

## **M5. 인공신경망 기본 2**

# 인공신경망 기본개념 및 MNIST 과제소개

- Train/Val/Test
- Overfitting/Underfitting
- Performance Measure

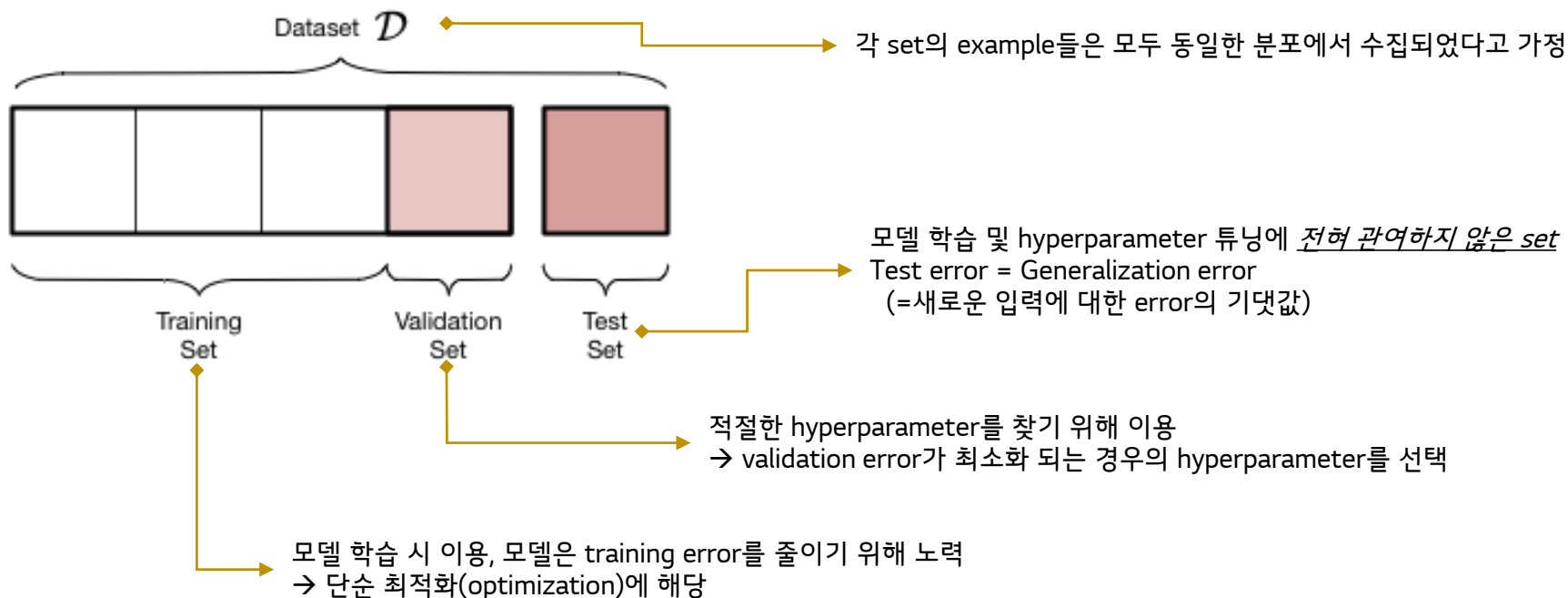
# Generalization

우리가 만든 알고리즘은 훈련 시 본 적이 없는 새로운 입력 데이터에 대해서도 잘 수행되어야 한다

## Generalization

The ability to perform well on *previously unobserved* inputs is called **generalization**

Generalization을 위해 데이터를 나누어 이용하자!



# Overfitting & Underfitting

내 모델이 과연 제대로 학습되었을까... ?

✓ Training set에 대한 error가 작아졌는가?

→ 모델은 내가 알려준 데이터(Training set)에 대해 정답을 잘 맞춰야 한다

→ If not?

## Underfitting

모델이 training set에 대해 충분히 낮은 training error에 도달하지 못한 경우 발생

✓ Training error와 test error의 갭이 작은가?

→ 모델은 내가 알려준 데이터도 잘 맞춰야 하지만, 본 적이 없는 데이터(Test set)에 대해서도 잘 맞춰야 한다.

→ If not?

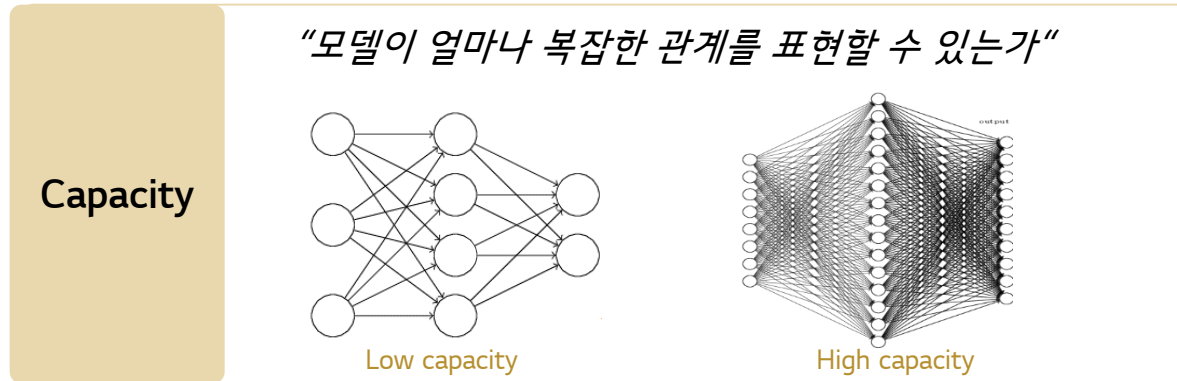
## Overfitting

Training error와 Test error의 갭이 너무 큰 경우 발생,  
즉, 알려 주는 데이터는 잘 맞추는데 처음 보는 데이터는 잘 못맞추는 경우

→ 태스크를 일반화하지 못하고 알려 준 데이터와 정답을 그냥 달달 외웠을 뿐이다..!

# Overfitting & Underfitting

Model capacity를 조절하여 overfitting, underfitting을 해결할 수 있다



Capacity가 낮은 모델은 training set을 학습하는 데 어려움을 겪는다  
즉, 너무 모델이 단순해서 데이터의 특징을 파악하기 힘들다

➡ Underfitting 발생!

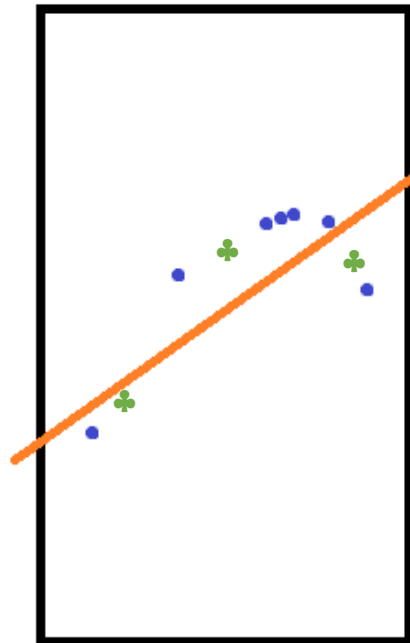
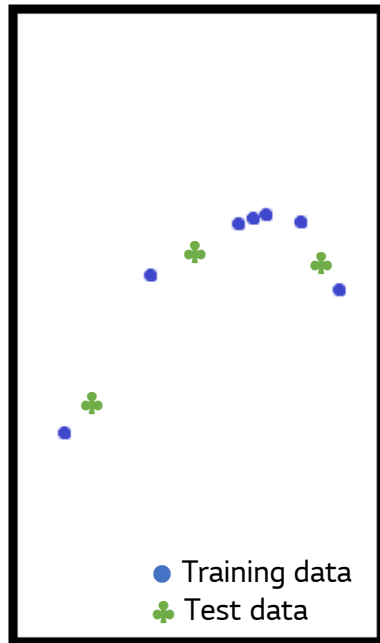
Capacity가 높은 모델은 training set의 속성을 외워버림으로써 과적합(overfit)이 되어,  
test set에 대해선 잘 작동하지 않을 우려가 있다

➡ Overfitting 발생!



# Overfitting & Underfitting

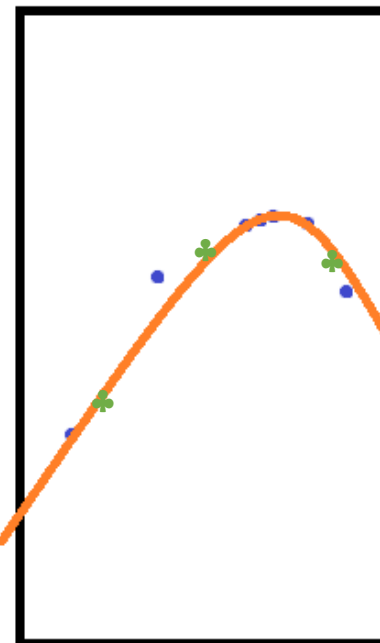
적절한 Capacity를 찾으면 Overfitting과 Underfitting을 방지할 수 있다



Low capacity  
Underfitting

못맞춤

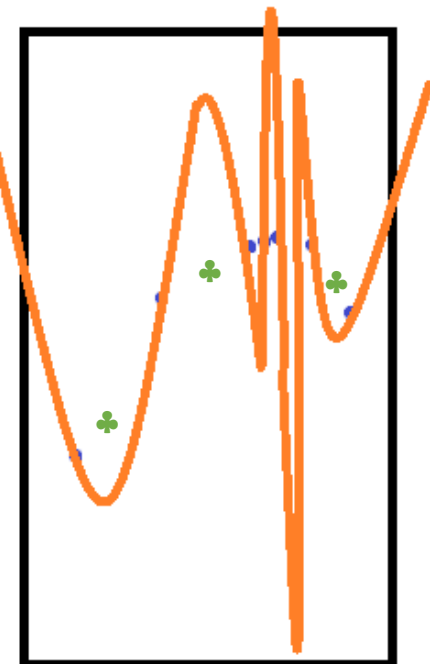
못맞춤



Appropriate Capacity

어느정도 잘맞춤

어느정도 잘맞춤



High capacity  
Overfitting

엄청 잘맞춤

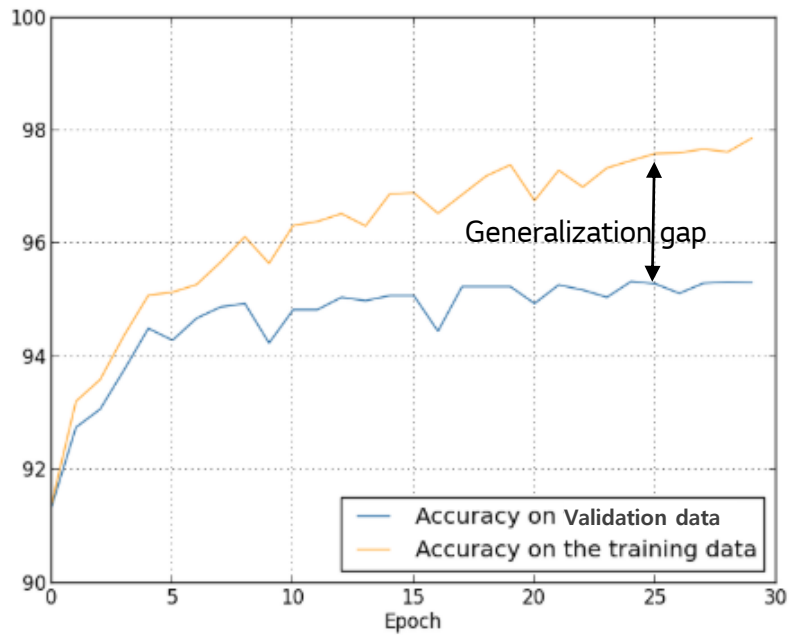
못맞춤

Training data

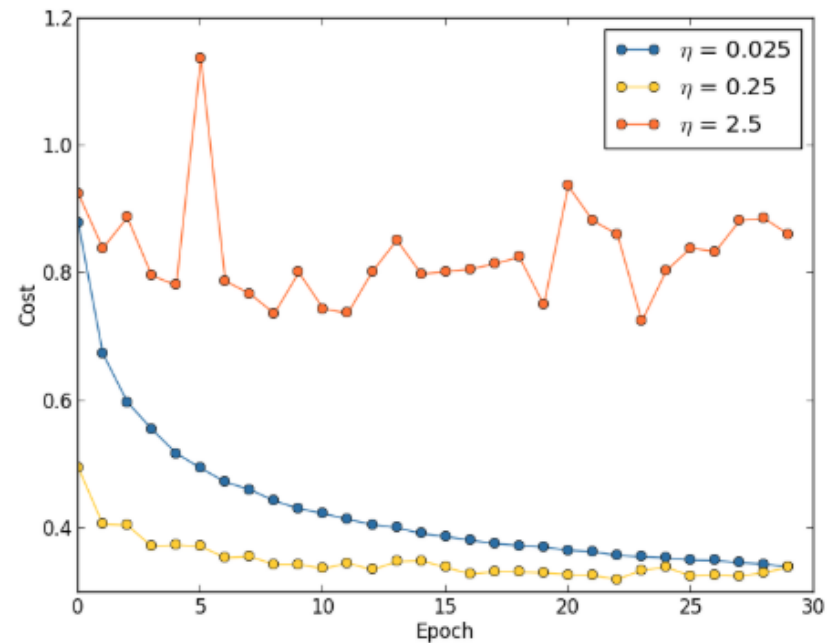
Test data

# Learning curve

Learning curve를 보고 모형 개선방안을 찾아보자



Overfitting의 조짐이 보이는가?!



적절한 Learning rate 찾기

# Performance measure

Task 목적에 따라 알맞은 measure가 달라진다

## Measurement

모델이 얼마나 과제를 잘 수행하는지를 객관적으로 수치화한 “성능 평가 지표”

이전에 보지 못했던 데이터에 대해 얼마나 잘 수행하는지가 중요하기 때문에  
Test set을 사용하여 성능을 측정

## 예) 양불판정 task

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive, Power	False positive, Type I error
	Predicted condition negative	False negative, Type II error	True negative

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

아 모르겠고, 우린 그냥 분류 정확도 높여주세요!

$$\text{Recall} = \frac{tp}{tp + fn}$$

(검출율)

불량제품을 판매하게 되면 브랜드이미지 타격이 큼니다.  
사람이 재검수해도 좋으니 김새만 있으면 다 골라내주세요!

$$\text{Precision} = \frac{tp}{tp + fp}$$

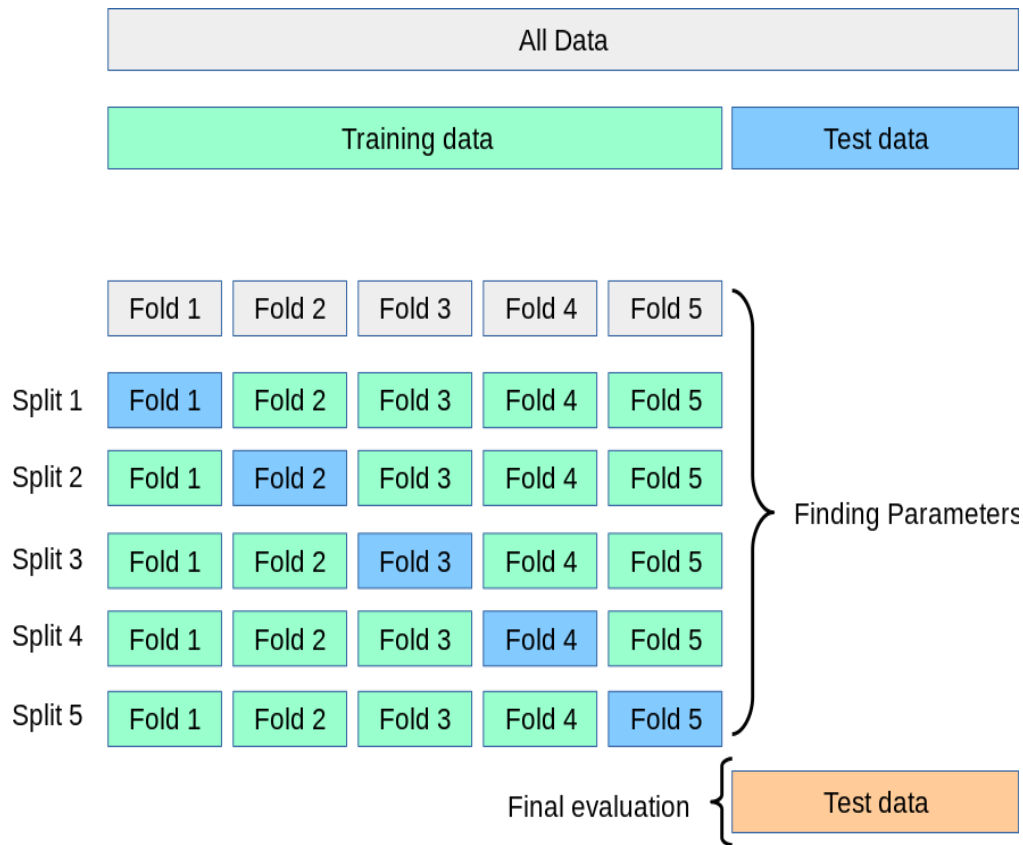
(정밀도)

모델이 불량이라고 판정한거 담당자가 다 보기 힘들어요...  
모델이 아무거나 다 불량이라고 해도 곤란해요



# k-Fold Cross Validation

데이터셋의 크기가 작은 경우 train/val/test 분배에 따라 성능 지표 변동이 크게 달라질 수 있는 문제가 발생, 이를 해결하기 위해 모든 데이터가 최소 한 번은 validation set으로 사용되도록 하는 방법



## [작업 프로세스]

1. Training set과 Test set으로 데이터를 나눈다. (Test set은 최종성능 평가를 위해 따로 빼놓는다.)
2. Training을 K개의 fold로 나눈다.(그림의 경우 K=5)
3. K개의 fold 중 1개는 Validation set으로 지정하고, K-1개는 Training set으로 지정한다.
4. K-1개의 Training set으로 모델을 생성하고, Validation set에 대해 예측을 진행하여, 이에 대한 성능지표를 추출한다.
5. 다음 fold에서는 Validation set을 바꿔서 지정하고, 이전 fold에서 Validation 역할을 했던 Set은 다시 Training set으로 활용한다.
6. 이를 모든 fold에 대해 K번 반복한다.