

강의계획표

주	해당 장	주제
1	1장	머신러닝이란
2	2장, 3장	머신러닝을 위한 기초지식, 구현을 위한 도구
3	4장	선형 회귀로 이해하는 지도학습
4	5장	분류와 군집화로 이해하는 지도 학습과 비지도 학습
5	6장	다양한 머신러닝 기법들 다항 회귀, 결정 트리, SVM
6		
7	7장	인공 신경망 기초 - 문제와 돌파구
8		중간고사
9	8장	고급 인공 신경망 구현
10	9장	신경망 부흥의 시작, 합성곱 신경망
11	10장	순환 신경망
12	11장	차원축소와 매니폴드 학습
13	12장	오토인코더와 잠재표현 학습
14	13장	인공지능의 현재와 미래
15		보강주
16		기말고사

Machine Learning을 위한 기초지식

청소년을 위한 인공지능 첫걸음 :: 인공지능, 머신러닝, 딥러닝 개념 한번에 이해하기

<https://www.youtube.com/watch?v=ai-exMIOxKw>

TED가 권유하는 7편의 인공지능 강연

<https://cooltool.com/blog/7-best-ted-videos-about-artificial-intelligence>

1. Machine Learning (1)

❖ 기계학습(機械學習, machine learning)

- 경험을 통해서 나중에 유사하거나 같은 일(task)를 더 효율적으로 처리할 수 있도록 시스템의 구조나 파라미터를 바꾸는 것
- 컴퓨터가 데이터로부터 특정 문제해결을 위한 지식을 데이터에서 자동으로 추출해서 사용할 수 있게 하는 기술

경험	일	효율(성능)
필기문자 이미지, 글자	문자 판독(인식)	정확도
사진, 얼굴영역	사진에서 얼굴영역 식별	정확도
이메일, 스팸여부	스팸 이메일 판단	정확도
풍경 사진	유사한 풍경 사진 식별	유사도
바둑 대국	바둑두는 방법	승률

1. Machine Learning (2)

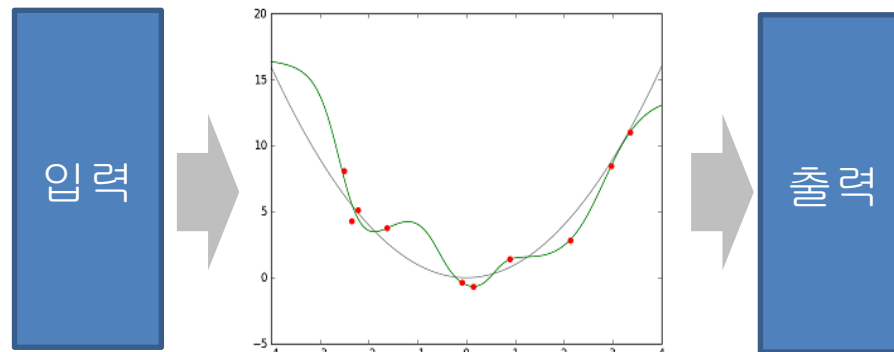
❖ 연역적 학습 vs. 귀납적 학습

연역법(Deduction)	귀납법(Induction)
1) 모든 포유동물은 심장을 가지고 있다. 2) 모든 말은 포유동물이다. ∴ 모든 말은 심장을 가지고 있다.	1) 참새는 하늘을 난다. 2) 제비는 하늘을 난다. ∴ 모든 새는 하늘을 난다.

★ 자연 과학적 방법론-귀납법과 연역법★				
	사유, 지식의 근원	진리인식방법	생활에의 적용	대표적 사상
귀납법 (induction)	후천적 경험 (經驗)	경험적 관찰과 실험 의한 지식의 습득	원칙과 공식의 발견	영국의 경험론
연역법 (deduction)	선천적 이성 (理性)	과학적 논리와 추리에 의함.	원칙과 공식의 적용	대륙의 합리론

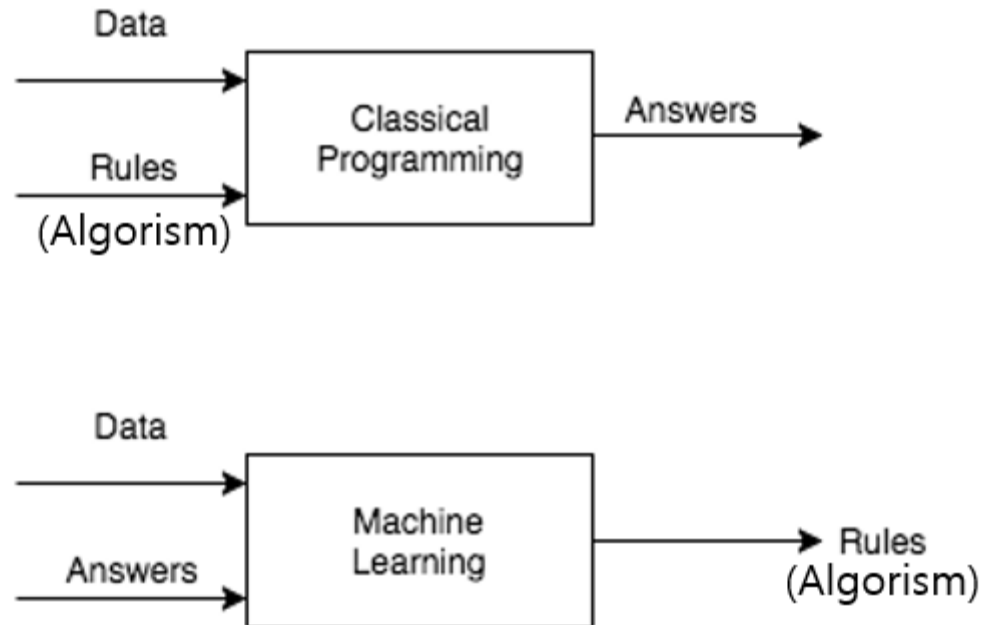
❖ 귀납적 학습 (inductive learning)

- 사례들 examples을 **일반화** generalization하여 **패턴** pattern, **모델** (model)을 추출
- 일반적인 기계학습의 대상
- 학습 데이터를 **잘 설명**할 수 있는 **패턴**을 찾는 것
 - **오컴의 면도날** Occam's razor: 가능하면 간단한 형태로 표현하는 것이 좋다



기계 학습(machine learning) == 함수 근사(function approximation)

1. Machine Learning (3)



< 전통적인 프로그래밍 Vs Machine Learning 비교 >

1. Machine Learning (4)

표 1-1 사람의 학습과 기계 학습의 비교

기준	사람의 학습	기계 학습
학습 과정	능동적	수동적
데이터 형식	자연에 존재하는 그대로	일정한 형식에 맞추어 사람이 준비함
동시에 학습 가능한 과업 수	자연스럽게 여러 과업을 학습	하나의 과업만 가능
학습 원리에 대한 지식	매우 제한적으로 알려져 있음	모든 과정이 밝혀져 있음
수학 의존도	매우 낮음	매우 높음
성능 평가	경우에 따라 객관적이거나 주관적	객관적(수치로 평가, 예를 들어 정확률 99.8%)
역사	수백만 년	60년 가량

1. Machine Learning (5)

❖ Applications



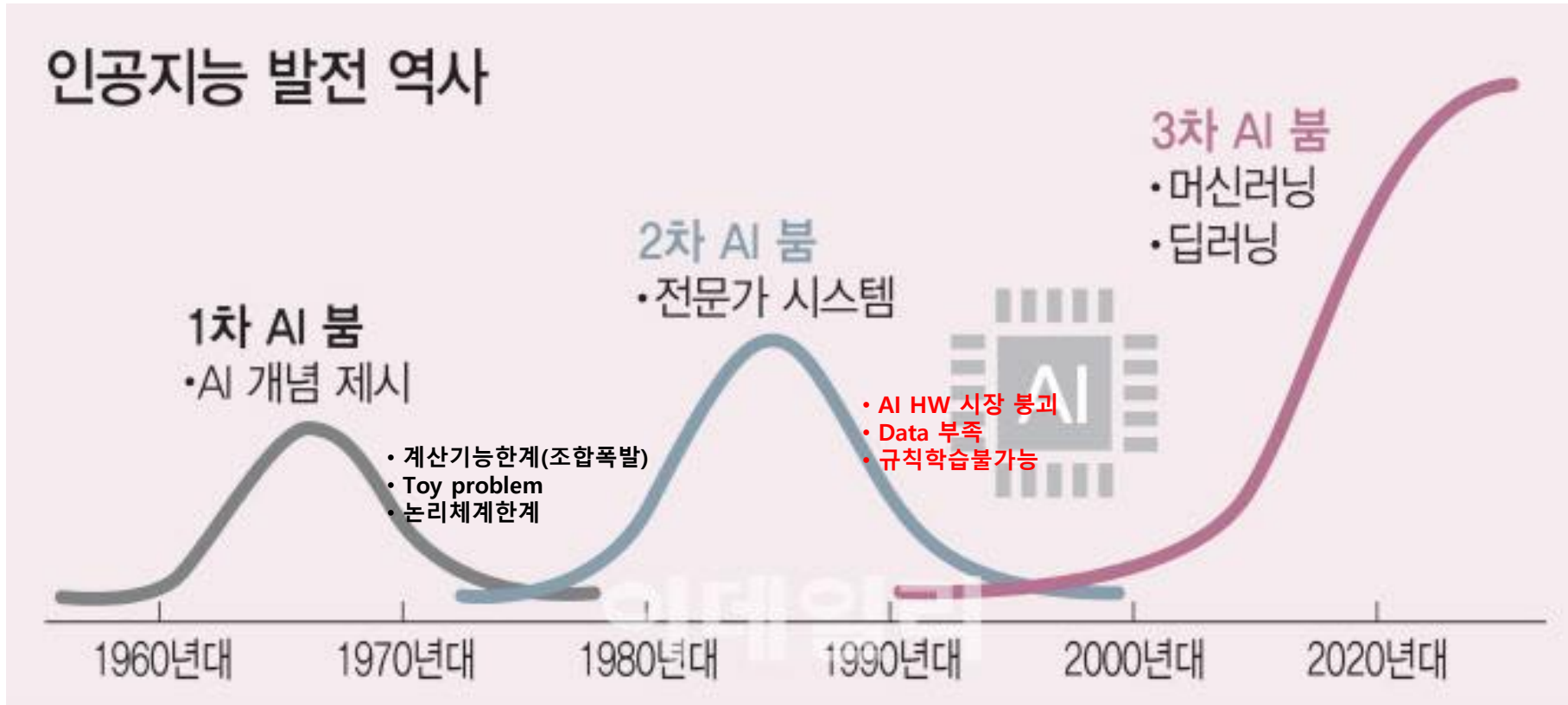
1. Machine Learning (6)

❖ Applications

- 복잡한 데이터들이 있고, 이들 데이터에 기반하여 결정을 내려야 하는 분야이다.
- 규칙과 공식이 너무 복잡할 때(영상인식, 음성인식)
- 작업 규칙이 지속적으로 바뀌는 상황(보안시스템의 침입탐지, 신용카드 사용 이상 감시)
- **데이터 특징이 계속 바뀌고** 프로그램을 계속해서 변경해야 하는 상황 (주식 거래나 에너지 수요 예측, 쇼핑 추세 예측)
- 구매자가 클릭할 확률이 가장 높은 광고가 무엇인지를 알아내는 시스템
- 분류(spam 메일)
- 신용카드 거래가 허위인지 판별하는 시스템
- 추천 시스템(넷플릭스)
- 이미지 인식/탐색 시스템
- 자율 주행자동차
- 텍스트 자동 인식, 기계번역

2. History of AI/ML/DL (1)

- <https://projectresearch.co.kr/2017/06/14/머신러닝ml의-간략한-역사/>
- <https://www.javatpoint.com/machine-learning>

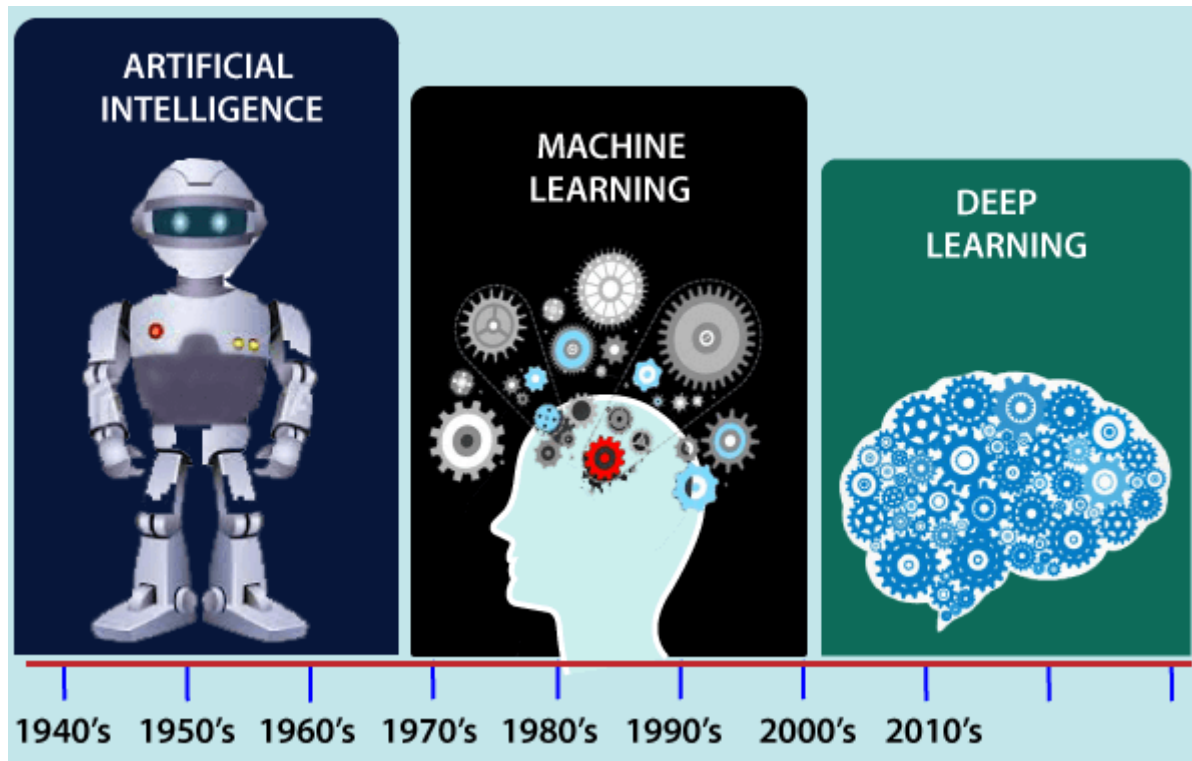


<https://www.edaily.co.kr/news/read?newsId=01279206625937512&mediaCodeNo=257>



2. History of AI/ML/DL (1)

❖ AI, ML, DL



<https://www.javatpoint.com/machine-learning>

3. Algorithms of ML (1)

❖ 지도학습(supervised learning)

- 입력(**문제**)-출력(**답**)의 데이터들로 부터 새로운 입력에 대한 출력을 결정할 수 있는 패턴 추출
- 분류(classification)
- 회귀(regression)

❖ 비지도학습(unsupervised learning, **자율학습**)

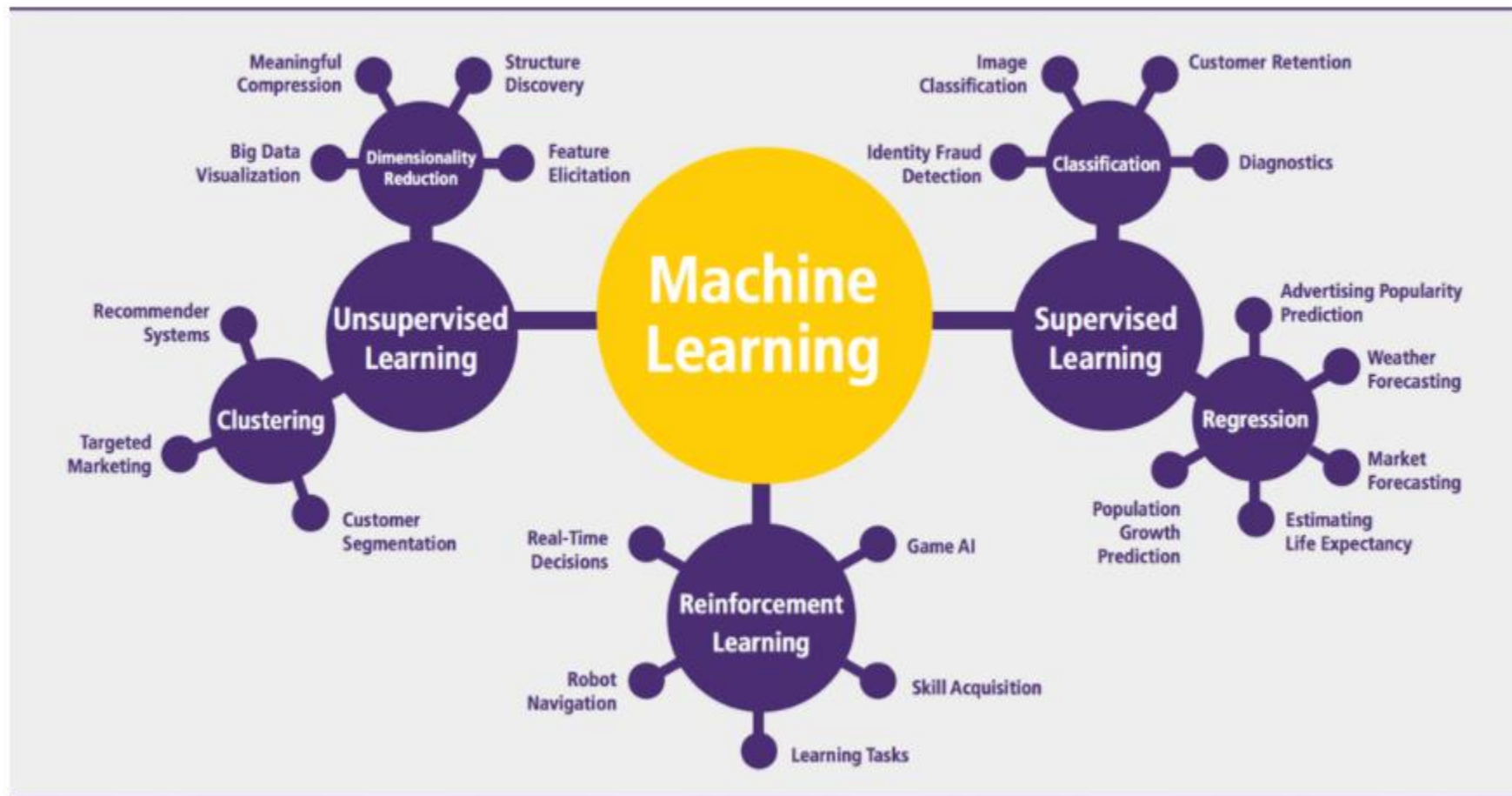
- **출력**에 대한 정보가 **없는 데이터**로부터 필요한 패턴 추출
- 군집(clustering)
- 차원축소(dimension reduction)

❖ 강화학습(reinforcement learning)

- 출력에 대한 정확한 정보를 제공하지는 않지만, **평가정보**(reward)는 주어지는 문제에 대해 **각 상태에서의 행동**(action)을 결정

3. Algorithms of ML (2)

Machine learning Algorithms and where they are used?



3. Algorithms of ML (3)

❖ 생성모델(Generative approach)

- 관찰된 데이터들이 어떻게 생성되었는지를 생각하여 클래스별로 그러한 생성시스템을 설명할 수 있는 확률분포를 먼저 추정, 결합확률
- $P(x|C_k)$: 클래스별 확률밀도함수
- class의 분포에 주목
- GAN, GPT

❖ 판별모델(Discriminative approach)

- 분류에 필요한 확률값 혹은 그와 유사한 판별함수의 값을 추정하여 사용. 조건부 확률
- $P(C_k|x)$:
- class의 차이점에 주목
- 판별에 기준이 되는 값만을 학습
- 확률분포의 복잡도에 영향을 받지 않음

❖ <https://m.blog.naver.com/PostView.nhn?blogId=ehdrnidd&logNo=221520140545&proxyReferer=https:%2F%2Fwww.google.com%2F>

❖ <https://runrun-learn.tistory.com/13>

❖ <https://blog.daum.net/hazzling/17186886>

❖ Example:

data in form (x, y)

$(1,0), (1,0), (2,0), (2,1)$

❖ $P(x, y)$

	y=0	y=1
x=1	1/2	0
x=2	1/4	1/4

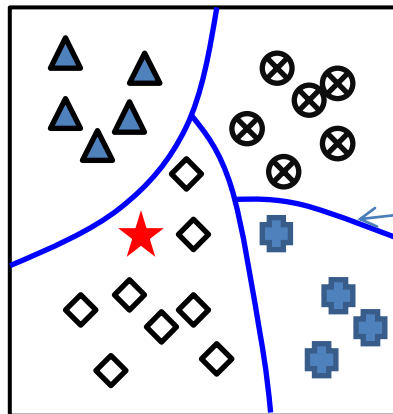
❖ $P(y|x)$

	y=0	y=1
x=1	1	0
x=2	1/2	1/2

4. Supervised Learning (1)

❖ 분류(classification)

- 데이터들을 정해진 몇 개의 부류(class)로 대응시키는 문제



결정 경계
(decision boundary)

- 분류 문제의 학습
 - 학습 데이터를 잘 분류할 수 있는 함수를 찾는 것
 - 함수의 형태는 수학적 함수일 수도 있고, 규칙일 수도 있음
- 분류기(classifier)
 - 학습된 함수를 이용하여 데이터를 분류하는 프로그램

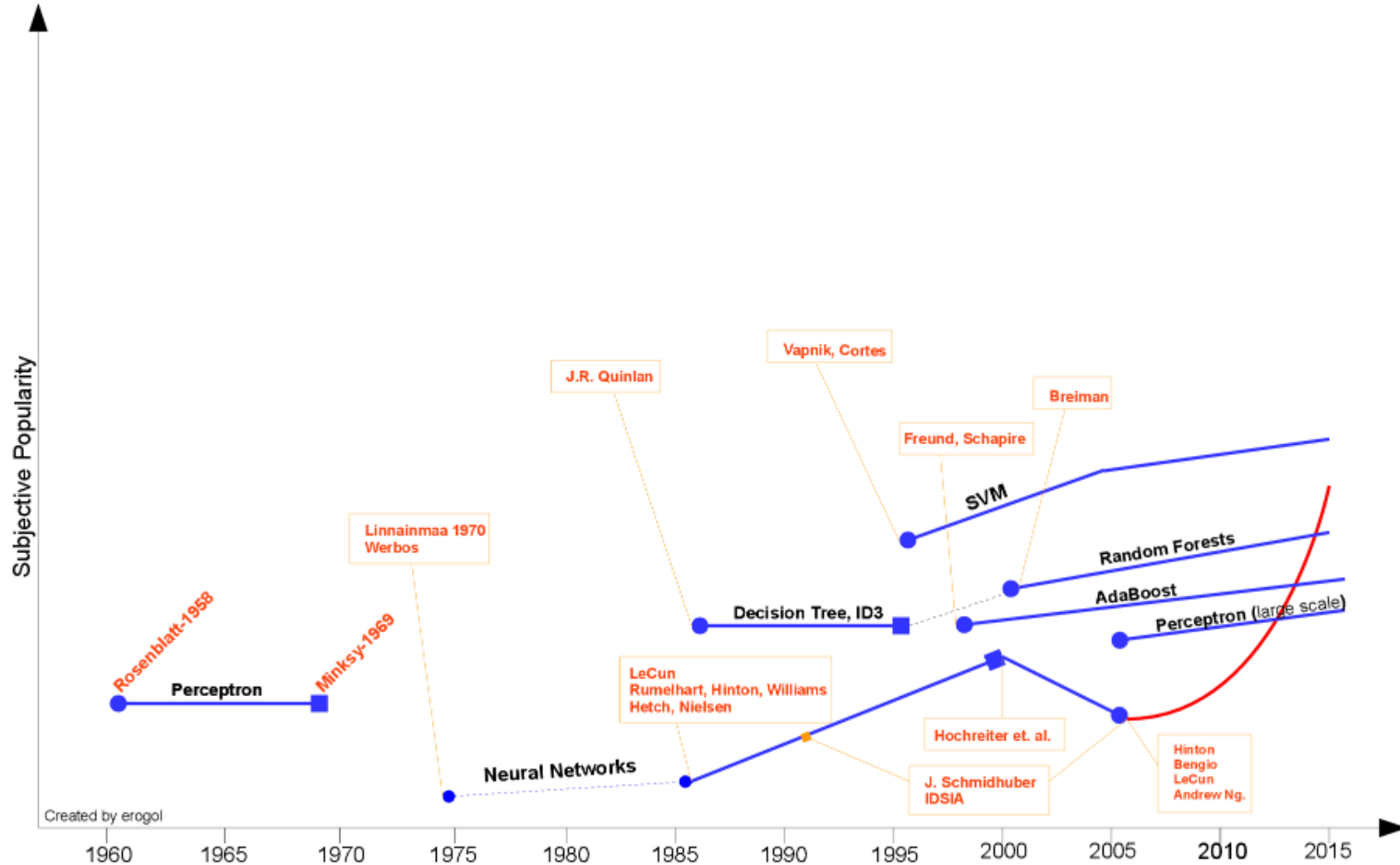
4. Supervised Learning (2)

❖ 분류기 학습 알고리즘

- Naïve Bayes 알고리즘
- Logistic Regression 알고리즘
- K-근접이웃 (K-nearest neighbor, KNN) 알고리즘
- 결정트리(decision tree) 알고리즘
- 랜덤 포리스트(random forest)
- 서포트 벡터 머신(Support Vector Machine, SVM)
- 다층 퍼셉트론 신경망
- 딥러닝(deep learning) 알고리즘
- 앙상블(Ensemble) 알고리즘: 배깅(bagging), 에이다부스트(AdaBoost)
- 확률 그래프 모델 (probabilistic graphical model)

4. Supervised Learning (3)

❖ 지도학습 알고리즘 개발



<http://www.erogol.com/brief-history-machine-learning/>

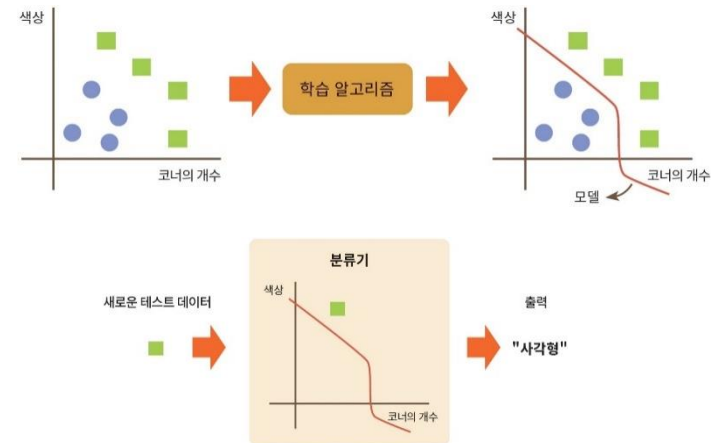
4. Supervised Learning (4)

❖ 이상적인 분류기

- 학습에 사용되지 않은 데이터에 대해서 분류를 잘 하는 것
- 일반화(generalization) 능력이 좋은 것

❖ 데이터의 구분

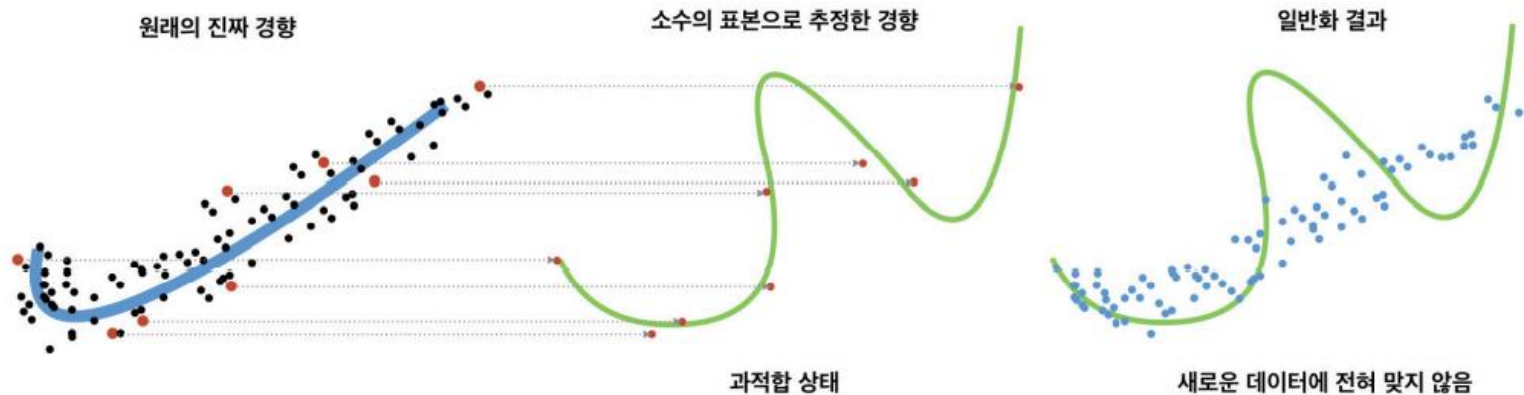
- 학습 데이터(training data)
 - 분류기(classifier)를 학습하는데 사용하는 데이터 집합
 - 학습 데이터가 많을 수록 유리
- 테스트 데이터(test data)
 - 학습된 모델의 성능을 평가하는데 사용하는 데이터 집합
 - 학습에 사용되지 않은 데이터이어야 함
- 검증 데이터(validation data)
 - 학습 과정에서 학습을 중단할 시점을 결정하기 위해 사용하는 데이터 집합



4. Supervised Learning (5)

❖ 과적합^{overfitting}

- 학습 데이터에 대해서 지나치게 잘 학습된 상태
- 데이터는 오류나 잡음을 포함할 개연성이 크기 때문에, 학습 데이터에 대해 매우 높은 성능을 보이더라도 학습되지 않은 데이터에 대해 좋지 않은 성능을 보일 수 있음
- 일반화^{generalization}가 제대로 되지 않는다는 것을 의미

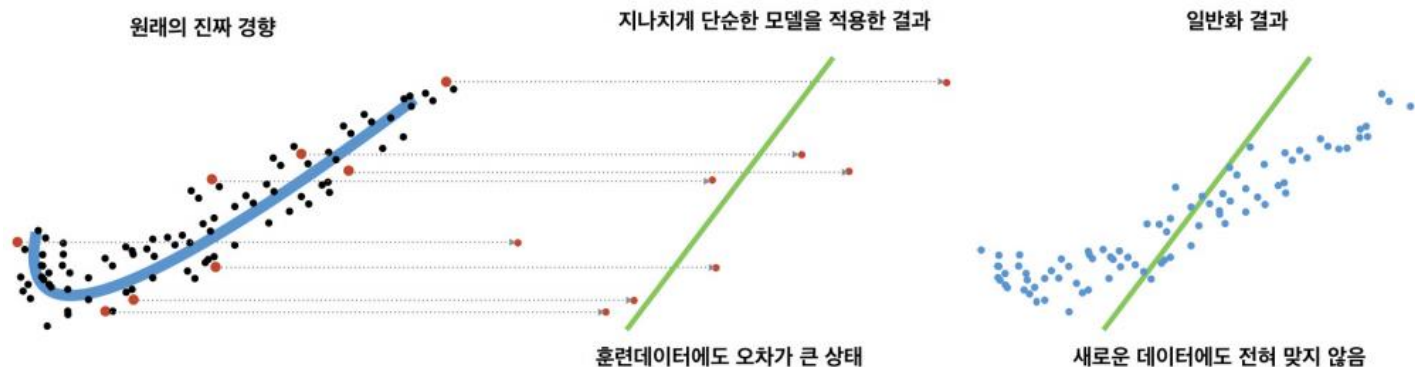


- 원인
 - 학습에 사용된 데이터의 수가 너무 적다는 것
 - 데이터를 설명하기 위해 사용된 녹색 곡선이 너무 복잡하다는 것
- 모델이 복잡하면 불리하게 만드는 복잡성에 대한 규제^{regulation} 혹은 정칙화^{regularization} 기법을 이용

4. Supervised Learning (6)

❖ 과소적합 underfitting

- 학습 데이터를 충분히 학습하지 않은 상태
- 새로운 데이터뿐만 아니라 학습 데이터조차 제대로 설명하지 못하는 모델

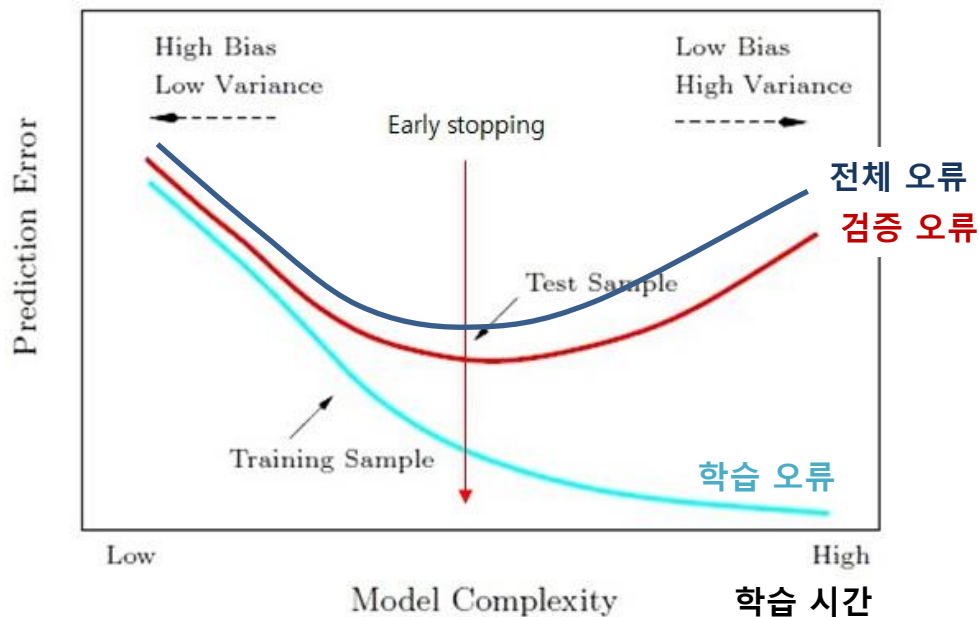


- 원인
 - 모델이 지나치게 단순한 경우 발생한다
 - 예측을 제대로 할 수 없는 특징들만 제공되는 경우에 발생한다.
- 입력 데이터의 특징을 바꾸거나 학습 모델의 복잡도를 높이는 방법으로 문제를 해결

4. Supervised Learning (7)

❖ 과적합 회피 방법

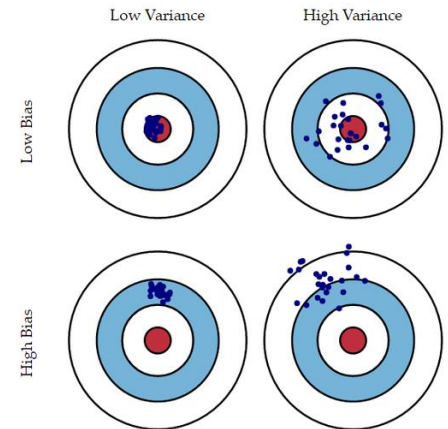
- 학습데이터에 대한 성능
 - 학습을 진행할 수록 오류 개선 경향
 - 지나치게 학습이 진행되면 과적합 발생
- 학습과정에서 별도의 **검증 데이터**(validation data)에 대한 성능 평가
 - 검증 데이터에 대한 오류가 감소하다가 증가하는 시점에 학습 중단



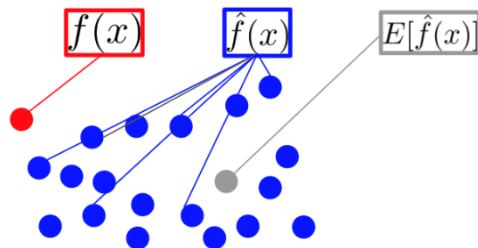
4. Supervised Learning (8)

❖ 기계학습에서의 Bias vs. Variance 관계

- <https://www.opentutorials.org/module/3653/22071>
- 예측값들과 정답이 대체로 멀리 떨어져 있으면 결과의 편향(bias)이 높다
- 예측값들이 자기들끼리 대체로 멀리 흩어져 있으면 결과의 분산(variance)이 높다



- 빨간 점: '정답'
- 여러 번 찍힌 점: 예측한 값



- x : 데이터
- $f(x)$: 정답
- $\hat{f}(x)$: 예측값
- $E[\hat{f}(x)]$: 기대값(평균)

▪ 편향:

Bias: How much predicted value differ from true values

$$\frac{(E[\hat{f}(x)] - f(x))^2}{\text{average predicted value} \quad \text{true}}$$

▪ 분산:

Variance: How predictions made on the same value vary on different realizations of the model

$$\frac{E[(\hat{f}(x) - E[\hat{f}(x)])^2]}{\text{predicted value} \quad \text{average predicted value}}$$

▪ 오차:

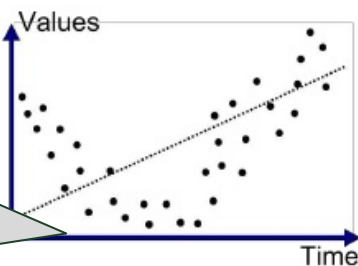
$$Error(x) = (E[\hat{f}(x)] - f(x))^2 + E[(\hat{f}(x) - E[\hat{f}(x)])^2] + \sigma_e^2$$

근본적
오차

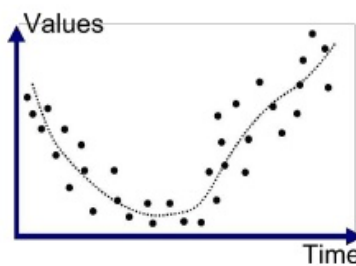
4. Supervised Learning (9)

❖ 회귀모델에서의 편향과 분산

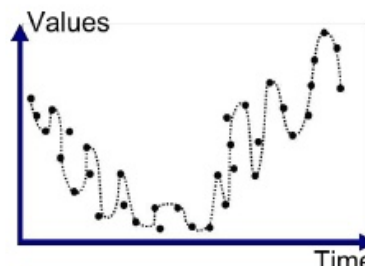
- 정답과 예측은 멀:
높은 편향
- 모델의 출력값들은
직선상에 있음: 적은
분산



high bias
low variance



medium bias
medium variance

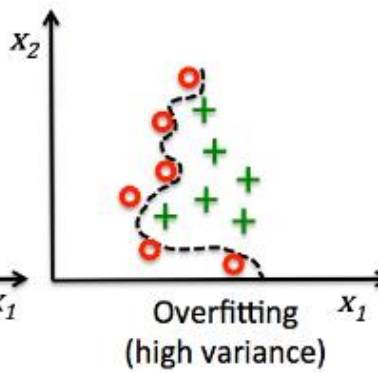
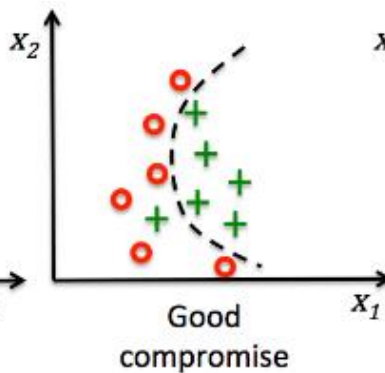
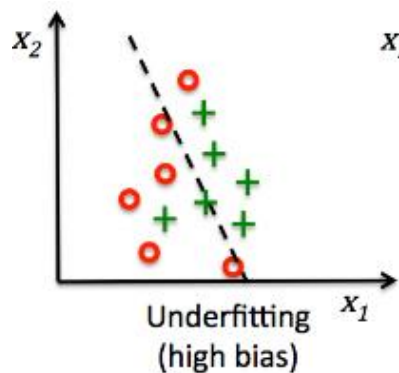


low bias
high variance

- 정답: 점
- 직선/곡선: 예측모델

- 정답과 예측은 거의
같음: 적은 편향
- 모델의 출력값들은
흩어져 있음: 높은 분산

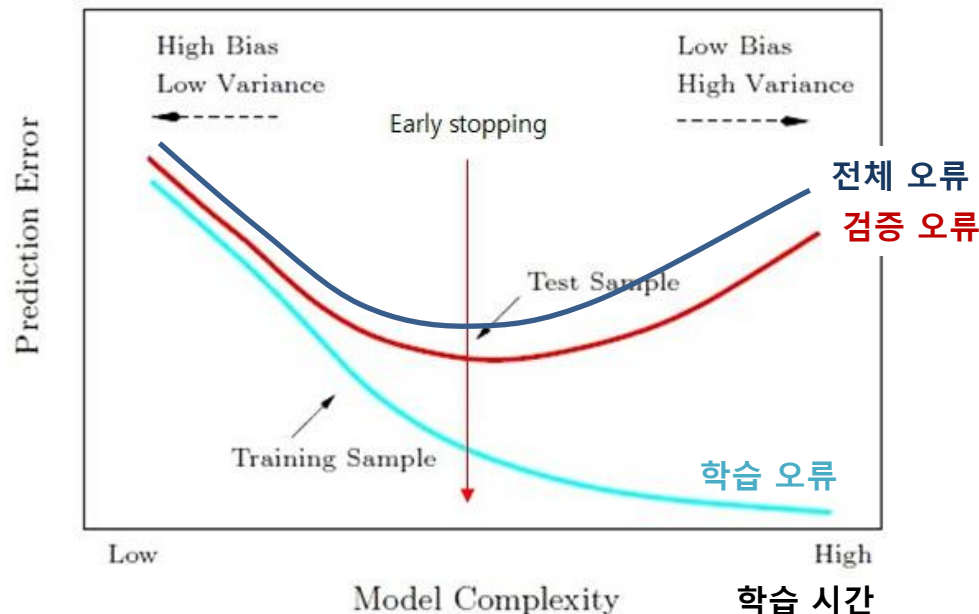
❖ 분류회귀모델에서의 편향과 분산



4. Supervised Learning (10)

❖ Model Complexity

- Training Error: training용 데이터로 모델을 훈련시킬 때 발생하는 오차로, 모델이 데이터를 반복 학습하는 횟수가 늘어날수록 모델의 복잡한 정도도 따라서 늘어남 => training error는 줄어듦
- Validation Error: 데이터로 모델을 평가할 때 발생하는 오차로, 어느 정도까지는 줄어들다가 어느 지점 이후부터는 다시 상승 => Validation Error가 최소인 지점에서 훈련을 멈춤



4. Supervised Learning (11)

❖ 분류기의 성능 평가

- **정확도 (accuracy)**
 - 얼마나 정확하게 분류하는가
 - **정확도** = (옳게 분류한 데이터 개수)/(전체 데이터 개수)
 - **테스트 데이터**에 대한 정확도를 분류기의 정확도로 사용
- 정확도가 높은 분류기를 학습하기 위해서는 **많은 학습데이터**를 사용하는 것이 유리
- **학습데이터와 테스트 데이터는 겹치게 않도록** 해야 함

4. Supervised Learning (12)

❖ 이진 분류기의 성능 평가

- 두 개의 부류만을 갖는 데이터에 대한 분류기

표 4.2 이진 분류기의 혼동행렬

		예 측	
		양성	음성
실 제	양성	진양성(True Positive) <i>TP</i>	위음성(False Negative) <i>FN</i>
	음성	위양성(False Positive) <i>FP</i>	진음성(True Negative) <i>TN</i>

- 자율주행차: 보행자가 있으면 양성(긍정), 없으면 음성(부정)
- 암환자 진료: 암이면 긍정, 암이 아니면 부정
- 불량품 검출: 불량품이면 긍정, 정상품이면 부정

4. Supervised Learning (13)

- 민감도(sensitivity)/재현율(recall)

$$\text{민감도} = \frac{TP}{TP + FN}$$

- 특이도(specificity)/신음성율(true negative rate)

$$\text{특이도} = \frac{TN}{FP + TN}$$

- 정밀도(precision) 정밀도 = $\frac{TP}{TP + FP}$

- 음성 예측도 음성 예측도 = $\frac{TN}{TN + FN}$

- 위양성율 위양성율 = $\frac{FP}{FP + TN} = 1 - \text{특이도}$

- 위발견율 위발견율 = $\frac{FP}{TP + FP} = 1 - \text{정밀도}$

- 정확도 정확도 = $\frac{TP + TN}{TP + FP + TN + FN}$

- F1 척도
$$F1 = 2 \frac{(\text{정밀도}) \cdot (\text{재현율})}{(\text{정밀도}) + (\text{재현율})} \quad F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{FN + FP}{2}}$$

표 4.2 이진 분류기의 혼동행렬

		예 측	
		양성	음성
실 제	양성	진양성(True Positive) TP	위음성(False Negative) FN
	음성	위양성(False Positive) FP	진음성(True Negative) TN

- 의사의 환자 진료: 특이도, 민감도

- 특이도 = $\frac{\text{정상예측}(TN)}{\text{실제정상}(FP + TN)}$

- 민감도 = $\frac{\text{환자예측}(TP)}{\text{실제환자}(TP + FN)}$

- 정보검색, 물체검출: 정밀도, 재현율

- 정밀도 = $\frac{\text{긍정검색}(TP)}{\text{전체(긍정)검색결과}(TP + FP)}$

- 재현율 = $\frac{\text{긍정검색}(TP)}{\text{실제데이터}(TP + FN)}$

4. Supervised Learning (14)

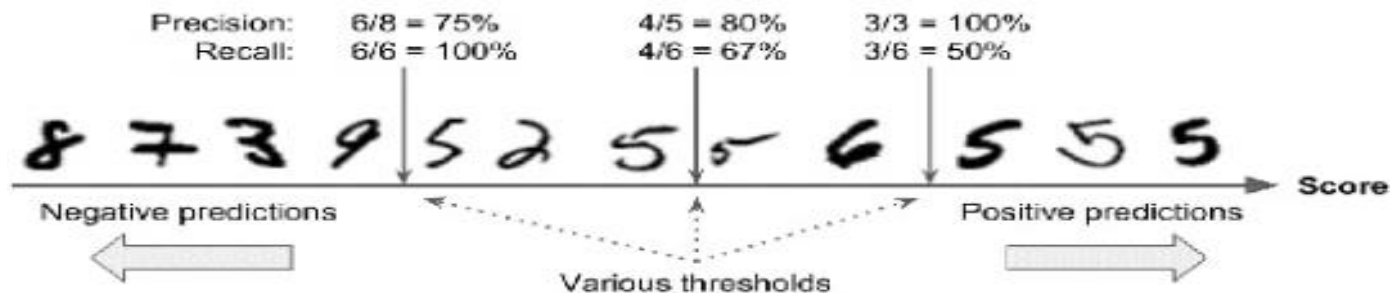
❖ 이진 분류기의 성능 평가 – cont.

▪ 예제1:

- True labels = [2, 0, 0, 2, 4, 4, 1, 0, 3, 3, 3]
- Predict labels = [2, 1, 0, 2, 4, 3, 1, 0, 1, 3, 3]

Class	True	predict	precision	recall	F1
0	3	2	$2/2=1.00$	$2/3=0.67$	0.80
1	1	3	$1/3=0.33$	$1/1=1.00$	0.50
2	2	2	$2/2=1.00$	$2/2=1.00$	1.00
3	3	3	$2/3=0.67$	$2/3=0.67$	0.67
4	2	1	$1/1=1.00$	$1/2=0.50$	0.67

▪ 예제2: '5'에 대한 평가 결과 (정확률 vs. 재현률의 tradeoff)



4. Supervised Learning (15)

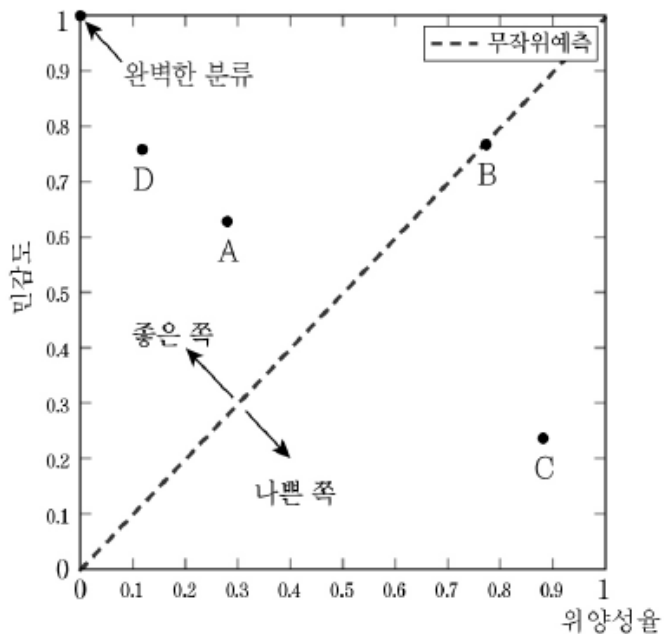
❖ 이진 분류기의 성능 평가 – cont.

▪ ROC 곡선

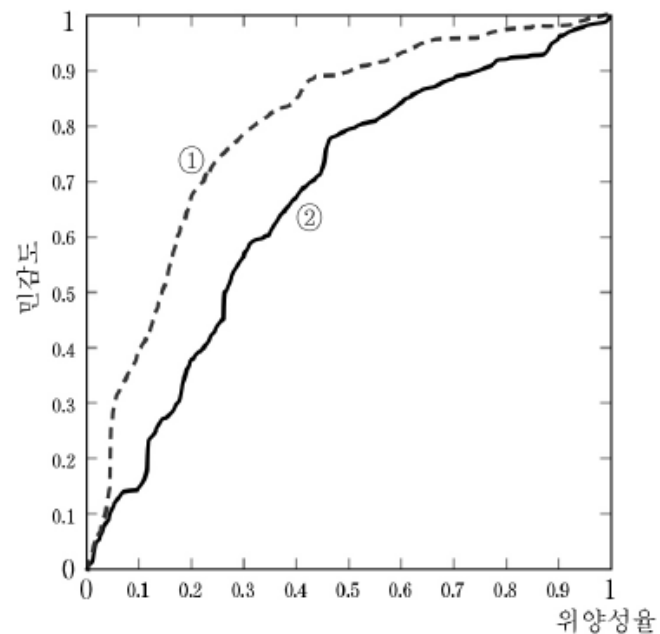
- 부류 판정 임계값에 따른 (위양성율, 민감도) 그래프

▪ AUC(Area Under the Curve)

- ROC 곡선에서 곡선 아래 부분의 면적
- 클 수록 바람직



(a)



(b)

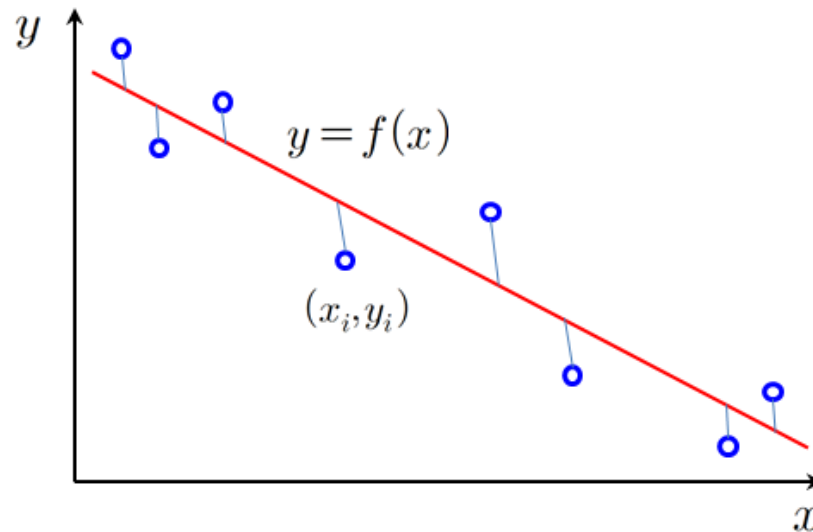
4. Supervised Learning (16)

❖ 회귀 (regression)

- 학습 데이터에 가장 잘 근사(approximation)되는 출력값이 실수인 함수를 찾는 문제

$$f^*(x) = \arg \min_f \sum_{i=1}^n (\mathbf{y}_i - \mathbf{f}(x_i))^2$$

- $\arg \min_f$: 목적함수를 최소로 하는 f 선택



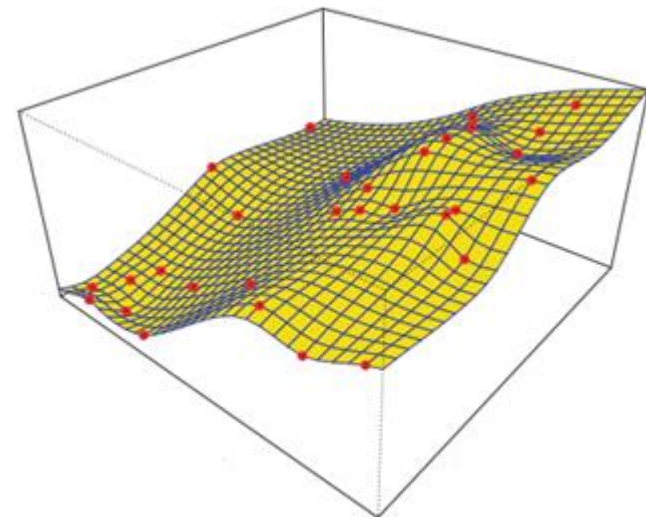
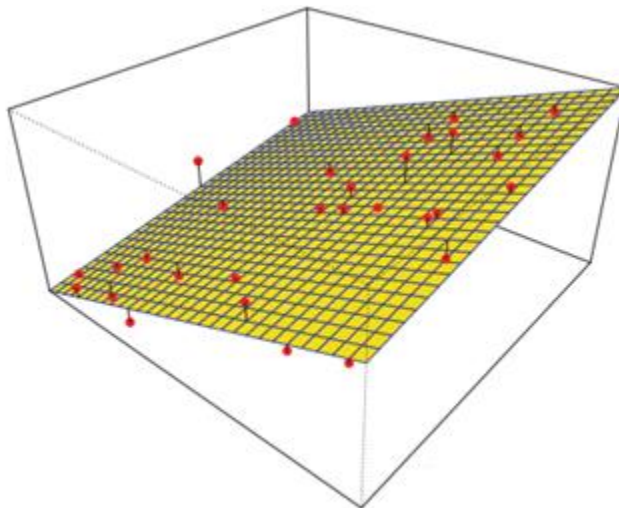
4. Supervised Learning (17)

❖ 회귀 성능

- 오차 : 예측값과 실제값의 차이
 - 테스트 데이터들에 대한 (예측값 - 실제값)²의 평균 또는 평균의 제곱근

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

- 모델의 종류(함수의 종류)에 영향을 받음



5. Unsupervised Learning (1)

❖ 비지도학습(unsupervised learning)

- 결과정보가 없는 데이터들에 대해서 특정 패턴을 찾는 것
 - 데이터에 잠재한 구조(structure), 계층구조(hierarchy)를 찾아내는 것
 - 숨겨진 사용자 집단(hidden user group)을 찾는 것
 - 문서들을 주제에 따라 구조화하는 것
 - 로그(log) 정보를 사용하여 사용패턴(usage pattern)을 찾아내는 것

▪ 비지도 학습의 대상

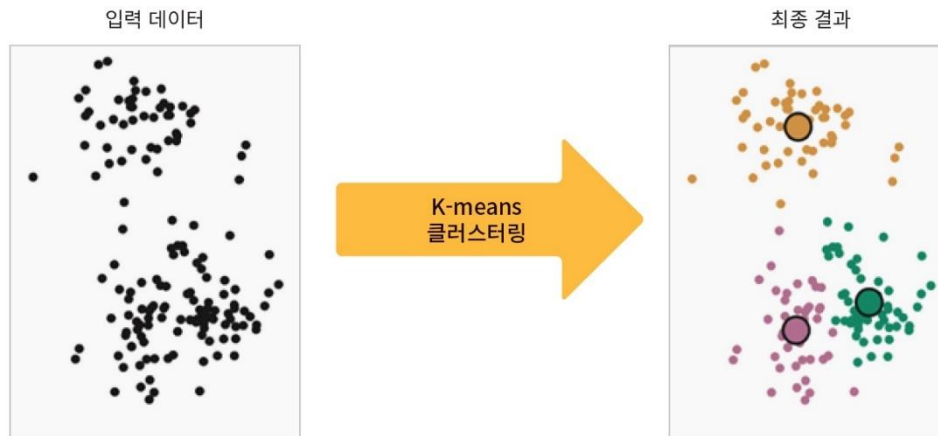
- 군집화(clustering)
- 밀도추정(density estimation)
- 차원축소(dimensionality reduction)



5. Unsupervised Learning (2)

❖ 군집화(clustering)

- 유사성에 따라 데이터를 분할하는 것(문서, 고객세분화, 생물학:소군집)



유사 영상의 군집



영상 분할(segmentation)

5. Unsupervised Learning (3)

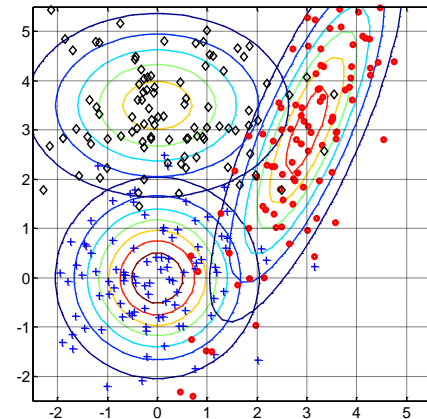
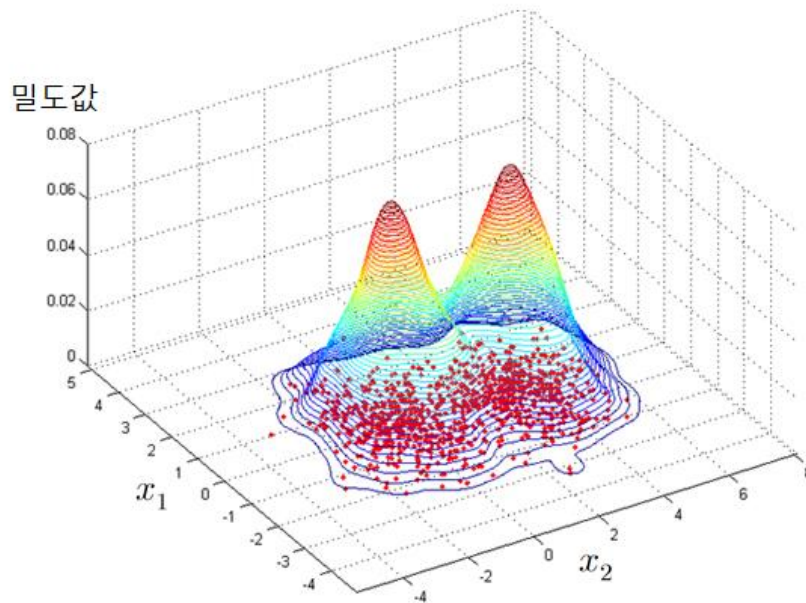
❖ 군집화 – cont.

- **일반 군집화**(hard clustering)
 - 데이터는 하나의 군집에만 소속
 - 예. k-means 알고리즘
- **퍼지 군집화**(fuzzy clustering)
 - 데이터가 여러 군집에 부분적으로 소속
 - 소속정도의 합은 1이 됨
 - 예. 퍼지 k-means 알고리즘
- **용도**
 - 데이터에 내재된 구조(underlying structure) 추정
 - 데이터의 전반적 구조 통찰
 - 가설 설정, 이상치(anomaly, outlier) 감지
 - 데이터 압축 : 동일 군집의 데이터를 같은 값으로 표현
 - 데이터 전처리(preprocessing) 작업
- **성능**
 - 군집내의 분산과 군집간의 거리

5. Unsupervised Learning (4)

❖ 밀도 추정(density estimation)

- 부류(class)별 데이터를 만들어 냈을 것으로 추정되는 확률분포를 찾는 것



▪ 용도

- 각 부류 별로 주어진 데이터를 발생시키는 확률 계산
- 가장 확률이 높은 부류로 분류

5. Unsupervised Learning (5)

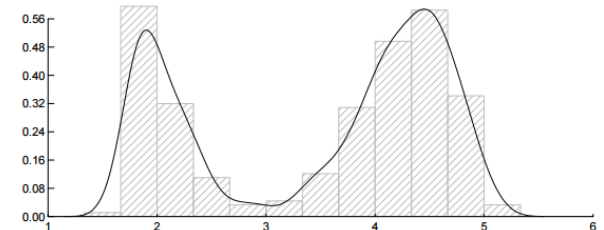
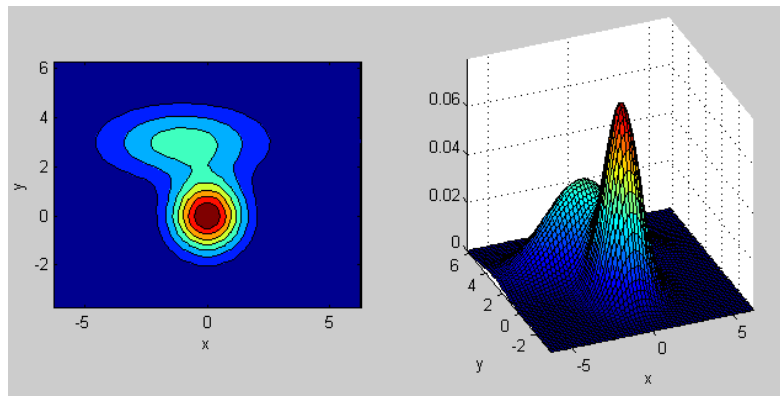
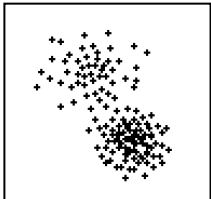
❖ 밀도 추정 – cont.

■ 모수적(parametric) 밀도 추정

- 분포가 특정 수학적 함수의 형태를 가지고 있다고 가정
- 주어진 데이터를 가장 잘 반영하도록 함수의 파라미터 결정
- 전형적인 형태 : 가우시안(Gaussian) 함수 또는 여러 개의 가우시안 함수의 혼합(Mixture of Gaussian)

■ 비모수적(nonparametric) 밀도 추정

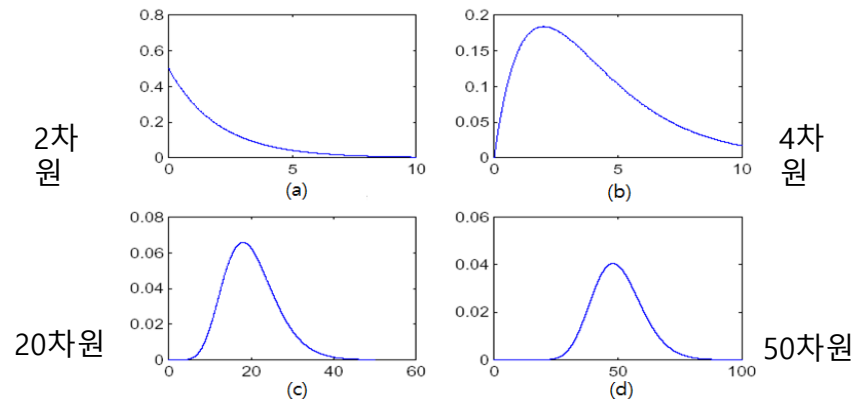
- 분포에 대한 특정 함수를 가정하지 않고, 주어진 데이터를 사용하여 밀도함수의 형태 표현
- 전형적인 형태 : 히스토그램(histogram)



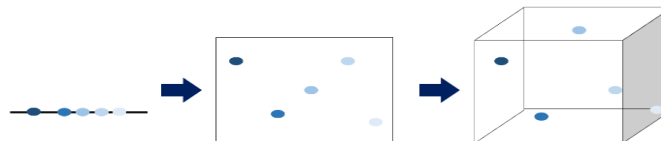
5. Unsupervised Learning (6)

❖ 차원축소(dimension reduction)

- 고차원의 데이터를 정보의 손실을 최소화하면서 저차원으로 변환하는 것
- 2, 3차원으로 변환해 시각화하면 직관적 데이터 분석 가능
- 차원의 저주(curse of dimensionality) 문제 완화
 - 차원이 커질수록 거리분포가 일정해지는 경향



- 차원이 증가함에 따라 부분공간의 개수가 기하급수적으로 증가
- 데이터 간의 거리도 증가, sparse 증가
- the average distance between two points of n dimension: $\sqrt{n/6}$



5. Unsupervised Learning (7)

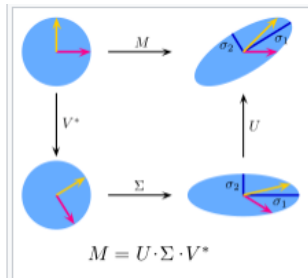
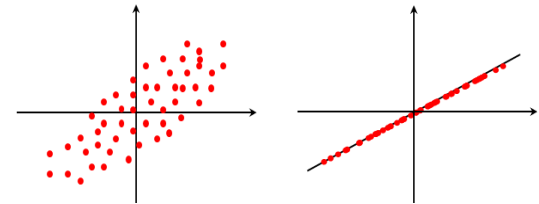
❖ 차원축소(dimension reduction)

▪ 주성분 분석 (Principle Component Analysis, **PCA**)

- 분산이 큰 소수의 축들을 기준으로 데이터를 사상(projection)하여 저차원으로 변환
- 데이터의 **공분산행렬**(covariance matrix)에 대한 **고유값**(eigenvalue)가 큰 소수의 **고유벡터**(eigenvector)를 사상 축으로 선택

▪ $A = U \cdot \Sigma \cdot V^*$

- U : $m \times m$ complex unitary matrix ($U^* \cdot U = U \cdot U^* = I$, $U^* = U^{-1}$)
- Σ : $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal
- V : $n \times m$ complex unitary matrix ($V^*V = V \cdot V^* = I$, $V^* = V^{-1}$)
- eigen vector of $A \cdot A^*$: U 's column
- eigen vector of $A^* \cdot A$: V 's column
- $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$

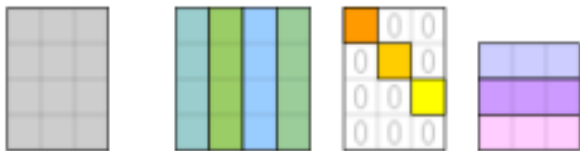


- Illustration of the singular value decomposition $\mathbf{U}\Sigma\mathbf{V}^*$ of a real 2×2 matrix \mathbf{M} .
- **Top:** The action of \mathbf{M} , indicated by its effect on the unit disc D and the two canonical unit vectors e_1 and e_2 .
- **Left:** The action of \mathbf{V}^* , a rotation, on D , e_1 , and e_2 .
- **Bottom:** The action of Σ , a scaling by the singular values σ_1 horizontally and σ_2 vertically.
- **Right:** The action of \mathbf{U} , another rotation.

5. Unsupervised Learning (8)

❖ SVD

$$M = U \Sigma V^*$$



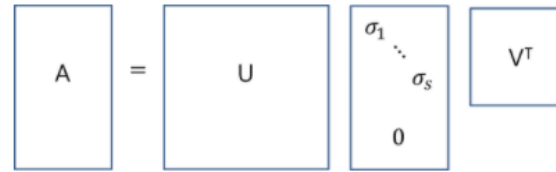
$$\begin{matrix} \mathbf{M} \\ m \times n \end{matrix} = \begin{matrix} \mathbf{U} \\ m \times m \end{matrix} \begin{matrix} \mathbf{\Sigma} \\ m \times n \end{matrix} \begin{matrix} \mathbf{V}^* \\ n \times n \end{matrix}$$



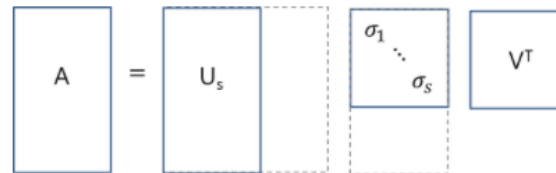
$$\mathbf{U} \mathbf{U}^* = \mathbf{I}_m$$



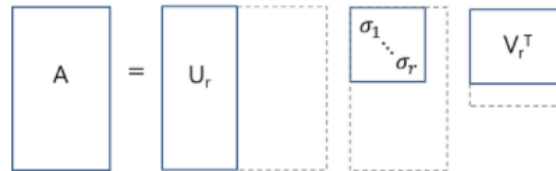
$$\mathbf{V} \mathbf{V}^* = \mathbf{I}_n$$



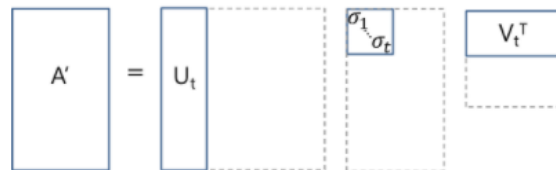
<그림 2> full SVD



<그림 3> thin SVD



<그림 4> Compact SVD



<그림 5> Truncated SVD



<그림 7> 100개의 singular value로 근사 (t = 100)



<그림 8> 50개의 singular value로 근사 (t = 50)

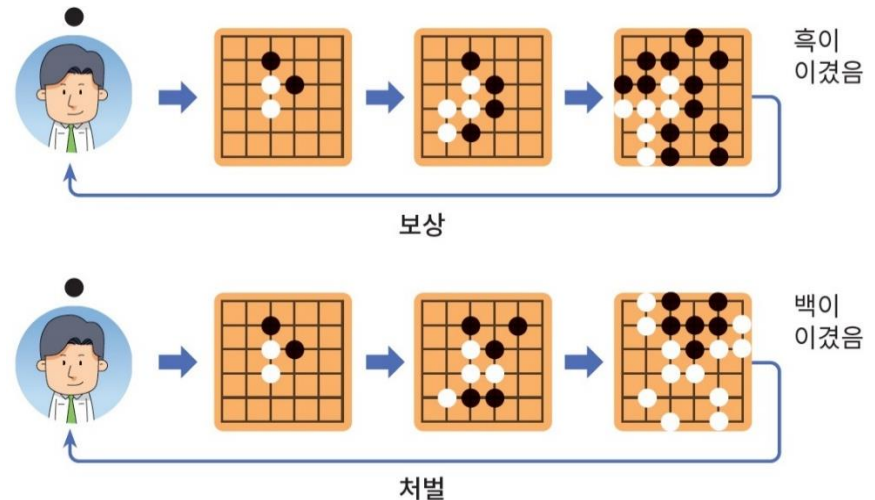
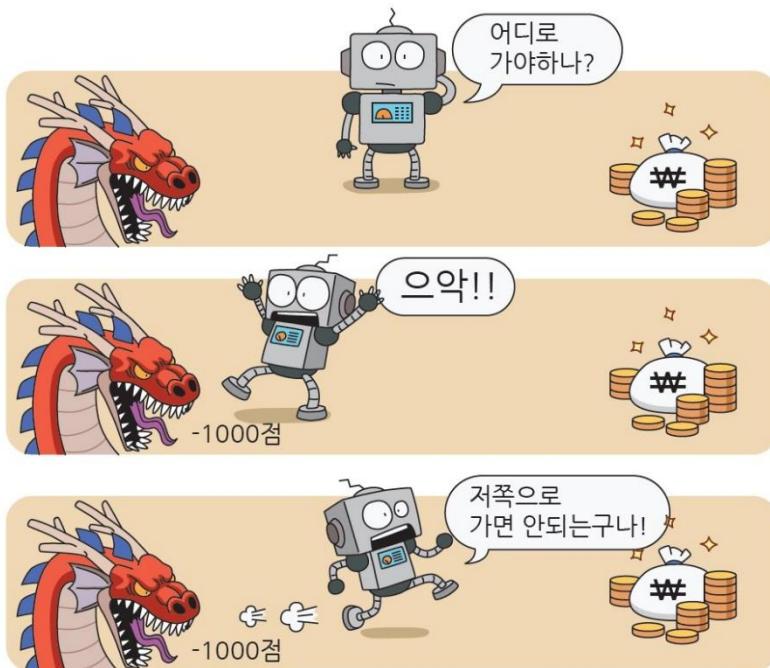


<그림 9> 20개의 singular value로 근사 (t = 20)

6.Reinforcement Learning

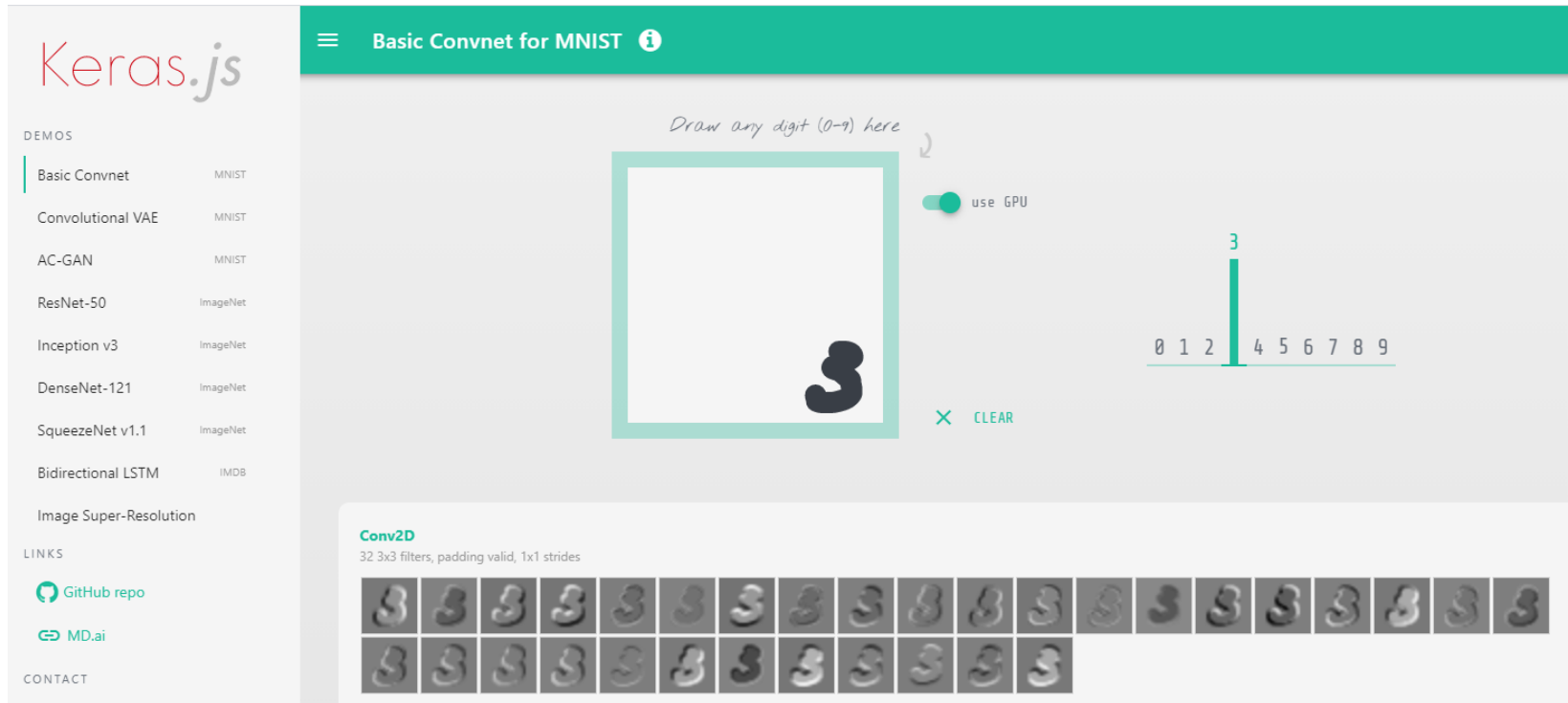
❖ 강화학습 (Reinforcement Learning)

- 컴퓨터가 어떤 행동을 취할 때마다 외부에서 처벌이나 보상이 주어진다.
- 알파고 최종 버전도 강화 학습 사용
- 게임에서 많이 사용된다(예: Frozen Lake).



Lab. Experience of ML

❖ <https://transcranial.github.io/keras-js/#/>



❖ 구글 <https://teachablemachine.withgoogle.com/>

❖ NAVER <https://www.edwith.org/>

Lab. Experience of Neural Network (1)

❖ <https://playground.tensorflow.org> (교재 433쪽)

The screenshot shows the TensorFlow Playground web application. The interface is divided into several sections: DATA, FEATURES, HIDDEN LAYERS, and OUTPUT. Annotations in Korean are provided for various components:

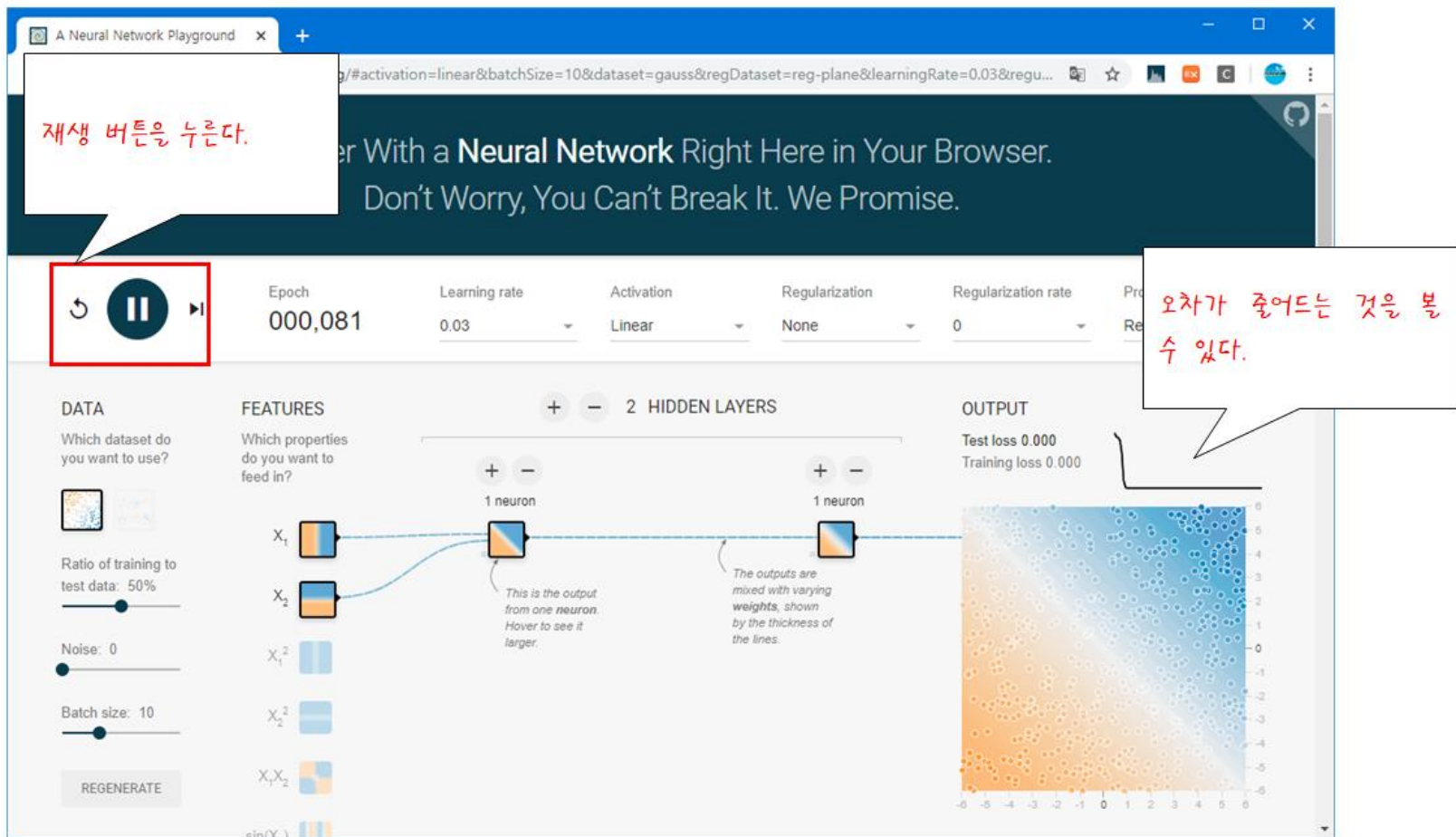
- 데이터 세트를 선택한다.** (Select the dataset.) - Points to the 'Which dataset do you want to use?' dropdown in the DATA section.
- 학습률을 선택한다.** (Select the learning rate.) - Points to the 'Learning rate' dropdown (set to 0.03) in the top configuration bar.
- Linear를 선택한다.** (Select Linear.) - Points to the 'Activation' dropdown (set to Linear) in the top configuration bar.
- Regression을 선택한다.** (Select Regression.) - Points to the 'Problem type' dropdown (set to Regression) in the top configuration bar.
- 버튼을 눌러서 뉴런 하나만 남긴다.** (Press the button to leave only one neuron.) - Points to the '+' and '-' buttons for the first hidden layer, which are currently set to 1 neuron.
- 버튼을 눌러서 뉴런 하나만 남긴다.** (Press the button to leave only one neuron.) - Points to the '+' and '-' buttons for the second hidden layer, which are currently set to 1 neuron.

The OUTPUT section shows a scatter plot of data points and the following loss values:

- Test loss: 0.128
- Training loss: 0.130

Lab. Experience of Neural Network (2)

❖ <https://playground.tensorflow.org> (구글)



7. 수학기표기 (1)

❖ 선형 대수학^{linear algebra}과 미적분, 확률과 통계

수, 벡터, 행렬

표기법	의미	특징 및 주의
a	스칼라 변수	보통 굵기의 이탤릭체
A	스칼라 상수	대문자
\mathbf{a}	벡터	굵은 정자체 소문자
\mathbf{A}	행렬	굵은 정자체 대문자
\mathbf{I}_n	$n \times n$ 차원 항등행렬	아래 첨자로 차원 표기

집합

표기법	의미	특징 및 주의
\mathbb{R}	실수 집합	대문자 외곽선
\mathbb{R}^n	n 차원 실수 벡터 집합	윗 첨자로 벡터의 차원 표기
$\mathbb{R}^{n \times m}$	n 행 m 열의 행렬 집합	윗 첨자로 행과 열의 수 표기
\in	원소	왼쪽이 오른쪽 집합의 원소
$\mathbf{A} \in \mathbb{R}^{n \times m}$	\mathbf{A} 는 n 행 m 열의 행렬	\mathbf{A} 가 $\mathbb{R}^{n \times m}$ 의 원소

7. 수학기표기 (2)

❖ 선형 대수학^{linear algebra}과 미적분, 확률과 통계

선형 대수

표기법	의미	특징 및 주의
A^T	행렬 A 의 전치	첨자 T로 전치 표현
A^{-1}	행렬 A 의 역행렬	첨자 -1로 곱셈에 대한 역원 표현
$\text{tr}(A)$	행렬 A 의 대각성분 합 ^{trace}	함수 표현
$A \otimes B$	아다마르 ^{Hadamard} 곱	연산자에 원을 씌어 원소별 연산 표현

미적분

표기법	의미	특징 및 주의
$f: \mathbb{R}^n \rightarrow \mathbb{R}^m$	함수	\mathbb{R}^n 에 속한 값을 \mathbb{R}^m 으로 옮기는 함수
f'	함수 f 의 미분	어깨점으로 미분 표현
$\frac{df}{dx}$	함수 f 의 x 에 대한 미분	미분의 변수를 명시적으로 표현
$f'(a)$	a 위치에서 함수 f 의 미분	특정 지점에서의 미분, $f: \mathbb{R} \rightarrow \mathbb{R}^*$
$\frac{\partial f}{\partial x_i}$	다차원 입력 함수 f 의 x_i 에 대한 편미분	x_i 제외한 모든 입력 변수를 상수로 취급, $f: \mathbb{R}^n \rightarrow \mathbb{R}^*$
$\frac{\partial f}{\partial x_i}(\mathbf{a})$	입력 벡터 \mathbf{a} 위치에서 함수 f 의 x_i 에 대한 편미분	입력 벡터 \mathbf{a} 위치에서 x_i 를 제외한 모든 변수를 상수로 취급하여 미분
∇f	함수 f 의 기울기	$f: \mathbb{R}^n \rightarrow \mathbb{R}$ 의 기울기
$\nabla_a f$	함수 f 의 기울기	벡터 a 에 대한 함수 f 의 기울기

7. 수학기표기 (3)

❖ 선형 대수학linear algebra과 미적분, 확률과 통계

확률과 통계

표기법	의미	특징 및 주의
$p(X)$	조건 X 가 일어날 확률	X 는 $a = b$ 와 같이 참 또는 거짓인 값
$p(X, Y)$	결합 확률	X 와 Y 가 함께 일어날 확률
$p(Y X)$	조건부 확률	X 가 참일 때, Y 가 일어날 확률
μ	평균	데이터를 대표하는 값
σ, σ^2	표준편차, 분산	$\sigma ()$ 처럼 표현하면 시그모이드 함수

8. Vector와 행렬 (1)

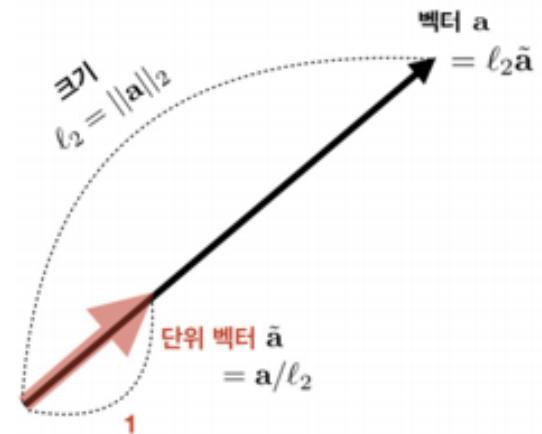
❖ ML에서의 데이터 표현

- 다수의 특징값을 원소로 하는 데이터를 입력으로 제공
- **스칼라** scalar란 1, 2, 3.14,...와 같이 크기 값을 가지는 양
- **벡터** vector는 순서가 있는 스칼라 값의 배열 (열벡터로 표기)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{x}^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^T = [x_1 \ x_2 \ \cdots \ x_n]$$

- n 차원 벡터 \mathbf{x} 는 \mathbb{R}^n $\mathbf{x} \in \mathbb{R}^n$
- 벡터의 크기, **노름** norm $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$
 - $p = 1$, 맨해튼 거리 Manhattan distance, l_1 노름
 - $p = 2$, 유클리드 노름 Euclidian norm, l_2 노름
 - 단위 벡터 unit vector : 길이가 1인 벡터
 - 벡터의 정규화 normalization : $\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$
- 벡터는 크기와 방향

$$\mathbf{x} = \|\mathbf{x}\|_2 \cdot \tilde{\mathbf{x}} = l_2 \times \text{방향}$$

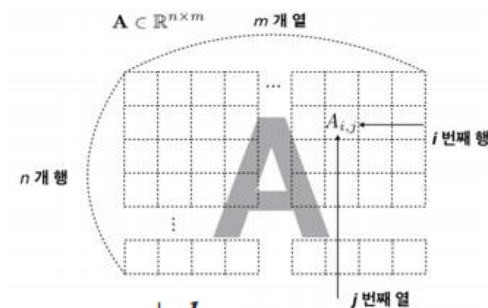


8. Vector와 행렬 (2)

❖ ML에서의 데이터

- **행렬matrix** : 스칼라 값이나 자료를 **행row**과 **열column**로 이루어진 배열
 - 행의 수가 n 이고 열의 수가 m 개인 행렬을 $n \times m$ 행렬

$$A \in \mathbb{R}^{n \times m}$$



❖ 벡터의 기본 연산

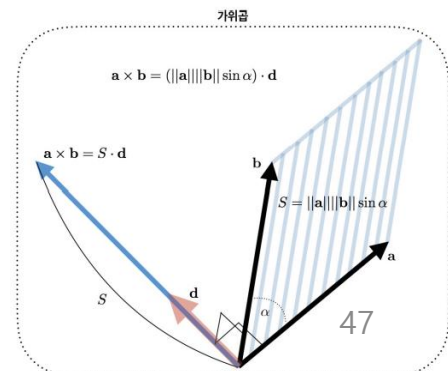
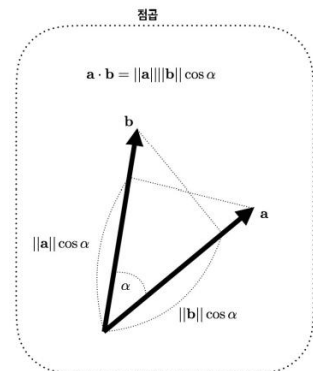
- 덧셈/뺄셈: $\mathbf{c} = \mathbf{a} \pm \mathbf{b} \Leftrightarrow c_i = a_i \pm b_i$
- **아다마르Hadamard 곱**: $\mathbf{c} = \mathbf{a} \otimes \mathbf{b} \Leftrightarrow c_i = a_i b_i$
- 점곱:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n (\mathbf{a} \otimes \mathbf{b})_i$$

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \alpha$$

- **가위곱cross product**: $\mathbf{a} \times \mathbf{b} = (\|\mathbf{a}\| \|\mathbf{b}\| \sin \alpha) \cdot \mathbf{d}$

- 크기는 두 벡터의 크기와 사잇각의 사인^{sine} 값에 비례
- 방향은 두 벡터에 동시에 수직인 방향의 단위 벡터 \mathbf{d}



8. Vector와 행렬 (3)

❖ 행렬의 기본 연산

- 두 행렬 A와 B를 곱할 때는 A의 열의 개수와 B의 행의 개수가 일치

$$\mathbf{A} \in \mathbb{R}^{n \times m} \quad \mathbf{B} \in \mathbb{R}^{m \times l}$$

$$\mathbf{C} = \mathbf{AB} \quad \mathbf{C} \in \mathbb{R}^{n \times l}$$

$$C_{i,j} = \sum_{k=1}^m A_{i,k} B_{k,j} = A_{i,1} B_{1,j} + A_{i,2} B_{2,j} + A_{i,3} B_{3,j} + \cdots + A_{i,m} B_{m,j}$$

- 행렬 A의 i 행 벡터를 $\mathbf{A}_{i,*}$ 라고 하고, B의 j 열 벡터를 $\mathbf{B}_{*,j}$ 라고 하면, 행렬의 곱 C의 i 행 j 열 원소는 두 벡터의 점곱

$$C_{i,j} = \mathbf{A}_{i,*} \mathbf{B}_{*,j} = \mathbf{A}_{i,*}^T \cdot \mathbf{B}_{*,j}$$

$$C_{i,j} = [A_{i,1} \ A_{i,2} \ \cdots \ A_{i,m}]^T \begin{bmatrix} B_{1,j} \\ B_{2,j} \\ \vdots \\ B_{m,j} \end{bmatrix} = \begin{bmatrix} A_{i,1} \\ A_{i,2} \\ \vdots \\ A_{i,m} \end{bmatrix} \cdot \begin{bmatrix} B_{1,j} \\ B_{2,j} \\ \vdots \\ B_{m,j} \end{bmatrix}$$

9. 미분과 기울기, 경사하강법의 개념 (1)

❖ 미분 derivative

$$f'(a) = \lim_{\Delta x \rightarrow 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

- 미분이란 순간 변화량을 구하는 것
- 독립 변수값의 변화량에 대한 함수값 변화량 비의 극한

$\frac{d}{dx}(C) = 0$	$\frac{d}{dx}[Cf(x)] = Cf'(x)$
$\frac{d}{dx}x^n = nx^{n-1}$	$\frac{d}{dx}e^x = e^x$
$\frac{d}{dx}[f(x) \pm g(x)] = f'(x) \pm g'(x)$	$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$

❖ 연쇄법칙 chain rule

- 어떤 함수 y 를 x 에 대해 미분할 때, 매개 변수 t 를 두어 다음과 같이 미분

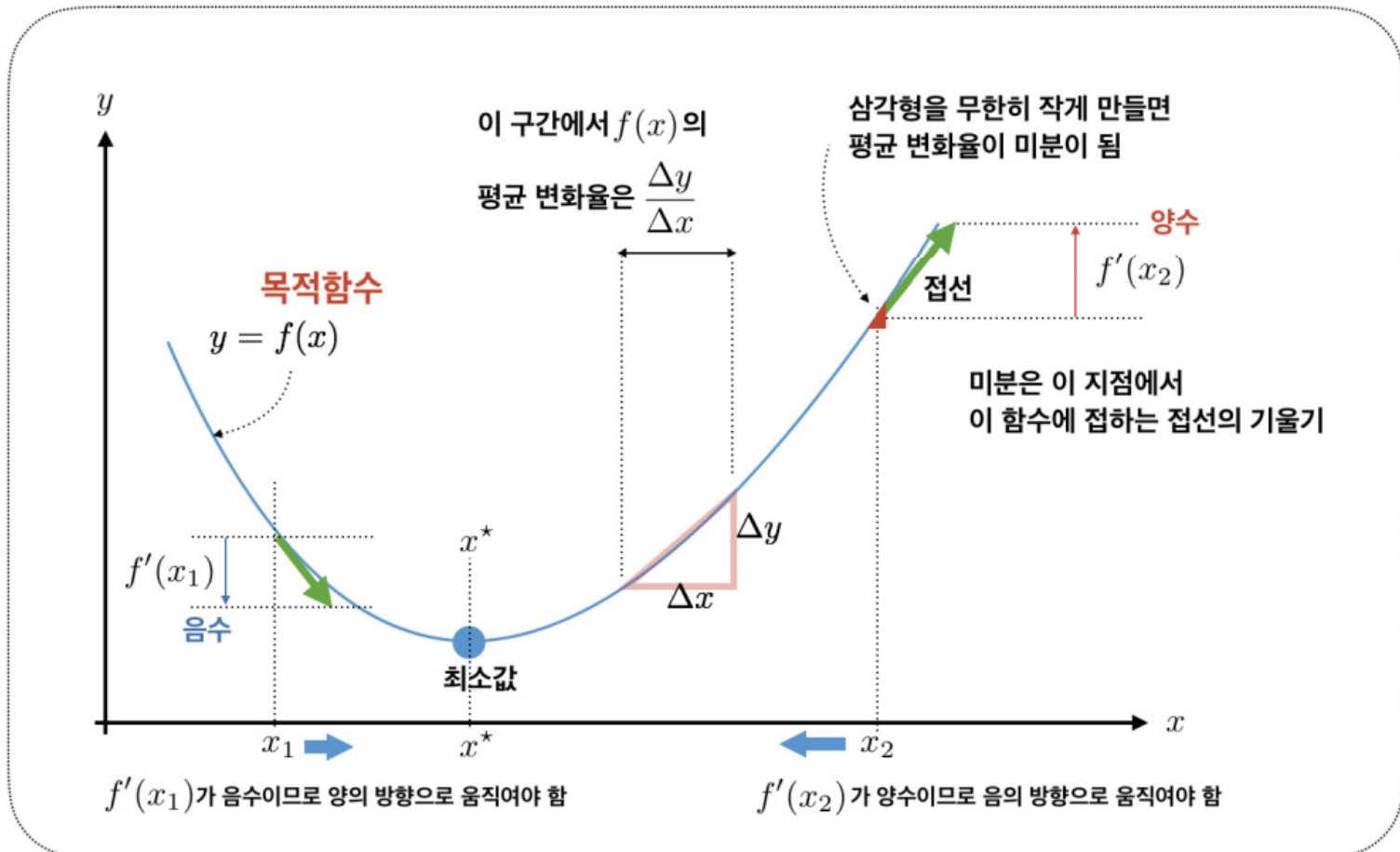
$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx}$$

- $y = f(g(x))$ 와 같은 합성함수에 대한 미분, $t = g(x)$, $y = f(t)$

$$\frac{d}{dx}[f(g(x))] = \frac{d}{dg(x)}f(g(x)) \frac{dg(x)}{dx} = f'(g(x))g'(x)$$

9. 미분과 기울기, 경사하강법의 개념 (2)

- ❖ 평균변화율
- ❖ 미분과 접선의 기울기(순간변화율)
- ❖ 경사 하강법 gradient descent



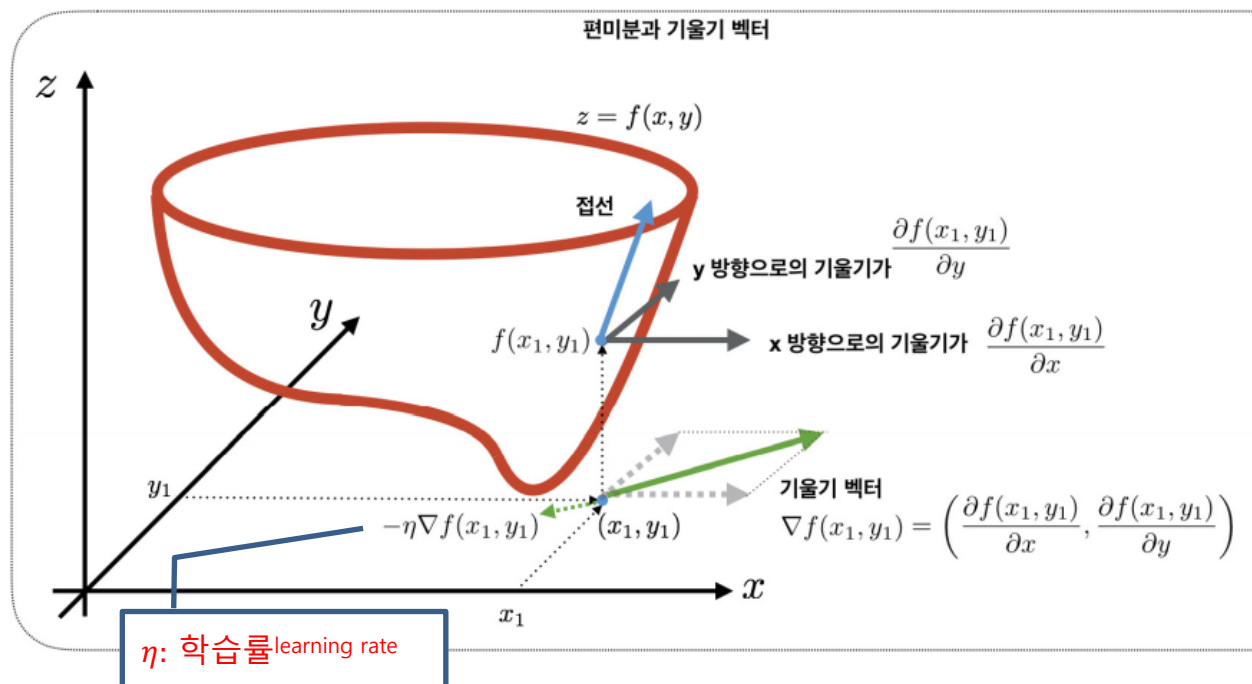
9. 미분과 기울기, 경사하강법의 개념 (3)

❖ 편미분 partial derivative

- 둘 이상의 변수들을 가지는 함수 f 가 있을 경우, 이 함수를 각각의 변수에 대해서 독립적으로 미분을 하는 방식
- 예) $f(x, y) = x^2 + xy + y^2$

$$\frac{\partial f}{\partial x}(x, y) = 2x + y, \quad \frac{\partial f}{\partial y}(x, y) = x + 2y$$

- 편미분은 다차원 공간에서 정의되는 목적함수의 최적해를 찾기 위함



9. 미분과 기울기, 경사하강법의 개념 (4)

❖ 편미분 partial derivative

- $\partial f(x_1, y_1) / \partial x$: 목적함수의 $f(x_1, y_1)$ 에 닿는 접선이 x 축 방향 기울기
- $\partial f(x_1, y_1) / \partial y$: y 축 방향 기울기
- $x \in \mathbb{R}^n$ 의 벡터를 입력으로 하는 함수 $f(x)$ 의 기울기 벡터 $\nabla f(x)$ 는 모든 차원의 기울기를 원소로 하는 벡터

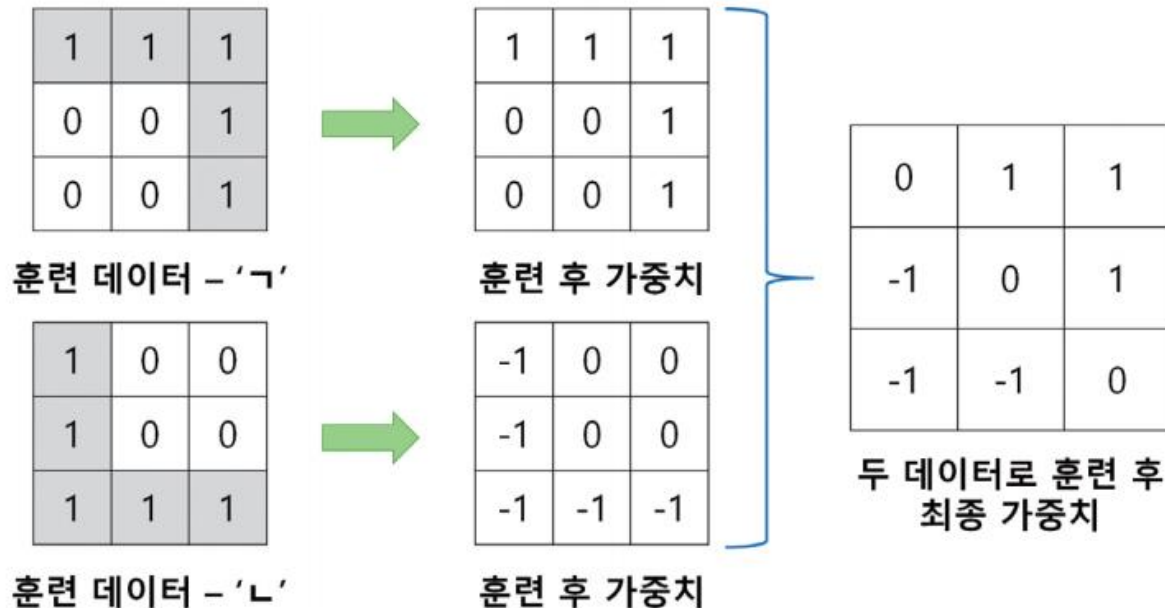
$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)$$

❖ 학습률 learning rate η (eta)

- 기울기 벡터의 크기를 얼마나 고려하여 이동할지를 결정하는 값

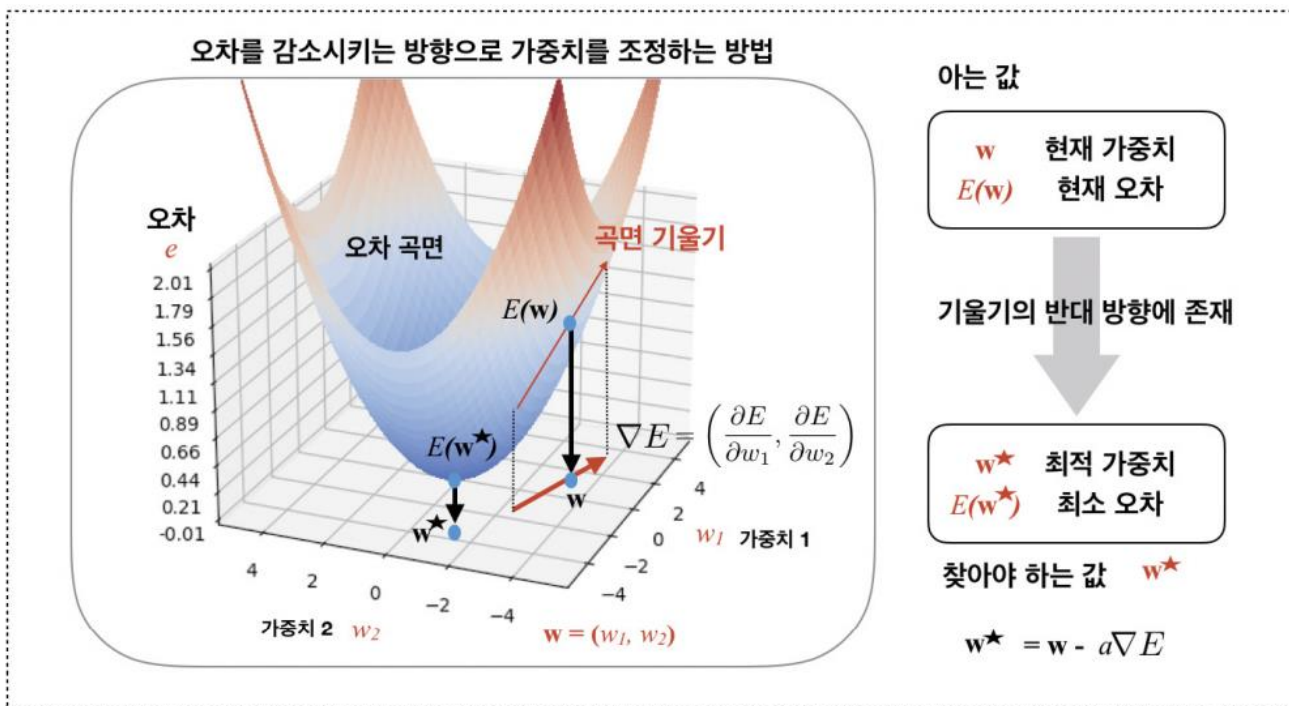
10. 모델, 파라미터, 그리고 학습 (1)

- ❖ 모델^{model} : 문제를 해결하는 방법
- ❖ 파라미터^{parameter} : 모델의 동작을 결정
- ❖ 학습^{learning} : 파라미터를 더 좋은 상태로 변경하는 과정



10. 모델, 파라미터, 그리고 학습 (2)

- ❖ 오차의 기울기를 이용한 학습의 기본 원리와 최적해
 - 모델은 현재의 파라미터를 바탕으로 어떤 행위를 하고 그 결과는 정답과 차이(오차^{error})가 남
 - 오차가 없다 = 학습이 완벽하게 잘 되었다 = 모델이 데이터를 잘 설명한다고 볼 수 있다
 - 학습이란 이 오차가 줄어드는 방향(모델이 데이터를 잘 설명하는 방향)으로 파라미터를 변경하는 일



10. 모델, 파라미터, 그리고 학습 (3)

❖ 지역 최소값^{local minimum}, 전역 최소값^{global minimum}

