n < 1$ 값을 계속 곱하여 나가므로 다층 신경망에서는 0으로 수렴할 수 있다

1보다 작아지지 않기 위한 대안으로 ReLU(Rectified Linear Unite) 함수를 도입하게 되었다

hidden layer에는 ReLU를 주로 사용한다

compile() 메서드

- “ compile() 메서드는 학습을 위한 모델의 인자를 설정한다
- “ 다음 세 개의 인자를 입력으로 받습니다.
 - **옵티마이저(optimizer)**
 - 훈련과정을 설정. 즉, 최적화 알고리즘을 설정을 의미
 - adam, sgd, rmsprop, adagrad 등이 있다.
 - **손실 함수 (loss function)**
 - 모델이 최적화에 사용되는 목적 함수입니다.
 - mse, categorical_crossentropy, binary_crossentropy 등이 있다.
 - **평가지표 (metrics)**
 - 훈련을 모니터링 하기 위해 사용된다.
 - 분류에서는 accuracy, 회귀에서는 mse, rmse, r2, mae, mspe, mape, msle 등이 있다.
 - 사용자가 메트릭을 정의해서 사용할 수도 있다.

Adam optimizer

“ 최적화 알고리즘 중에서 매우 뛰어난 성능

<https://dalpo0814.tistory.com/29>

sparse_categorical_crossentropy

categorical_crossentropy : 다중 분류 loss, One-hot encoding 된 클래스, [0.2, 0.3, 0.5] 와 같은 출력 값과 실측 값의 오차 값을 계산한다.

sparse_categorical_crossentropy : 다중 분류 loss, Y값이 One-hot encoding 처리 안된 정수 타입인 경우에 사용

sparse_categorical_crossentropy로 설정할 경우에는 학습 시 Y값을 One-hot encoding 하지 않고 그대로 사용한다
예측 값은 전체 클래스의 확률 값들이 모두 출력 된다

evaluate() 메서드

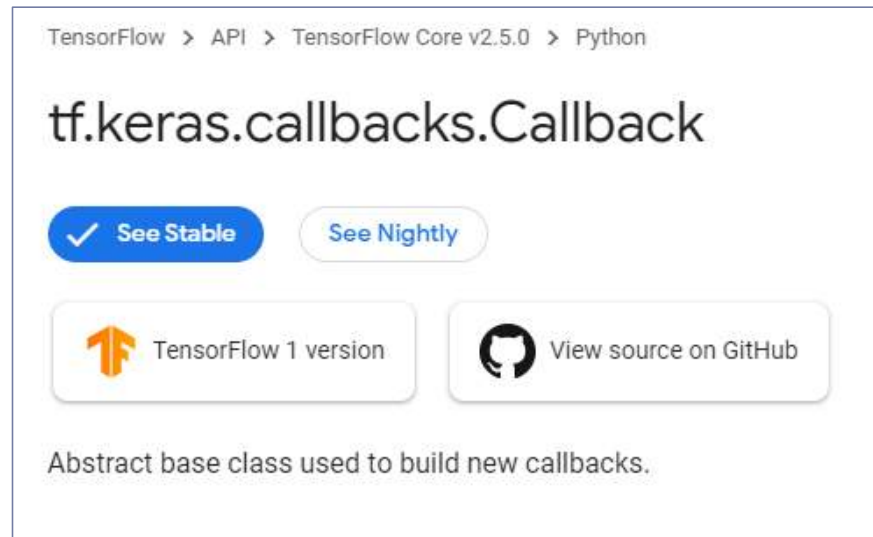
“ Test 모드에서 모델에 대한 손실(loss) 값 및 메트릭(metric) 값을 반환한다

evaluate() 실행 전에 fit()으로 미리 학습을 수행하여야 한다

evaluate()함수의 인자는 학습에 사용되지 않은 X와 Y값을 입력해준다

Callback 클래스 사용

- “ 콜백(Callback) : 모델 학습 중 특정 지점에서 호출되는 메서드 구현
https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/Callback



Callback 클래스 사용

- “ 1 epoch 학습 완료 시 마다 on_epoch_end()메서드가 호출된다
학습 결과값을 조건식으로 검사하고 일치하면 stop_trainig을
True로 설정한다 fit()함수는 model 의 stop_training이 True로 되어
있을 경우 학습을 중단시킨다

```
import tensorflow as tf

class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy')>0.8):
            print("\nReached 80% accuracy so cancelling training!")
            self.model.stop_training = True
```

Callback 클래스 사용

- “ tf.keras.callbacks.Callback 클래스를 상속받아서 만든 사용자 Callback 클래스의 인스턴스를 만들어 fit() 메서드의 callbacks 인자에 넣어 설정해주어야 동작한다

```
callbacks = myCallback()
```

```
model.fit(x_train, y_train, epochs=10, callbacks=[callbacks])
```

Callback 클래스의 다양한 메서드들

- “ on_epoch_begin : 각 epoch 시작시 마다 호출됨
- “ **on_epoch_end : 각 epoch 종료시 마다 호출됨**
- “ on_train_begin : Called at the beginning of training.
- “ on_train_end : Called at the end of training.
- “ on_batch_begin : Called at the beginning of a batch in training time
- “ on_batch_end : Called at the end of a batch in training time
- “ on_predict_batch_begin : Called at the beginning of a batch in predict methods.
- “ on_predict_batch_end : Called at the end of a batch in predict methods.

TEST & VALIDATION

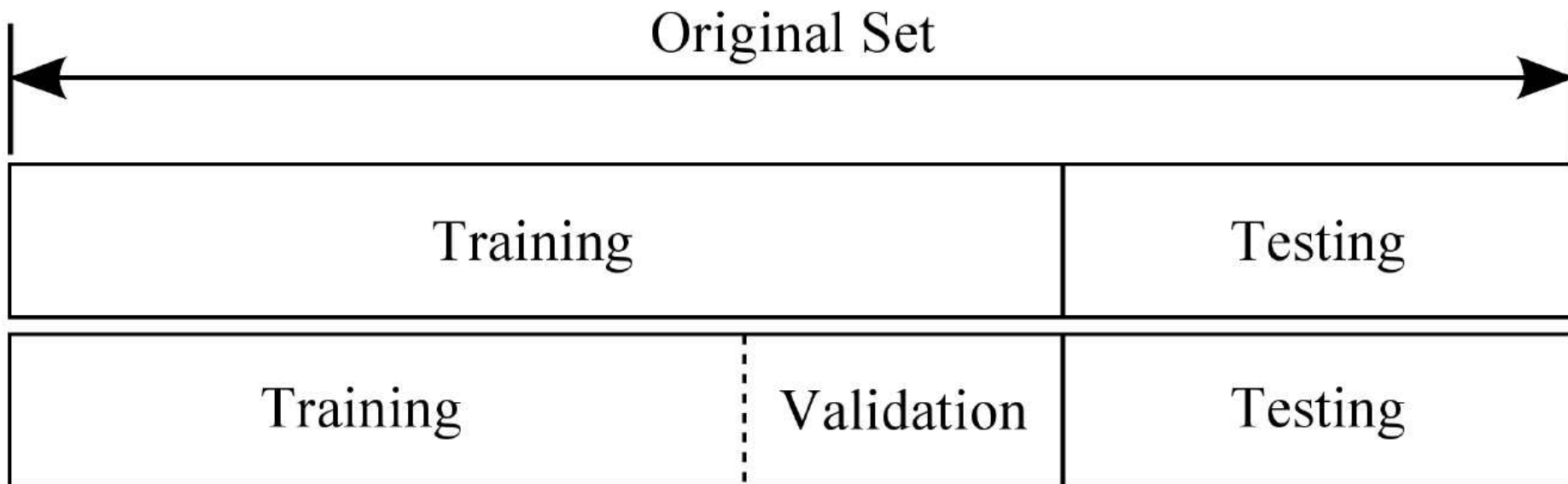
Performance evaluation: is this good?

EVALUATION USING TRAINING SET?

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

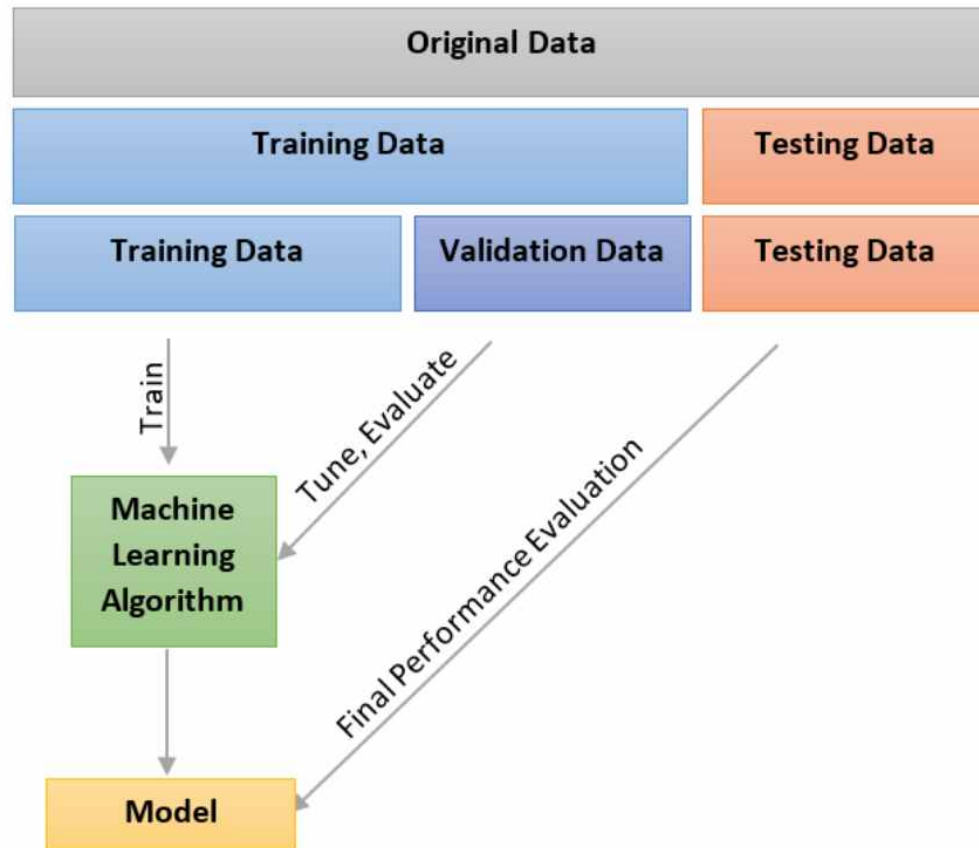
- 100% correct (accuracy)
- Can memorize

TRAINING, VALIDATION AND TEST SETS



모델 평가

TRAINING, VALIDATION AND TEST SETS



TRAINING, VALIDATION AND TEST SETS

- ▶ **Training set**
매개변수 학습
- ▶ **Validation set**
하이퍼파라미터 성능 평가
- ▶ **Test set**
모델 성능 평가

배치와 미니배치(Mini Batch)



- 배치 경사 하강법은 경사 하강법을 할 때, 전체 데이터를 사용하므로 가중치 값이 최적 값에 수렴하는 과정이 매우 안정적이지만, 계산량이 너무 많이 듭니다.
- **미니 배치 경사 하강법**은 경사 하강법을 할 때, 전체 데이터의 일부만을 보고 수행하므로 최적 값으로 수렴하는 과정에서 값이 조금 헤매기도 하지만 훈련 속도가 빠릅니다.
미니 배치의 크기를 batch size 라고 한다

The End