

Machine Learning

Machine Learning

- **Module I** : Machine Learning
- **Module II** : Neural Network
- **Module III** : 활용

Module I .

Machine Learning

- 1 Introduction
- 2 지도 학습 (Supervised Learning)
- 3 비지도 학습 (Unsupervised Learning)

1. Introduction

1. 머신 러닝

사람처럼 할 수 있을까?

분야 별 제각각 다른 로직을 작성하는 것이 아니라,
배우면서 점점 잘하게 되는 만능 알고리즘?

- 엔지니어가 일일이 튜닝 하지 않아도,
- 수 많은 데이터들을 이용해,
- 내부 동작을 스스로 구성해가며,
- 기존에 고안했던 방법론 이상의 성능을 내었으면..



➤ 대량의 데이터에서 **특징을 추출**해서 연관관계 분류 기준을 **스스로 학습** !

1. Introduction

2. 알고리즘 종류

<u>Machine Learning Types</u>	<u>Tasks</u>	<u>Analysis methods/Algorithms</u>
지도학습 (Supervised Learning)	예측, 추정 (Prediction, Estimation)	Linear Regression
	분류 (Classification)	k-NN(k-Nearest Neighbor) Decision Tree Naïve Bayes Classification Logistic Regression SVM Ensemble (Bagging, Boosting, Random Forest)
비지도학습 (Unsupervised Learning)	그룹화 (Grouping)	k-Means Clustering k-Medoids Density-based Clustering
	차원 축소 (Dimension Reduction)	PCA(Principal Component Analysis) SVD(Singular Value Decomposition)

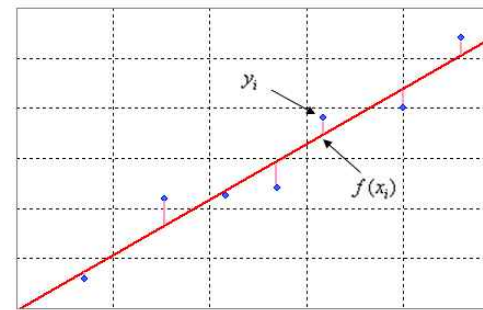
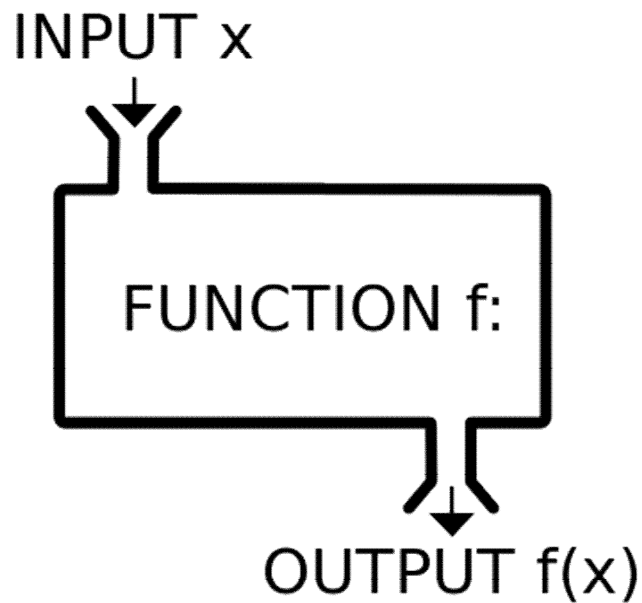
2. 지도 학습 (Supervised Learning)

1. 예측

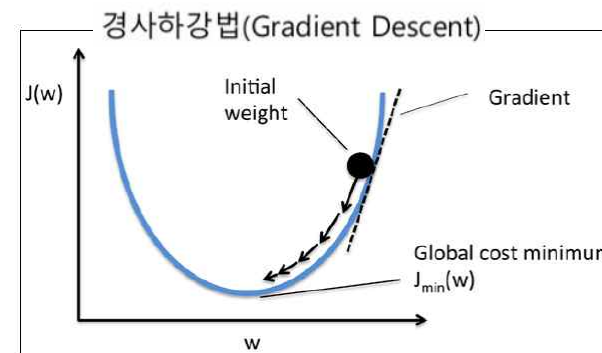
❖ 선형 회귀 (Linear regression)

키와 몸무게의 관계? 기온과 아이스크림 판매량의 관계? 예약대수로 판매량 예측?

- 두 수간의 인과관계를 조사하는 것



최소자승법 $\sum_{i=1}^n (y_i - f(x_i))^2$ 이 최소가 되도록 하는 함수 $f(x)$ 를 구하는 것



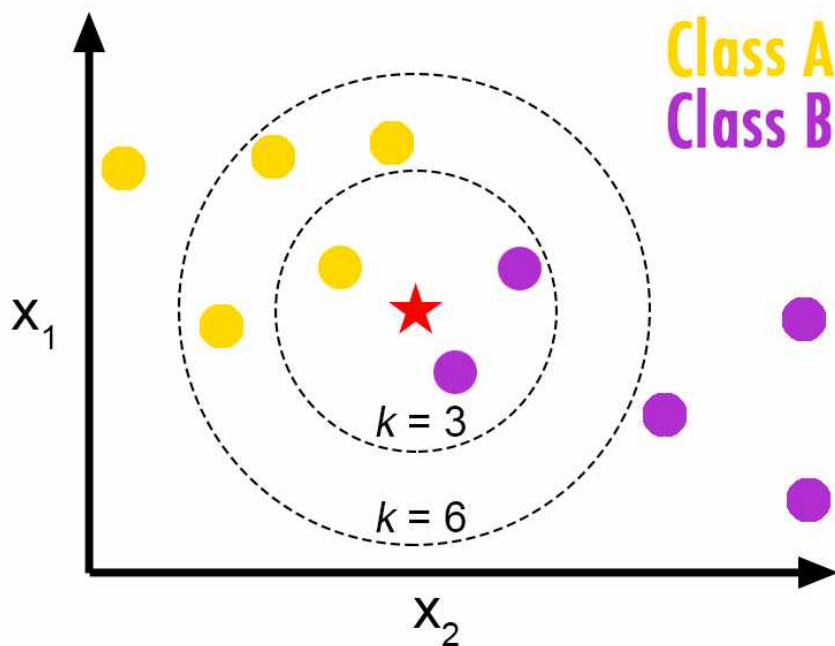
2. 지도 학습 (Supervised Learning)

2. 분류

❖ 최 근접 이웃 (kNN, k-Nearest Neighbors)

토마토는 야채일까? 과일일까?

- 선호도 예측 및 추천 시스템에서 사용



$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

유클리디안 거리(Euclidean distance)

- ① k의 값을 정해줌.
- ② 제일 가까운 거리에 있는 k개의 분류 값 참조.
(명목 형: 다수결, 수치 형: 평균값)

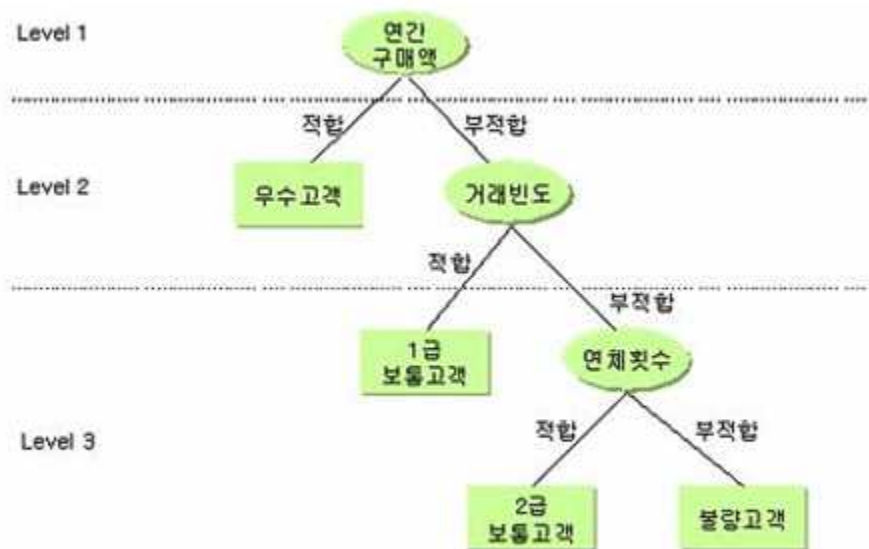
2. 지도 학습 (Supervised Learning)

2. 분류

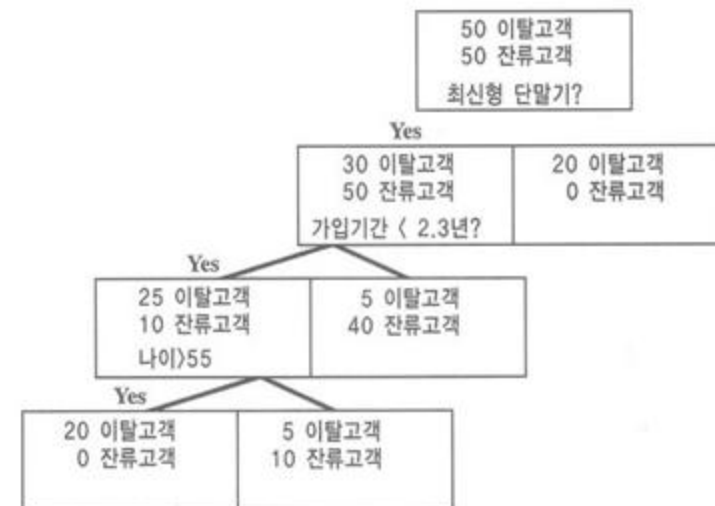
❖ 의사결정트리(Decision Trees)

불량 고객 분석? 더 나아가 불량 고객 예측 ? 고객 이탈 예측도 !

- 분석과정을 이해하고 설명이 필요한 경우 및 CRM에 활용.



[고객 분석]

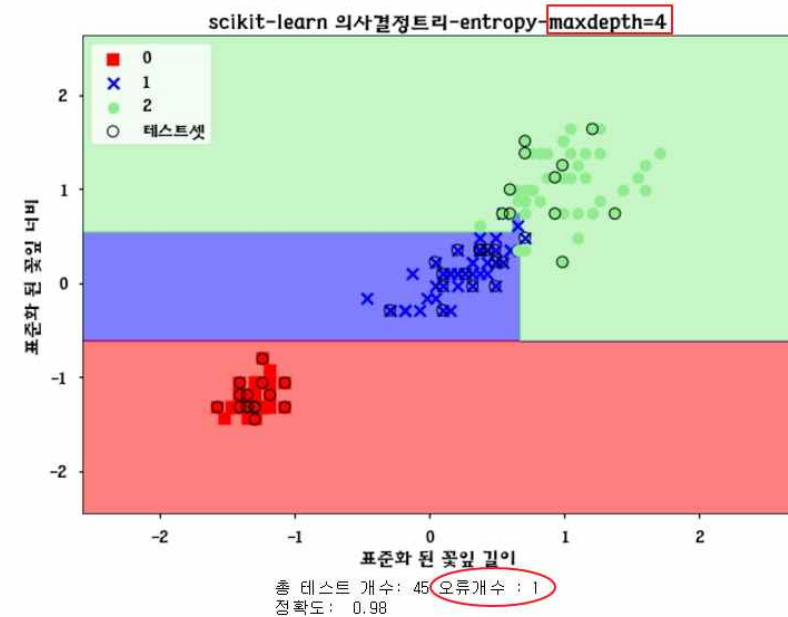
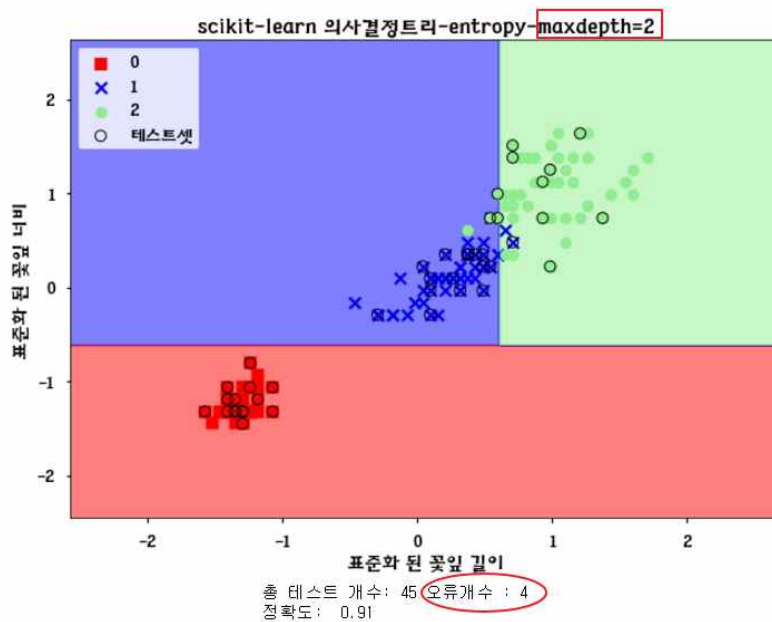


2. 지도 학습 (Supervised Learning)

2. 분류

❖ 의사결정트리(Decision Trees – python)

- Decision Trees의 문제점



모의고사는 열심히 풀었는데, 수능을 망하는 상황 !

2. 지도 학습 (Supervised Learning)

2. 분류

❖ 나이브 베이즈(Naive Bayes)

스팸 일까? 정상 메일일까? 이 뉴스의 카테고리는 뭘까?

- 텍스트 분류에 주로 사용.

① 데이터 전처리 필요

② 데이터의 모든 단어들로 **카테고리 별 점수 계산**

③ 점수가 가장 높은 것을 옳은 분류로 결정

	구성 요소	클래스
학습 벡터 1	function, class, struct, int	c/c++
학습 벡터 2	method, class, int	java
학습 벡터 3	pointer, struct, int, float	c/c++
학습 벡터 4	final, int, float	java
학습 벡터 5	string, array, synchronized	java
입력 벡터(I)	It has struct, int, float.	???

$$\frac{P(e_1, e_2, \dots, e_M | c_i) P(c_i)}{P(\mathcal{D})} \approx P(e_1 | c_i) P(e_2 | c_i) \dots P(e_M | c_i) P(c_i)$$

$$\begin{aligned}
 \log(P(I|C_{c/c++})) &= \log(P(it|C_{c/c++})) + \log(P(has|C_{c/c++})) + \log(P(struct|C_{c/c++})) + \log(P(int|C_{c/c++})) + \log(P(float|C_{c/c++})) + \log(P(C_{c/c++})) \\
 &= \log\left(\frac{0+1}{8+14}\right) + \log\left(\frac{0+1}{8+14}\right) + \log\left(\frac{2+1}{8+14}\right) + \log\left(\frac{2+1}{8+14}\right) + \log\left(\frac{1+1}{8+14}\right) + \log\left(\frac{2}{5}\right) \\
 &\approx -5.855 \\
 \log(P(I|C_{java})) &= \log(P(it|C_{java})) + \log(P(has|C_{java})) + \log(P(struct|C_{java})) + \log(P(int|C_{java})) + \log(P(float|C_{java})) + \log(P(C_{java})) \\
 &= \log\left(\frac{0+1}{9+14}\right) + \log\left(\frac{0+1}{9+14}\right) + \log\left(\frac{0+1}{9+14}\right) + \log\left(\frac{2+1}{9+14}\right) + \log\left(\frac{1+1}{9+14}\right) + \log\left(\frac{3}{5}\right) \\
 &\approx -6.252
 \end{aligned}$$

2. 지도 학습 (Supervised Learning)

2. 분류

❖ 나이브 베이즈(Naive Bayes – java)

```
public static void main(String[] args){
    String stopListFilePath = "D:\\DEV\\workspace\\Chapter10\\data\\ex6DataEmails\\en.txt";
    String dataFolderPath = "D:\\DEV\\workspace\\Chapter10\\data\\ex6DataEmails\\train"; // learning data
    String testFolderPath = "D:\\DEV\\workspace\\Chapter10\\data\\ex6DataEmails\\test"; // test data

    ArrayList<Pipe> pipeList = new ArrayList<Pipe>();
    pipeList.add(new Input2CharSequence("UTF-8"));
    Pattern tokenPattern = Pattern.compile("[\\p{L}\\p{N}_]+");
    pipeList.add(new CharSequence2TokenSequence(tokenPattern));
    pipeList.add(new TokenSequenceLowercase());
    pipeList.add(new TokenSequenceRemoveStopwords(new File(stopListFilePath), "utf-8", false, false, false));
    pipeList.add(new TokenSequence2FeatureSequence());
    pipeList.add(new FeatureSequence2FeatureVector());
    pipeList.add(new Target2Label());
    SerialPipes pipeline = new SerialPipes(pipeList);

    FileIterator folderIterator = new FileIterator(
        new File[] {new File(dataFolderPath)},
        new TxtFilter(),
        FileIterator.LAST_DIRECTORY); //클래스 레벨에 마지막 디렉토리 이름을 적용하도록 함.

    InstanceList instances = new InstanceList(pipeline);
    instances.addThruPipe(folderIterator);

    ClassifierTrainer classifierTrainer = new NaiveBayesTrainer();
    Classifier classifier = classifierTrainer.train(instances);
    Trial trial = new Trial(classifier, testInstances);
    In:
    for System.out.println(
        "F1 for class '" + classifier.getLabelAlphabet().lookupLabel(1)+"': " + trial.getF1(classifier.getLabelAlphabet().lookupLabel(1));
    System.out.println(
        "Precision: " + trial.getPrecision(1));
    tes:
    System.out.println(
    Tri: "Recall: "+trial.getRecall(1));

    System.out.println(
        "F1 for class '" + classifier.getLabelAlphabet().lookupLabel(1)+"': " + trial.getF1(classifier.getLabelAlphabet().lookupLabel(1));
    System.out.println(
        "Precision: " + trial.getPrecision(1));
    System.out.println(
        "Recall: "+trial.getRecall(1));
```

F1 for class 'spam':0.9731800766283524
Precision: 0.9694656488549618
Recall: 0.9769230769230769

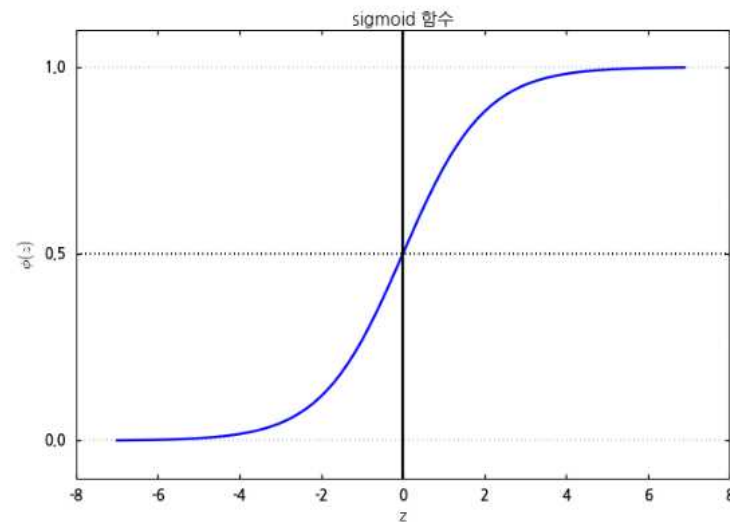
2. 지도 학습 (Supervised Learning)

2. 분류

❖ 로지스틱 회귀 (Logistic regression)

5억을 가지고 있으면 부자인가 아닌가?

- 독립변수와 종속변수 간 이진 분류에 탁월
(예/아니오, 1/0, 합격/불합격)



$$\hat{y} = \frac{1}{1 + e^{-(w \times x + b)}} = \frac{1}{1 + e^{-z}}$$

$$z = w \times x + b$$

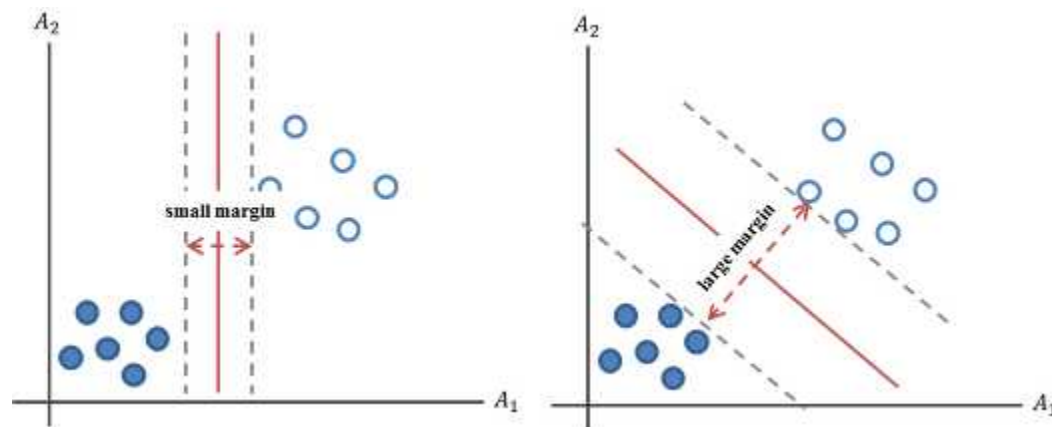
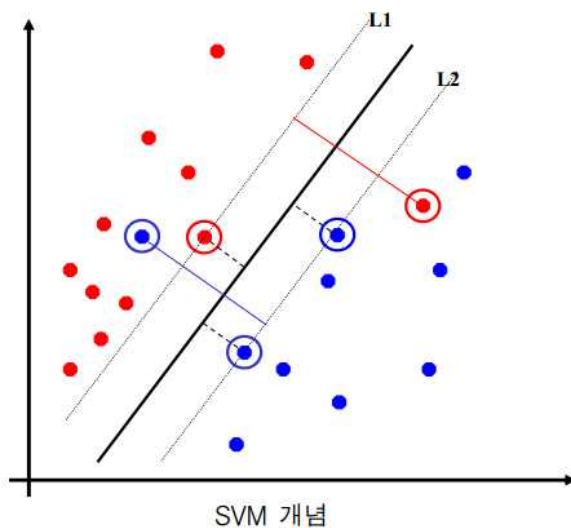
2. 지도 학습 (Supervised Learning)

2. 분류

❖ SVM(Support Vector Machine)

상품명으로 카테고리 분류? 문서 분류?

- 주어진 많은 데이터들을 가능한 멀리 두 개의 집단으로 분리시키는 최적의 초평면을 찾는 방식
- 선형/비선형 (고차원 공간으로 사상 - 커널 트릭)

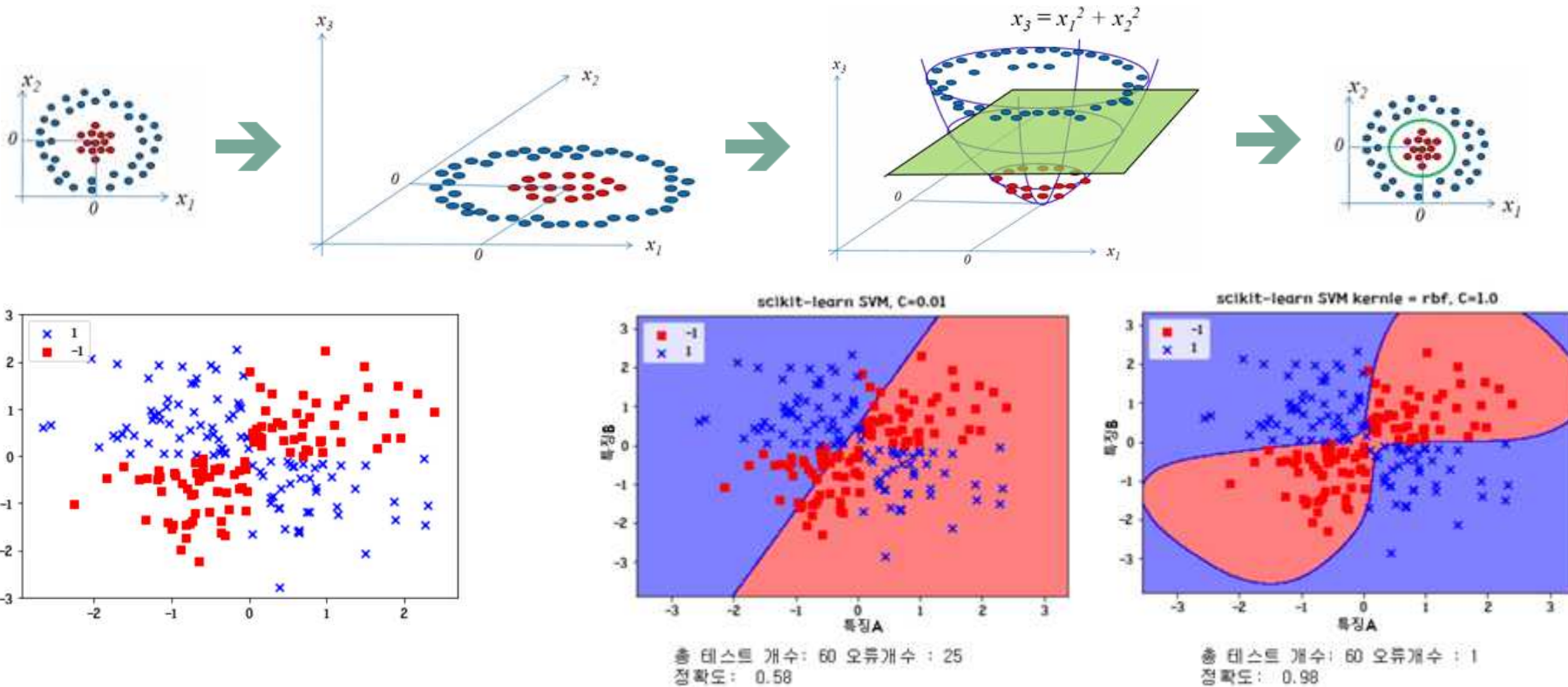


2. 지도 학습 (Supervised Learning)

2. 분류

❖ SVM(Support Vector Machine) – python

- 커널 트릭을 사용하는 SVM

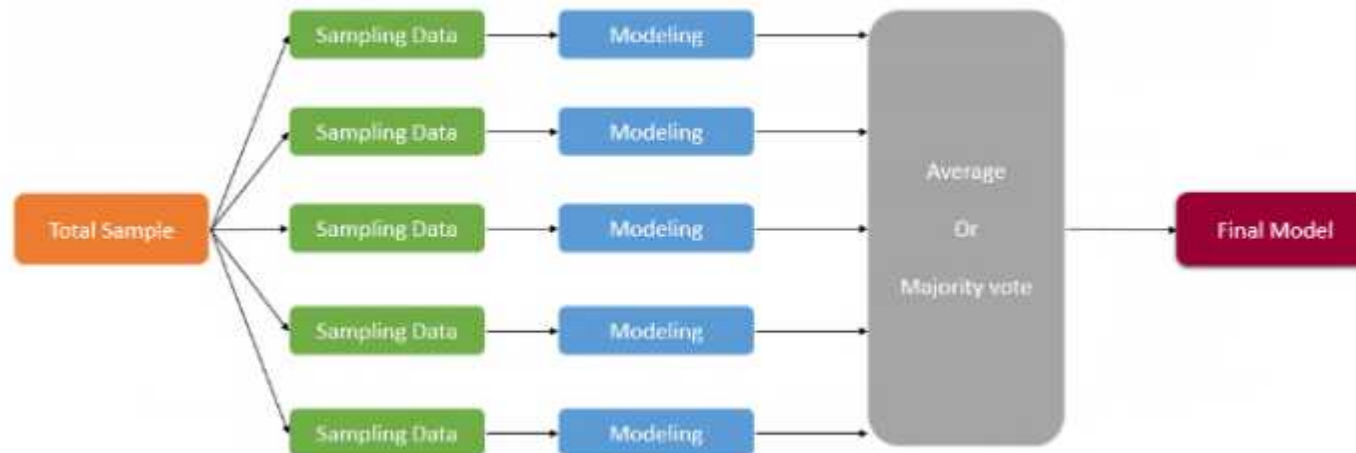


2. 지도 학습 (Supervised Learning)

2. 분류 - Ensemble learning

❖ Bagging(Bootstrap AGGREGatING)

- 샘플 데이터 중 데이터를 랜덤하게 뽑아가며 여러 개의 Weak classifier를 training 함. 테스트 데이터가 오면 Weak classifier의 결과 값 중 많이 나온 값으로 분류.

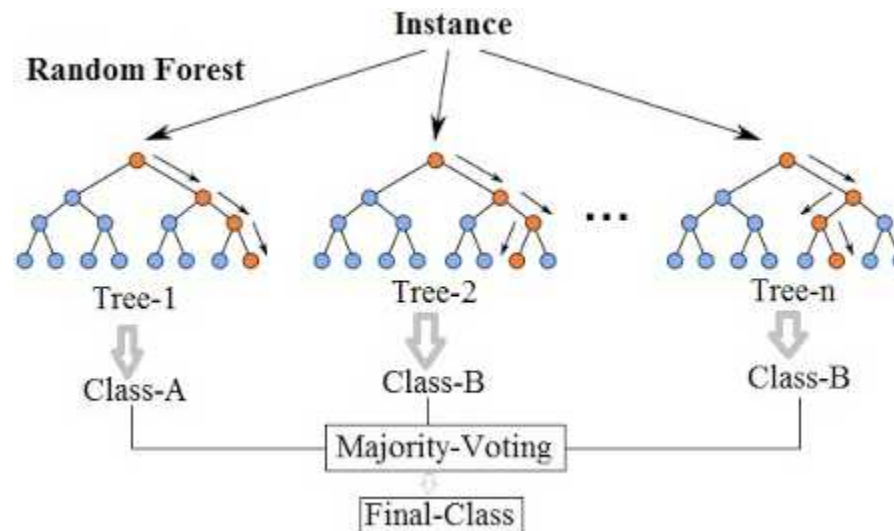


2. 지도 학습 (Supervised Learning)

2. 분류 - Ensemble learning

❖ Random Forest

- Decision Tree(Overfitting개선) + Bagging => Random Forest

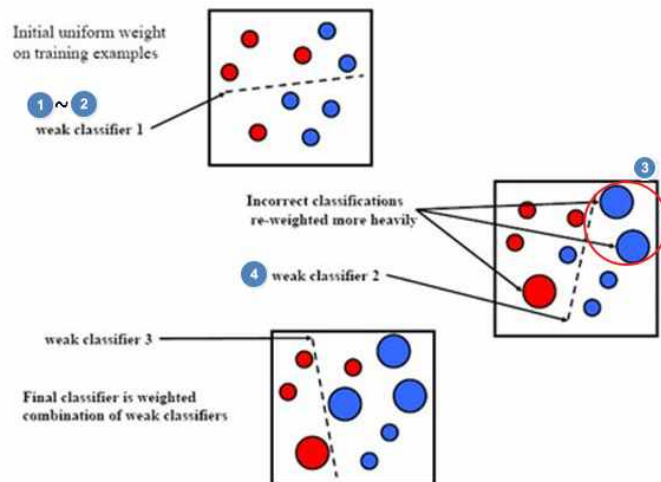


2. 지도 학습 (Supervised Learning)

2. 분류 - Ensemble learning

❖ AdaBoost

자동차 번호판 인식? 핸드폰 카메라 얼굴 인식?



$$H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$

AdaBoost의 학습 과정.

① weak classifier 학습.

② 오류 값을 이용하여 weak classifier의 신뢰도를 구함.

③ 틀린 data sample은 다음 weak classifier가 더 잘 구분할 수 있도록 weight 높임.

④ 반복

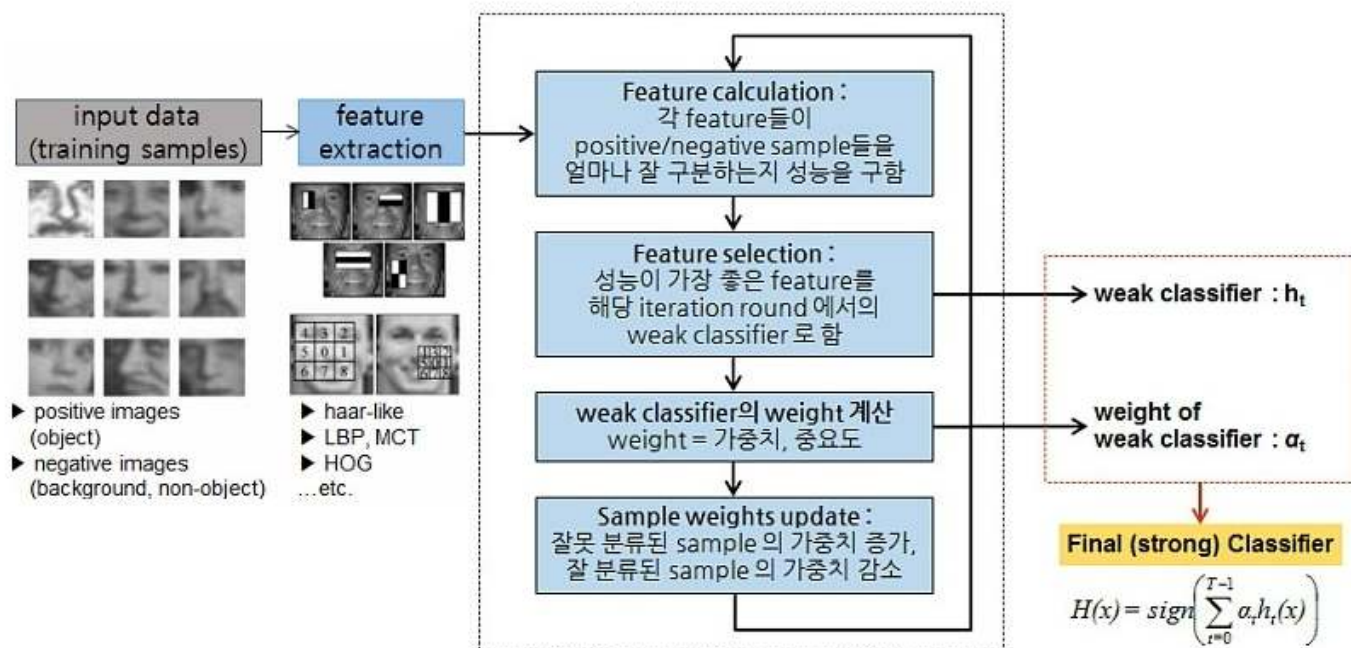
→ 데이터가 들어오면 각 weak classifier의 신뢰도에 따라 투표권을 갖고, 투표결과에 따라 결정.

2. 지도 학습 (Supervised Learning)

2. 분류 - Ensemble learning

❖ AdaBoost

- 얼굴 검출 활용 예



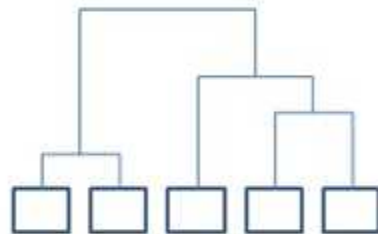
3. 비지도 학습 (Unsupervised Learning)

1. 군집(Clustering)

❖ 군집 (Clustering)

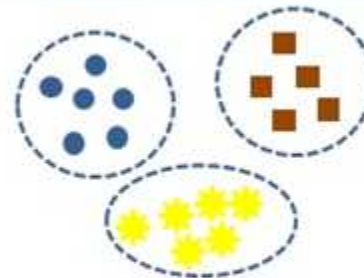
- 데이터의 구조를 이해, 파악하여 활용하는 목적으로 분석 초기단계에서 사용.
- 문서 검색, 구조 분류 등

계층적 군집 (Hierarchical Clustering)



- 병합적 (상향식) 군집
- 분할적 (하향식) 군집

분할적 군집 (Partitional Clustering)



- K-means
- K-medoids
- DBSCAN

- 마케팅 분야 : 고객에 대한 소비성향 및 특징 분석 시 사용.
- 기타 : 이상 거래 탐지, 명확한 기준이없는 문서 분류, 세분화 작업

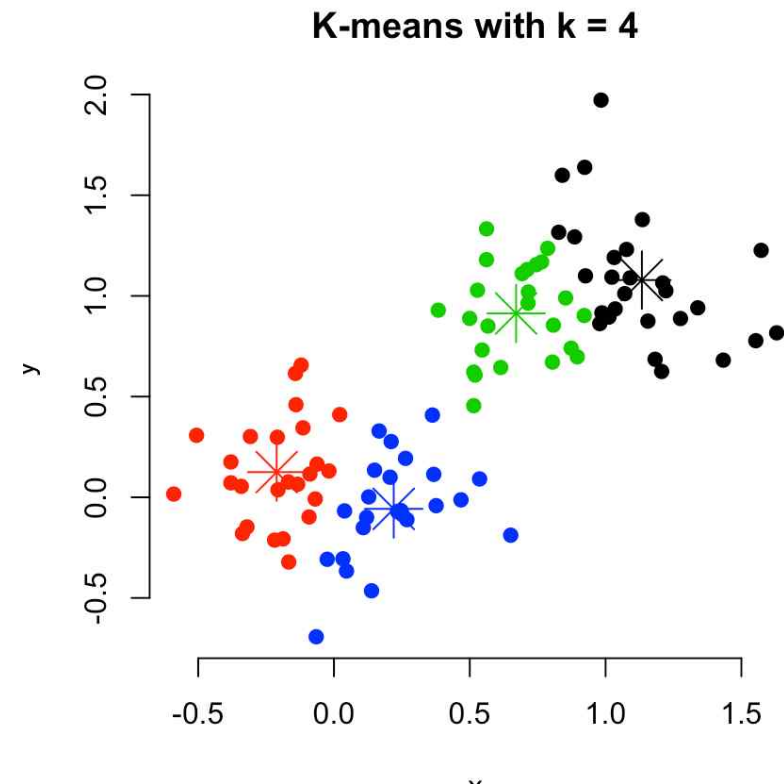
3. 비지도 학습 (Unsupervised Learning)

2. 비 계층 군집(Clustering)

❖ K-means clustering

- 동떨어져있는 noise에 민감함
- 초기 중심 선택에 따라 결과가 다를 수 있음
- 오목한 형태의 군집모델은 문제.

- ① 분류할 클러스터 수(k)정함
- ② k개의 가상의 초기 중심 정해짐.
- ③ 각 데이터들은 제일 가까운 초기 중심으로부터의 거리를 계산하여 군집의 중심을 최적의 위치로 옮김.
- ④ 데이터들의 소속이 바뀌지 않을 때까지 반복

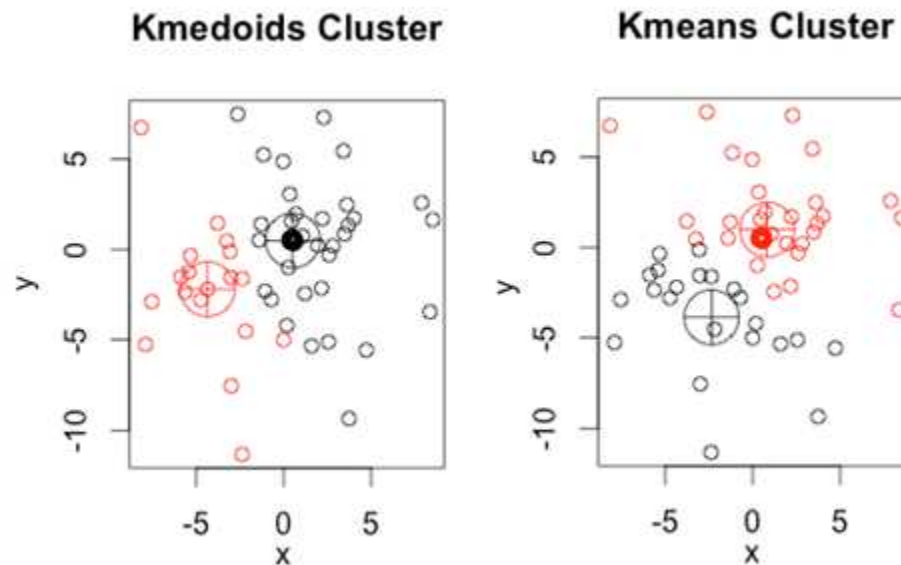


3. 비지도 학습 (Unsupervised Learning)

2. 비 계층 군집(Clustering)

❖ K-medoids

- k-means 단점을 개선한 알고리즘.
(클러스터 중심을 좌표표면상 임의의 점이 아닌 데이터 세트 값 중 하나 선택.)
- noise 처리에 강하나, 계산량이 많음
- 오목한 형태의 군집모델에는 문제.

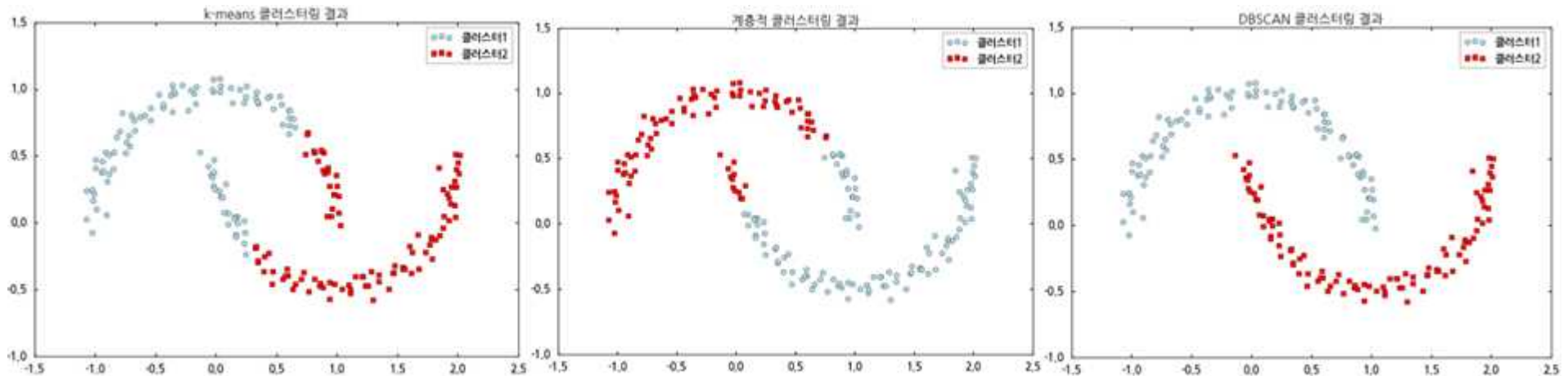


3. 비지도 학습 (Unsupervised Learning)

2. 비 계층 군집(Clustering)

❖ DBSCAN Clustering

- 일정한 밀도를 가지는 데이터의 무리는 거리와 관계없이 같은 클러스터로 판단.
- K-means와 달리 군집 개수를 미리 정해줄 필요가 없음.
- 다차원, 고밀도 데이터는 군집화가 어려움.

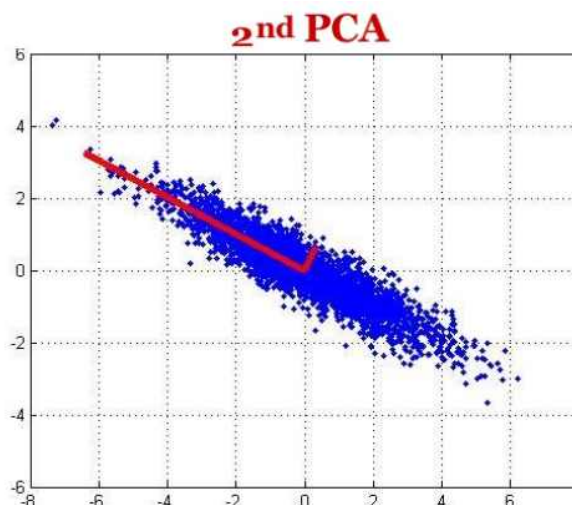
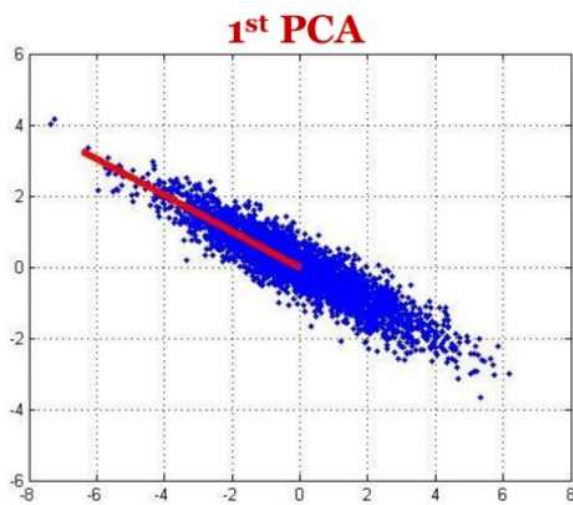


3. 비지도 학습 (Unsupervised Learning)

4. 차원 축소(Dimension Reduction)

❖ 주성분분석(PCA, Principal Component Analysis)

- 고차원의 데이터 → 저 차원데이터로 압축
- 데이터의 분포를 가장 잘 표하는 성분을 찾아, 주된 특성만 살리는 방향으로 압축.
- 지도학습 전 데이터 처리에 주로 활용.



d=1



d=4



d=16



d=100



Original Image

3. 비지도 학습 (Unsupervised Learning)

4. 차원 축소(Dimension Reduction)

❖ 잠재 디리클레 할당 (LDA, Latent Dirichlet Allocation)

대량의 문서에서 주제를 찾아 주는 차원 축소 기법 !

- 토픽의 개수(k) 를 정해주면, k개에 대해 분류. (토픽 모델링)
- 한국어는 형태소 분석기로 명사와 동사만 걸러낸 후 LDA 사용.

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

3. 비지도 학습 (Unsupervised Learning)

4. 차원 축소(Dimension Reduction)

❖ 잠재 디리클레 할당 (LDA – java)

```
public static void main(String[] args) throws Exception {
    String dataFolderPath = "D:\\DEV\\workspace\\Chapter10\\data\\bbc";
    String stopListFilePath = "D:\\DEV\\workspace\\Chapter10\\data\\bbc\\en.txt";

    ArrayList<Pipe> pipeList = new ArrayList<Pipe>();
    pipeList.add(new Input2CharSequence("UTF-8"));
    Pattern tokenPattern = Pattern.compile("[\\p{L}\\p{N}_]+");
    pipeList.add(new CharSequence2TokenSequence(tokenPattern));
    pipeList.add(new TokenSequenceLowercase());
    pipeList.add(new TokenSequenceRemoveStopwords(new File(stopListFilePath), "utf-8", false, false, false));
    pipeList.add(new TokenSequence2FeatureSequence());
    pipeList.add(new Target2Label());
    SerialPipes pipeline = new SerialPipes(pipeList);

    FileIterator folderIterator = new FileIterator(
        new File[] {new File(dataFolderPath)},
        new TxtFilter(),
        FileIterator.LAST_DIRECTORY);

    InstanceList instances = new InstanceList(pipeline);

    instances.addThruPipe(folderIterator);

    int numTopics = 5;
    ParallelTopicModel model =
        new ParallelTopicModel(numTopics, 0.01, 0.01); //LDA 잠재 디리클레 할당

    model.addInstances(instances); // 모델에 인스턴스 추가
    model.setNumThreads(4); // 여러개의 스레드가 병렬 처리 되도록 설정.

    model.setNumIterations(1000); // 반복 실행 횟수 지정 (예측력 높아짐)
    model.estimate(); // LDA 모델 생성 완료
}
```

[토픽 모델링 결과]

```
0 0.08646 year company market growth economy firm
1 0.11085 government people labour election party blair
2 0.06756 game england year time win world
3 0.0575 film year music show awards award
4 0.05845 people mobile technology games users digital
```

business
politics
sport
entertainment
tech

3. 비지도 학습 (Unsupervised Learning)

5. Recommendation



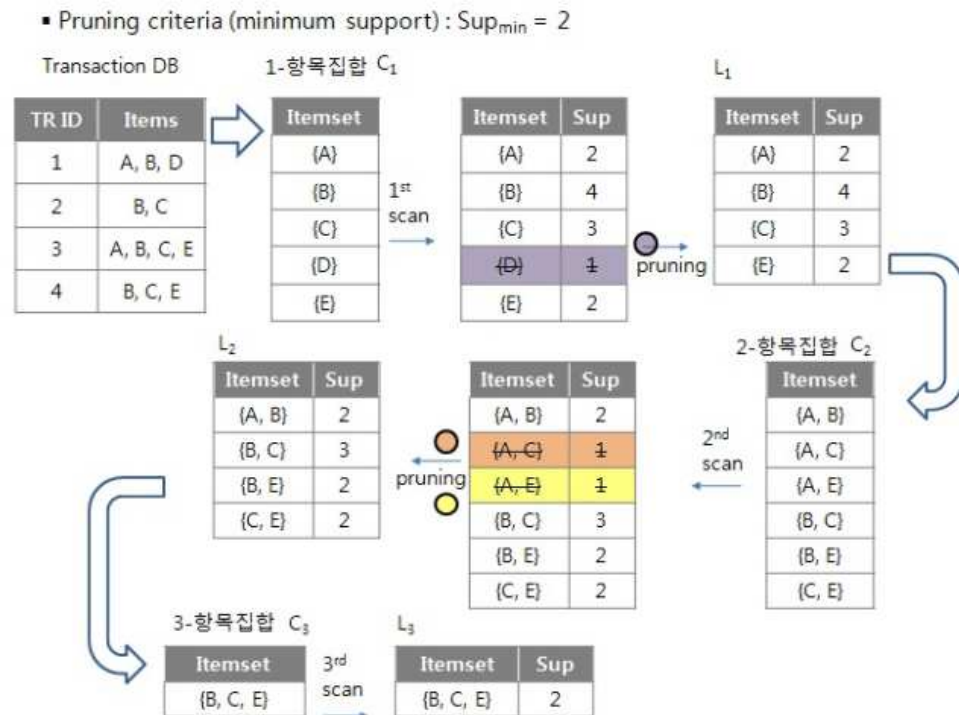
3. 비지도 학습 (Unsupervised Learning)

5. Recommendation

❖ 연관 규칙 - Apriori Algorithm

시리얼을 구매할 때, 우유를 같이 살 확률이 90% ?

- 동시 발생 빈도 기반
- 패턴의 길이를 점점 늘리며
min Support조건 확인

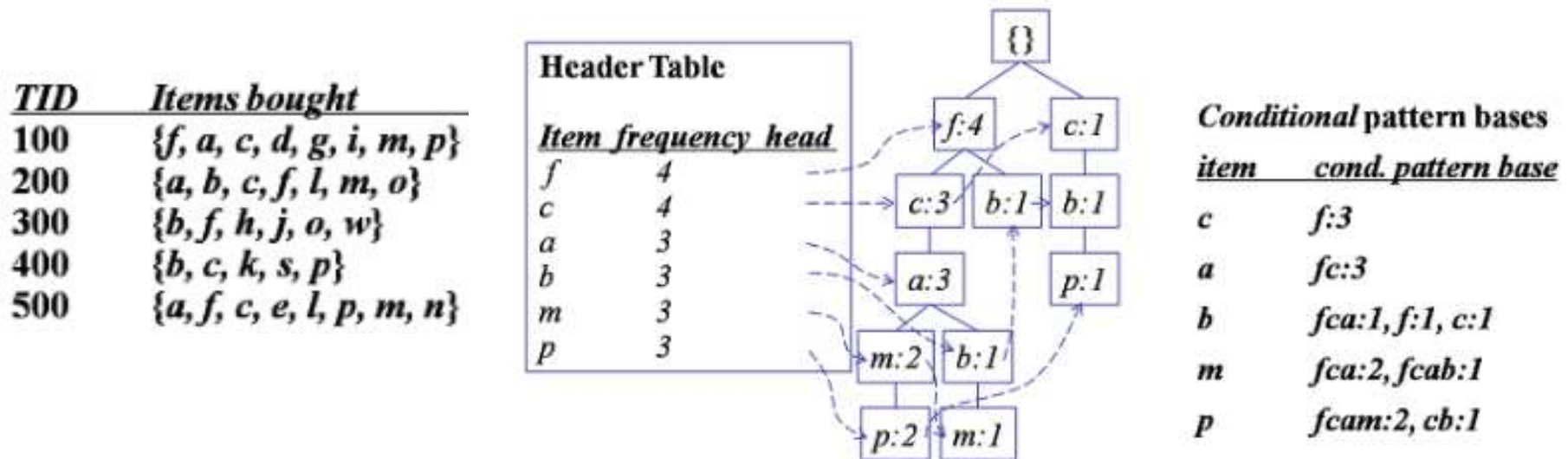


3. 비지도 학습 (Unsupervised Learning)

5. Recommendation

❖ 연관 규칙 - FP-Growth Algorithm

- Apriori Algorithm DB scan 횟수 개선
- 1th scan : min Support 조건에 만족하는 데이터로 테이블 생성 및 정렬
- 2th scan : DB를 scan하면서 테이블에 있는 데이터를 기준으로 FP 트리 생성.



3. 비지도 학습 (Unsupervised Learning)

5. Recommendation

❖ 순차 패턴 (Sequential pattern analysis)

노트북을 산 후, USB 구매?

- 시간차에 의한 구매 패턴 예측 가능.

비디오점의 대여 기록 data

고객번호

구매기록

1	{겨울연가} => {아폴로13, 캐스트웨이}
2	{겨울연가} => {아폴로13, 공동경비구역}
3	{러브레터} => {시월이야기, 동감}{시월애}
4	{겨울연가} => {캐스트웨이}

지지도 50% 이상의 순차 패턴은 ?

{겨울연가} => {아폴로13}

and

{겨울연가} => {캐스트웨이}

3. 비지도 학습 (Unsupervised Learning)

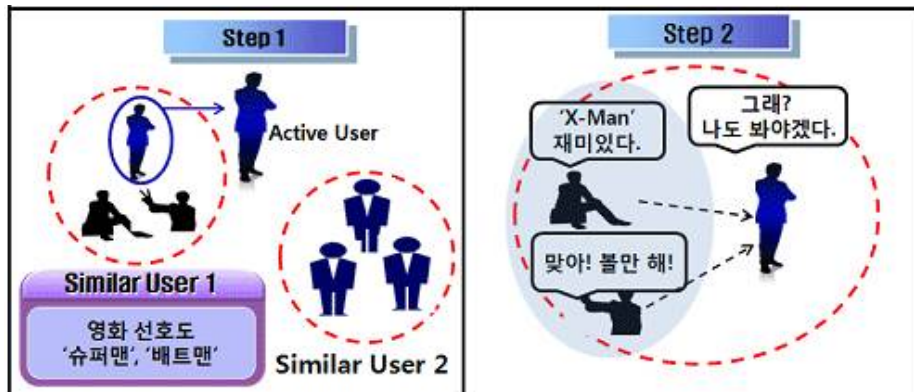
5. Recommendation

❖ 협력 필터(CF, Collaborative Filtering)

로맨스 장르의 영화가 좋은데 이런 비슷한 영화 없을까?

- 사용자 기반 : 취향이 비슷한 사람을 찾아서 그 사람이 선택한 것 중 미구매 상품 추천.
- 아이템 기반 : 고객이 과거에 구매했던 상품들의 속성과 유사한 다른 상품 중 미구매 상품 추천.

User-based



Item-based



3. 비지도 학습 (Unsupervised Learning)

5. Recommendation

❖ 추천 종류 별 알고리즘 적용



Module II.

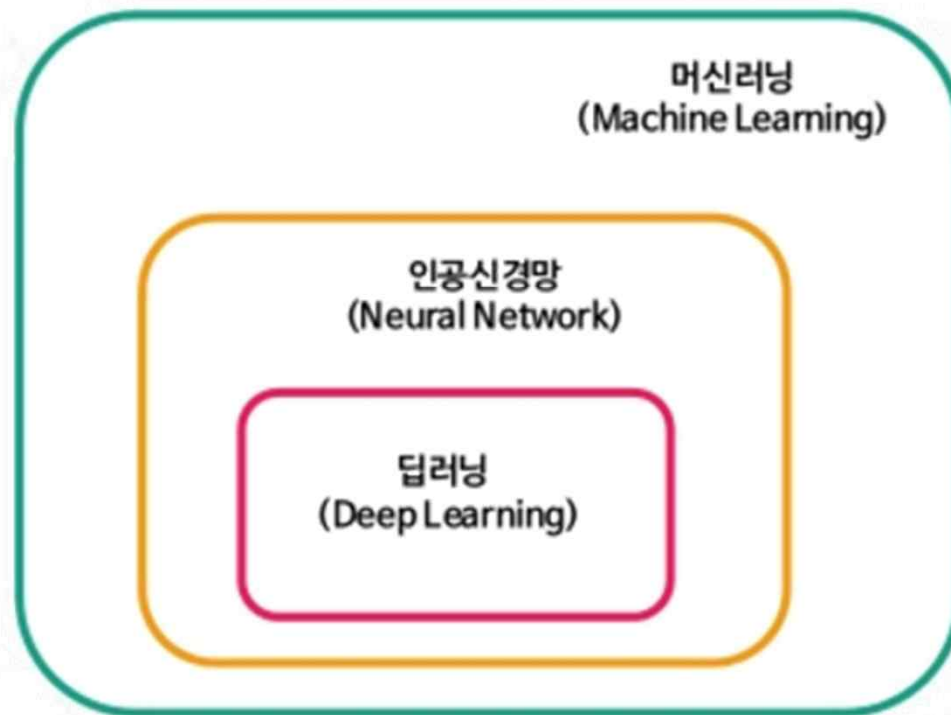
Neural Network

- 1 Introduction
- 2 Neural Network
- 3 Deep Learning

1. Introduction

1. Neural Network – Deep Learning

딥러닝? 머신러닝?

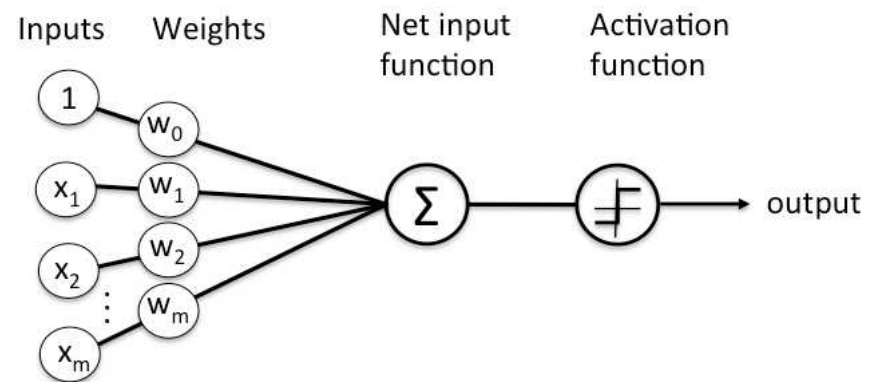
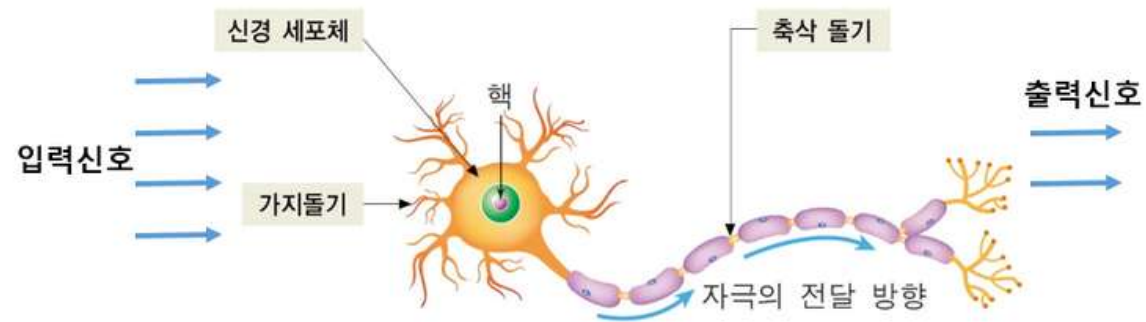


2. Neural Network

1. 퍼셉트론

❖ Perceptron

- 단순 퍼셉트론



2. Neural Network

1. 퍼셉트론

❖ Perceptron – python

- 꽃의 특징 데이터로 품종 분류



임계값 = 0
 $\eta = 0.1$

$$\sum w_j x_j > 0 \rightarrow 1 \quad \text{Iris-Versicolor}$$

$$\sum w_j x_j \leq 0 \rightarrow -1 \quad \text{Iris-Setosa}$$

$$w_j = w_j + \eta(y - y^{\wedge})x_j$$

	$x_0 = 1$	x_1 (꽃받침길이)	x_2 (꽃받침너비)	x_3 (꽃잎길이)	x_4 (꽃잎너비)	품종
1		5.1	3.5	1.4	0.2	Iris-Setosa
2		4.9	3.0	1.4	0.2	Iris-Setosa
...	
51		6.4	3.5	4.5	1.2	Iris-Versicolor
...	
100		5.7	2.8	4.1	1.3	Iris-Versicolor

2. Neural Network

1. 퍼셉트론

❖ Perceptron - python

```

1
2 # coding: utf-8
3
4 # In[1]:
5
6 get_ipython().magic('matplotlib inline')
7
8 import numpy as np
9 import pandas as pd
10 import matplotlib
11 import matplotlib.pyplot as plt
12 from matplotlib import style
13 from perceptron import Perceptron
14
15 style.use('seaborn-talk')
16
17 if __name__ == '__main__':
18     style.use('seaborn-talk')
19
20     df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/iris/iris.data', header=None)
21
22     y = df.iloc[0:100, 4].values
23     y = np.where(y=='Iris-setosa', -1, 1)
24     X = df.iloc[0:100, [0, 2]].values
25
26     plt.scatter(X[:50, 0], X[:50, 1], c='r', marker='o', label='setosa', )
27     plt.scatter(X[50:100, 0], X[50:100, 1], c='b', marker='x', label='versicolor')
28
29     plt.xlabel('petal length(cm)')
30     plt.ylabel('sepal length(cm)')
31     plt.title('Iris_Perceptron')
32     plt.legend(loc='best')
33     plt.show()
34
35     ppn1 = Perceptron(eta=0.1)
36     ppn1.fit(X,y)
37     print(ppn1.errors_)

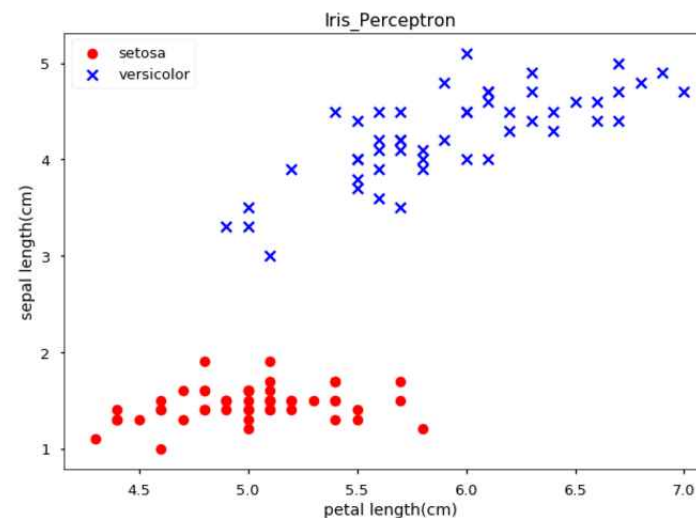
```

[1~100 data]

5.1,3.8,1.9,0.4,Iris-setosa
4.8,3.0,1.4,0.3,Iris-setosa
5.1,3.8,1.6,0.2,Iris-setosa
4.6,3.2,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
6.5,2.8,4.6,1.5,Iris-versicolor
5.7,2.8,4.5,1.3,Iris-versicolor

50 setosa

50 versicolor



```

[ 0.2  1.4  0.94]
[ 0.   0.8  1.32]
[-0.2  0.2  1.7 ]
[-0.2  0.32  2.12]
[-0.4 -0.7  1.84]
[-0.4 -0.7  1.84]
[-0.4 -0.7  1.84]
[-0.4 -0.7  1.84]
[-0.4 -0.7  1.84]
[-0.4 -0.7  1.84]
[1, 3, 3, 2, 1, 0, 0, 0, 0, 0]

```

*** 머신러닝 결과 ***

$-0.4 + (-0.7) \times (\text{꽃받침길이}) + 1.84 \times (\text{꽃잎길이}) > 0 \rightarrow 1$ Iris-Versicolor

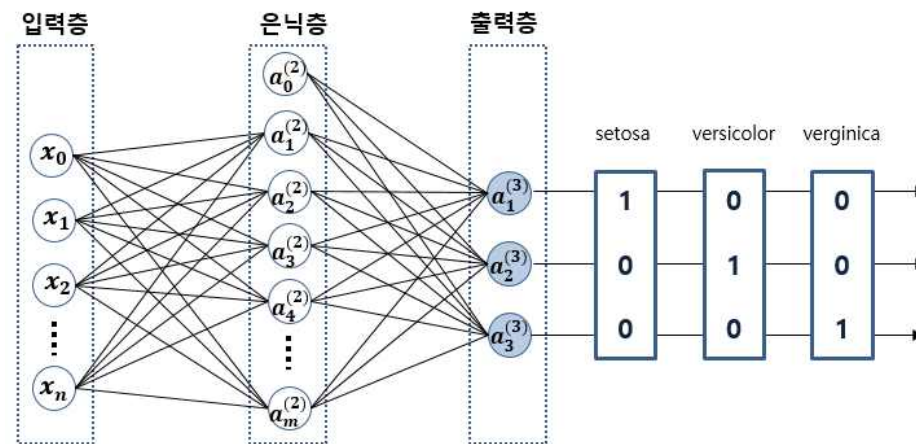
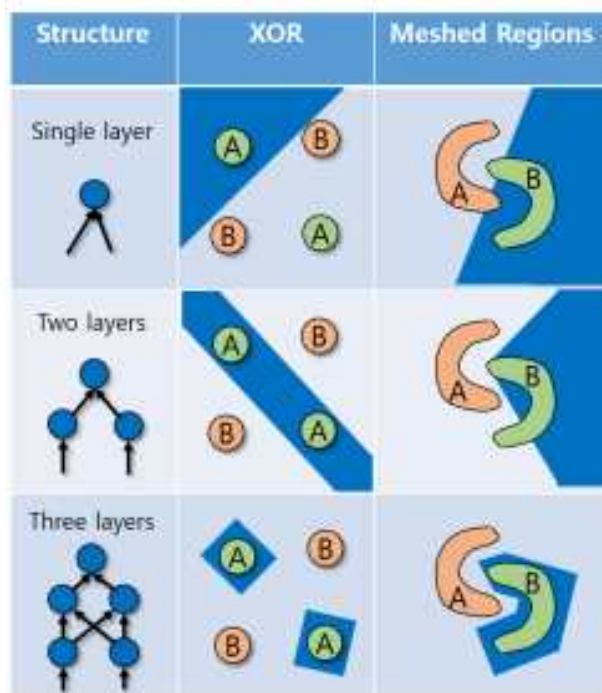
$\leq 0 \rightarrow -1$ Iris-Setosa

2. Neural Network

1. 퍼셉트론

❖ 다층 퍼셉트론 (MLP/FFNN, Multi-layer perceptron)

- 한 방향으로만 활성화 (순방향 신경망, FFNN)

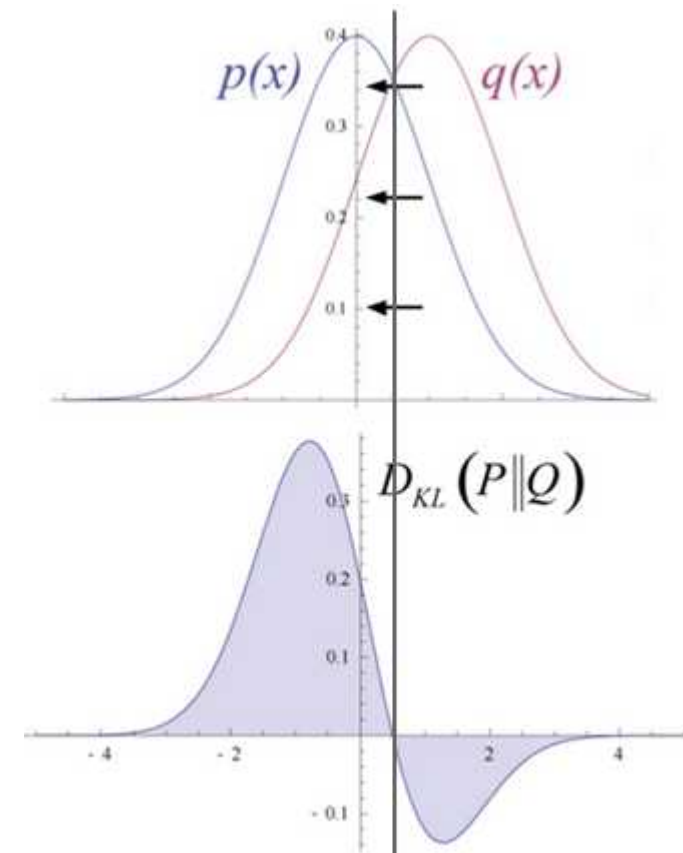
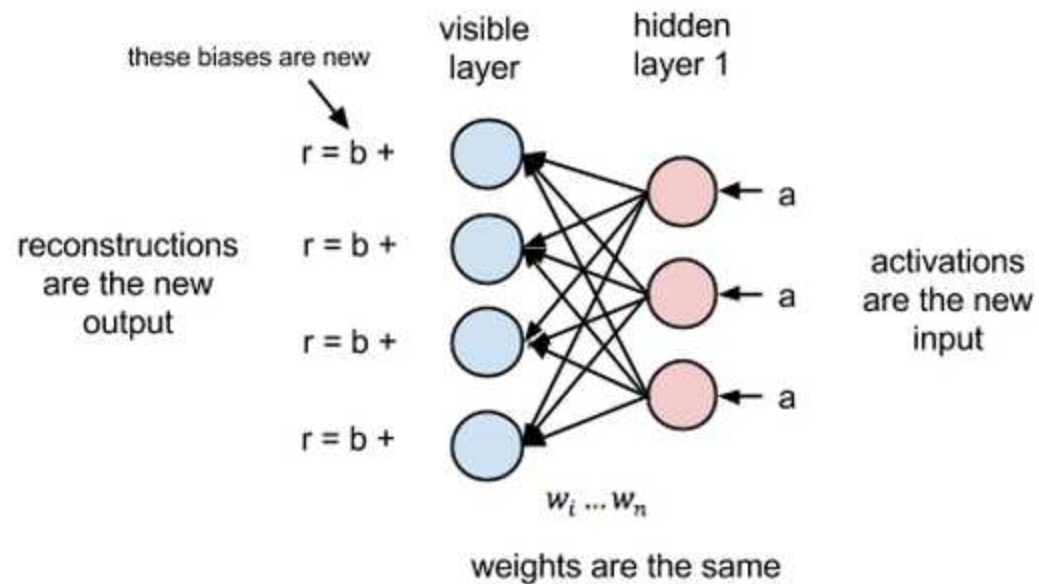


2. Neural Network

2. 신경망

❖ 제한된 볼츠만 머신 (RBM, Restricted Boltzmann machine)

차원감소, 분류, 특징 값 학습, 주제 모델링

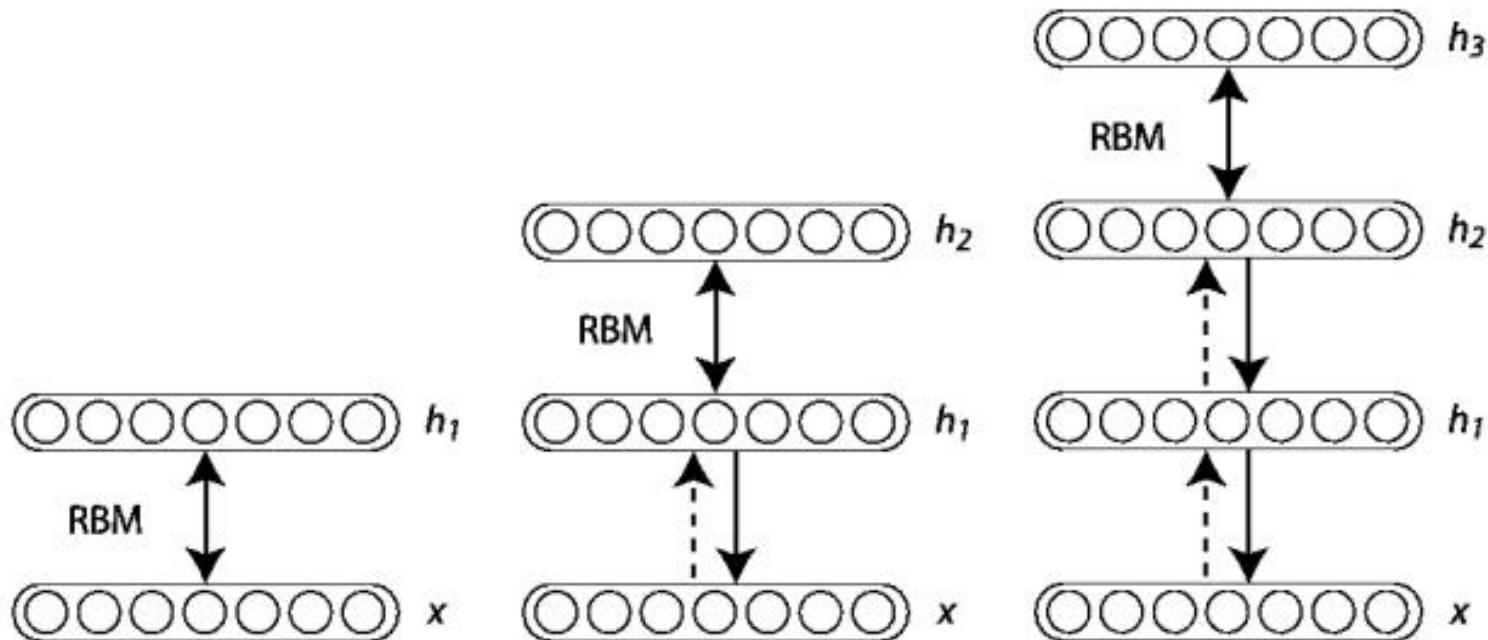


2. Neural Network

2. 신경망

❖ 심층 신뢰 신경망 (DBN, Deep Belief Network)

- RBM을 여러 겹 쌓아서 DBN을 만듦.

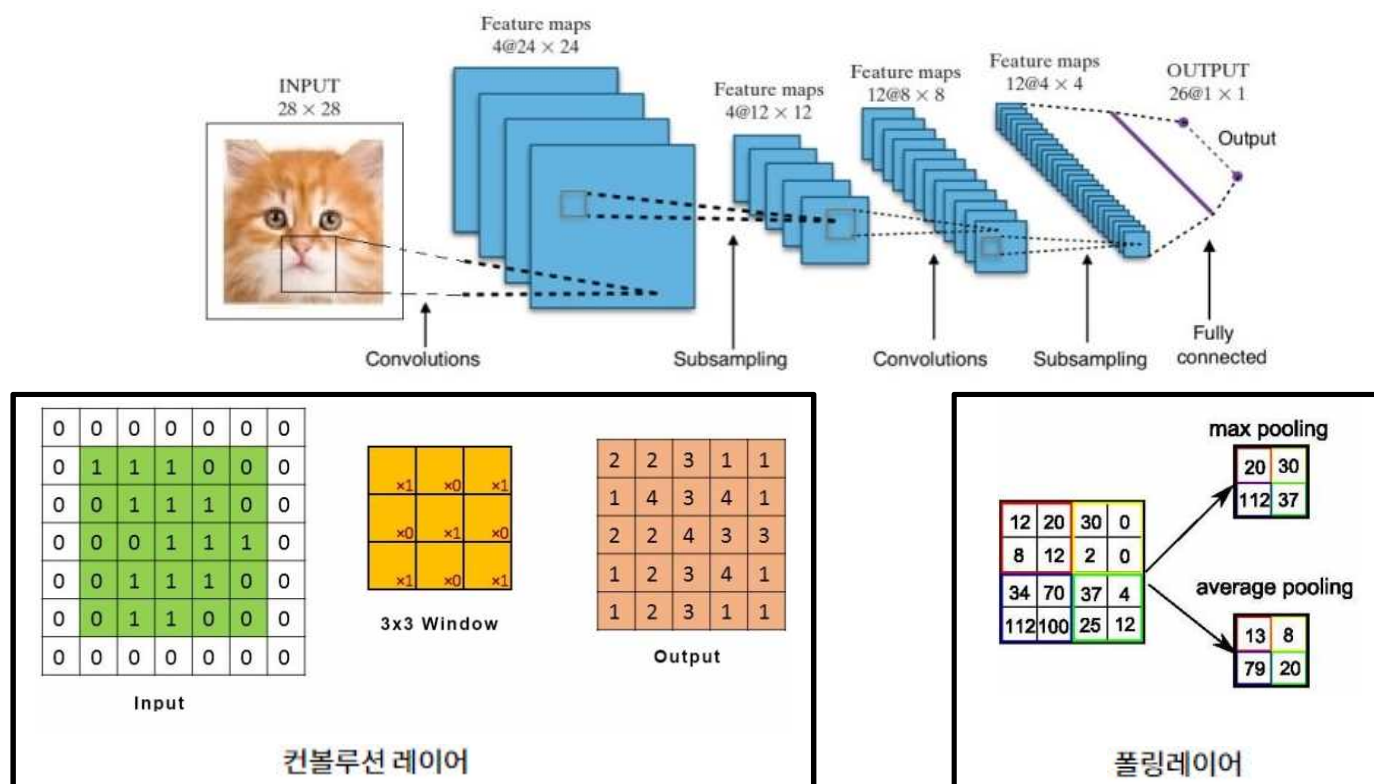


3. Deep Learning

1. 이미지 처리

❖ 합성곱 신경망(CNN, Convolutional Neural Network)

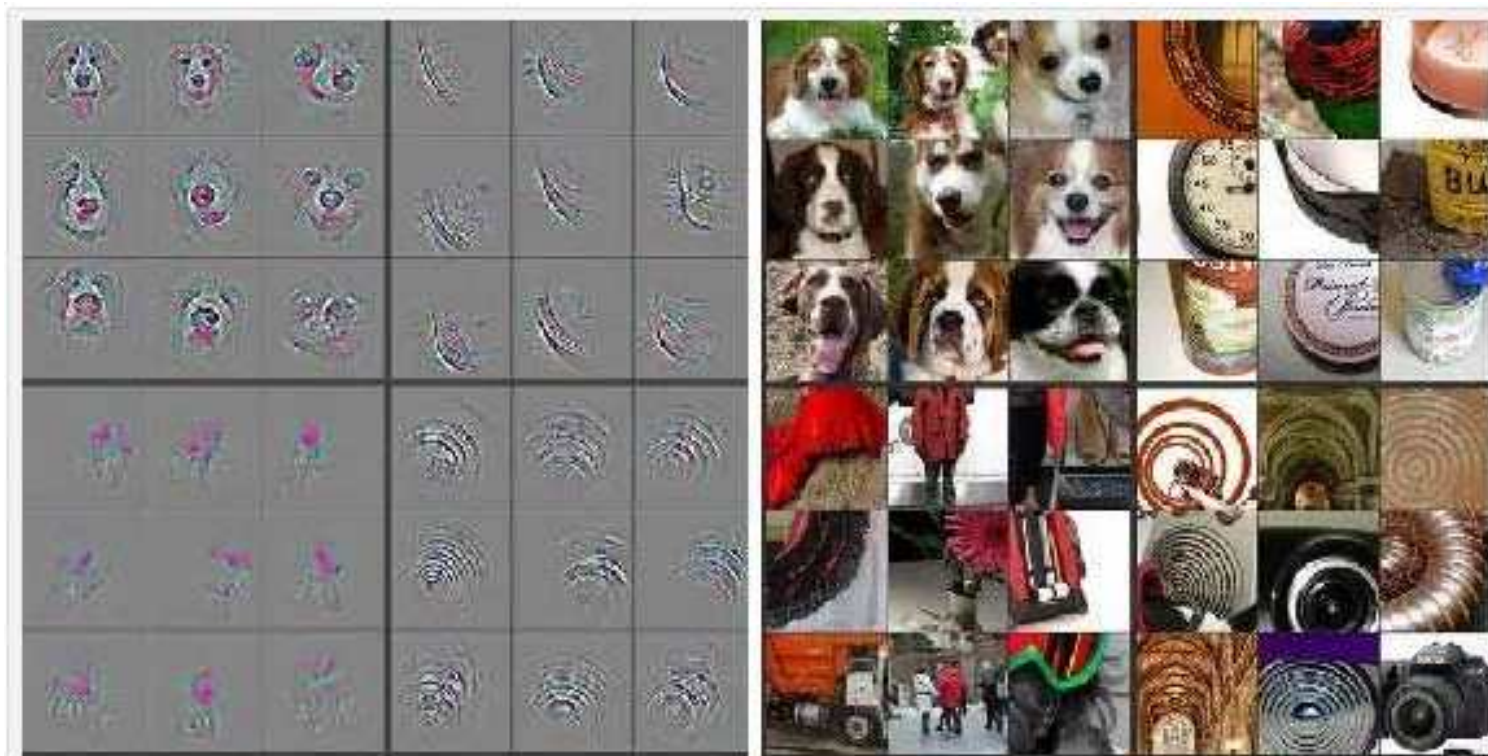
이 이미지는 고양이일까? 강아지 일까?



3. Deep Learning

1. 이미지 처리

❖ 합성곱 신경망(CNN, Convolutional Neural Network)



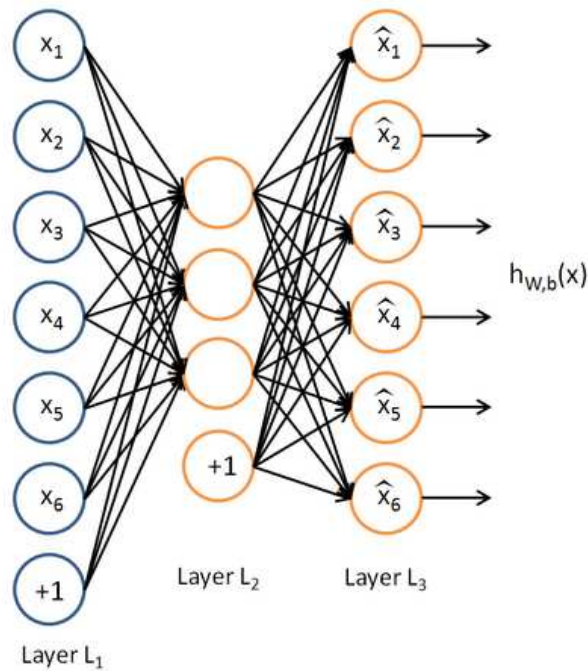
원본 이미지(우측)와 컨볼루션 네트워크에 의해 추출된 특징 지도(좌측) (이미지 출처: M. Zeiler)

3. Deep Learning

1. 이미지 처리

❖ 자기부호화기(AutoEncoder)

- 입력 = 출력
- Hidden layer에서 입력 벡터를 차원 축소 하여 특징 추출
- 추출한 특징을 SVM, NW에 사용하여 분류에 쓰임.

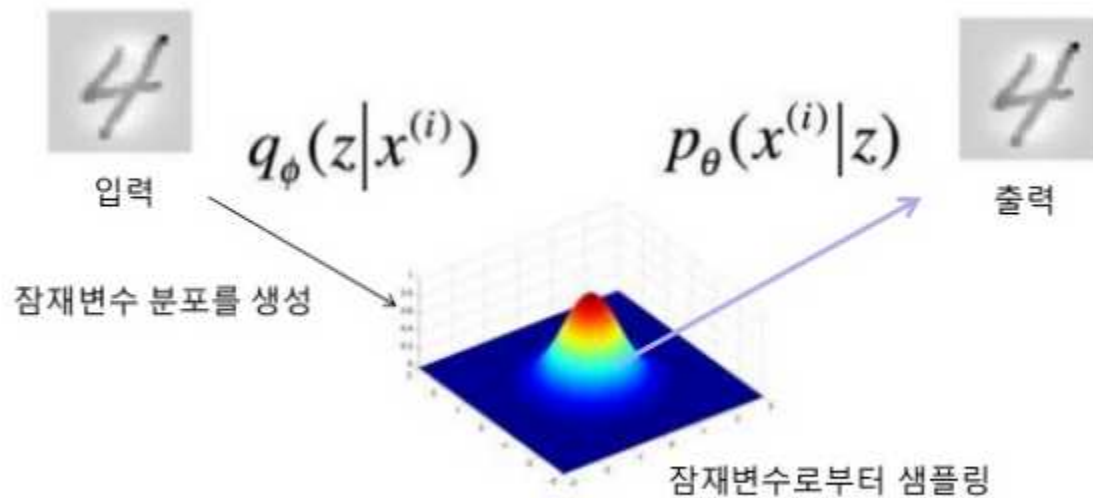


3. Deep Learning

1. 이미지 처리

❖ Variational AutoEncoder(VAE)

사진에 나타나지 않은 표정 생성?



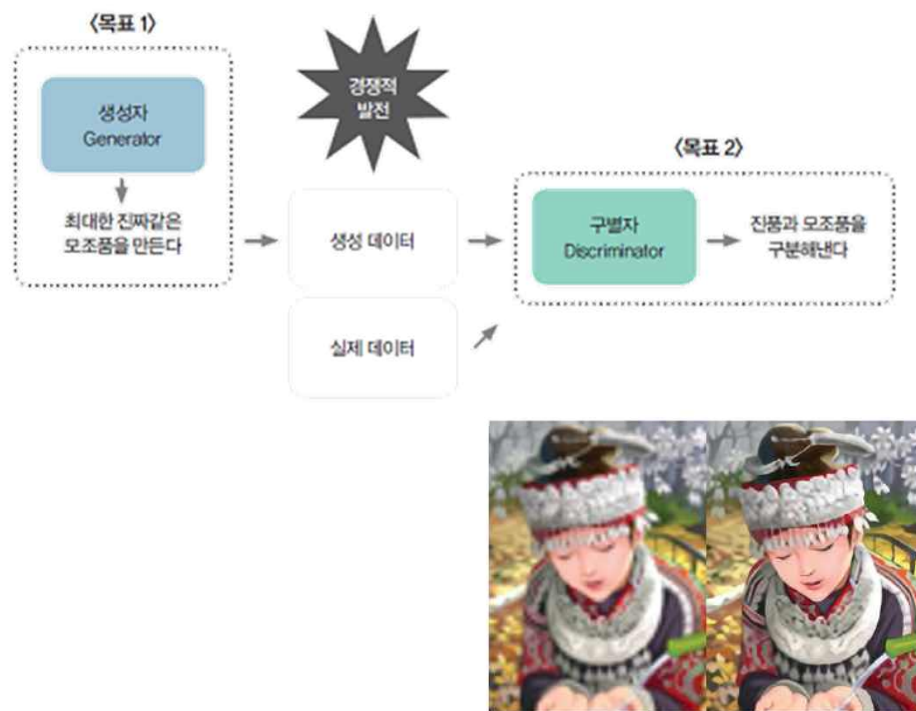
표정의 생성

3. Deep Learning

1. 이미지 처리

❖ Generative Adversarial Network (GAN)

- 기존 학습 방법 → 감별사와 위범 같은 경쟁 방법
- 대충 스케치 해주면 진짜 같아지는 학습



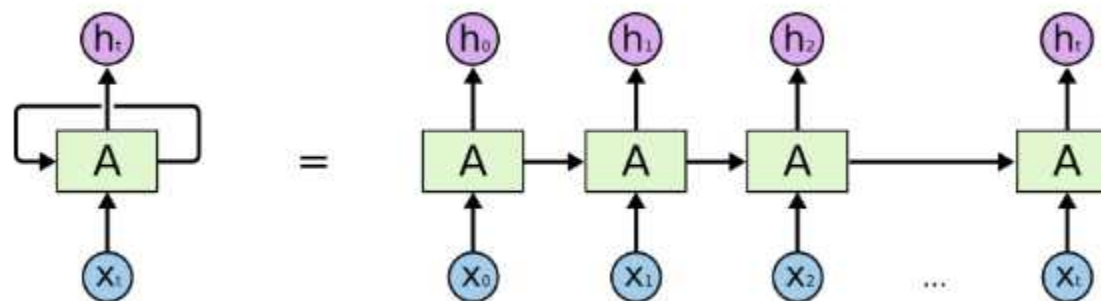
3. Deep Learning

2. 텍스트 처리

❖ 순환 신경망(RNN, Recurrent Neural Network)

연관 검색어? 구글 번역?

- 시계열 데이터 처리, 데이터의 패턴 학습



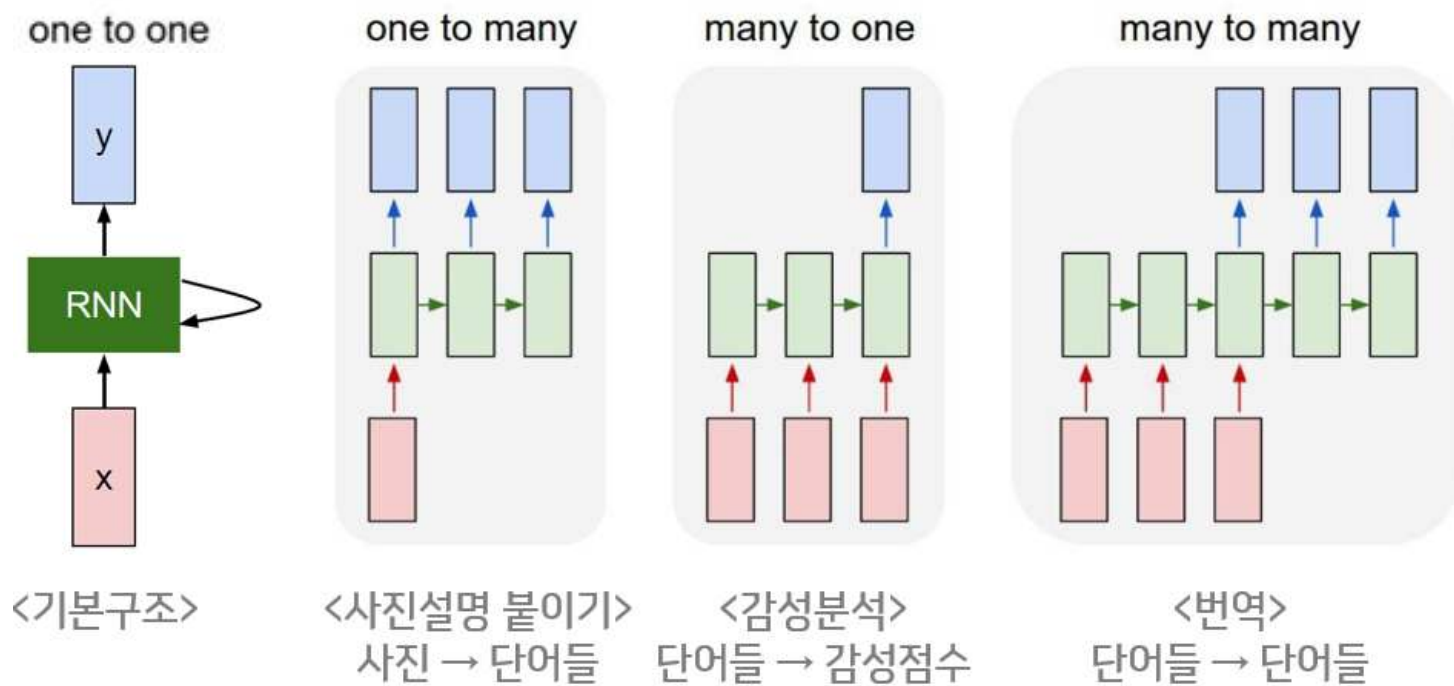
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state some function with parameters W old state input vector at some time step

3. Deep Learning

2. 텍스트 처리

❖ 순환 신경망(RNN, Recurrent Neural Network)



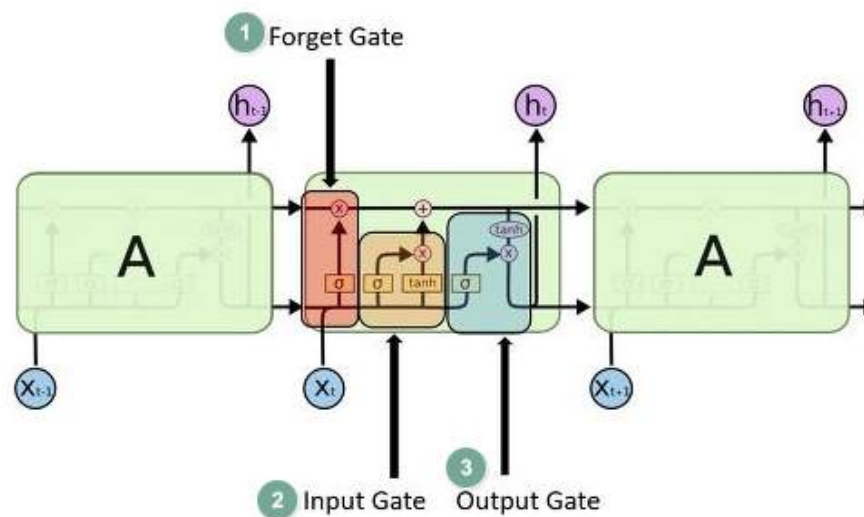
One-to-many : 소녀는 사과를 고르고 있다.
Many-to-one : 안정, 불안, 공포
Many-to-many : 구글 번역, 동영상에 대한 여러 개의 설명

3. Deep Learning

2. 텍스트 처리

❖ 장기 단기 메모리 (LSTM, Long Short-Term Memory)

- RNN의 기억 능력을 향상 시킴.



1. Forget Gate : 값을 업데이트 할지 결정
2. Input Gate - 값을 업데이트
3. Output Gate - 출력할 값 결정

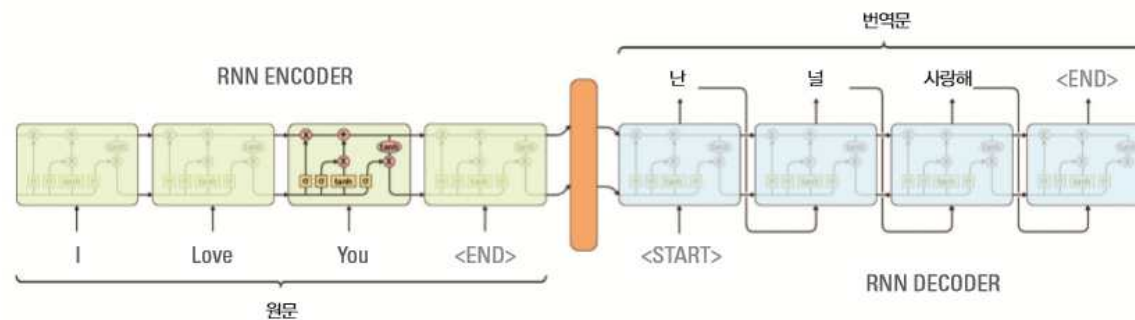
3. Deep Learning

2. 텍스트 처리

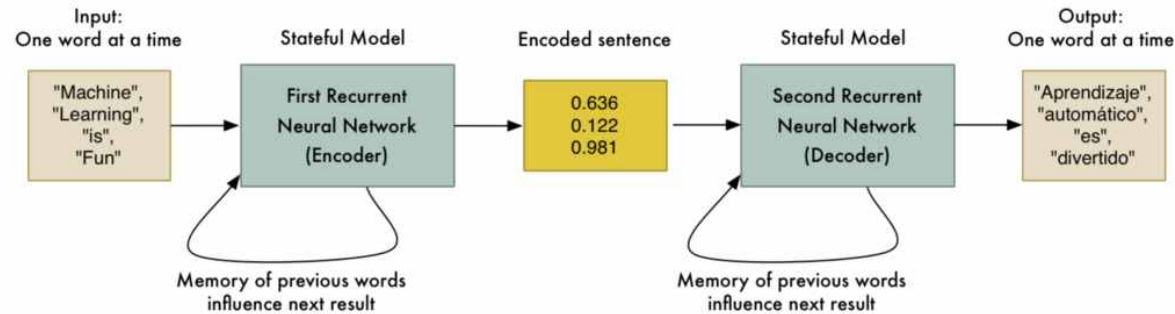
❖ 순환 신경망(RNN, Recurrent Neural Network)

- 번역 사용

RNN은 문장 하나에 대한 고유한 벡터 값을 만들 수 있음.



RNN Encoder-Decoder를 활용한 기계번역 모델의 구조. Encoder와 Decoder는 각각 같은 크기의 LSTM-RNN으로 이루어져 있으며, 각자의 Weight를 따로 가지고 있다. Encoder에서 원문을 받아 원문의 정보를 LSTM-RNN으로 담은 후, 이 정보를 Decoder로 옮겨 번역문으로 해석한다.



Module III.

활용



활용

1. 유사어 활용

❖ 단어/문서의 벡터화

- 유사어 찾기

Word2vec/Doc2vec, SVM(LSA) – 위키피디아 등

I love NLP and I like dogs

	I	Love	NLP	And	Like	Dogs
I	0	1	0	1	1	0
Love	1	0	1	0	0	0
NLP	0	1	0	1	0	0
And	1	0	1	0	0	0
Like	1	0	0	0	0	1
Dogs	0	0	0	0	1	0

```
I = [0 1 0 1 1 0]
Love = [1 0 1 0 0 0]
NLP = [0 1 0 1 0 0]
And = [1 0 1 0 0 0]
Like = [1 0 0 0 0 1]
Dogs = [0 0 0 0 1 0]
```

```
pprint(doc_vectorizer.most_similar('공포/Noun'))
# => [('서스펜스/Noun', 0.5669919848442078),
#      ('미스터리/Noun', 0.5669919848442078),
#      ('스릴러/Noun', 0.5669919848442078),
#      ('장르/Noun', 0.4822016954421997),
#      ('판타지/Noun', 0.4395076632499695),
#      ('무게/Noun', 0.4077949523925781),
#      ('호러/Noun', 0.4026390314102173),
#      ('환타지/Noun', 0.4003834724426295),
#      ('멜로/Noun', 0.39946430921554565),
#      ('공포영화/Noun', 0.3899948000907898),
#      ('^/Punctuation', 0.3852730989456177),
#      ('^^/Punctuation', 0.3797937035560608)]
```



Response	Percentage
Yes	78%
No	22%

2. 제품의 특징 추출

❖ 특정 주제 단어의 벡터화

- 위키피디아를 학습한 word2vec

```
In [1]: from gensim.models import word2vec
In [2]: model = word2vec.Word2Vec.load('wiki.model')
In [3]: model.most_similar(positive=["Python","파이썬"])
Out[3]:
[('Lisp', 0.8989054560661316),
 ('OCaml', 0.8782240152359009),
 ('MATLAB', 0.8769774436950884),
 ('Java', 0.8666226267814636),
 ('Tcl', 0.8658475875854492),
 ('Markup', 0.8623142242431641),
 ('VHDL', 0.860676646232605),
 ('Emacs', 0.858577311038971),
 ('자바스크립트', 0.8584067821502686),
 ('OpenGL', 0.8581513166427612)]

In [9]: model.most_similar(positive=["서울","맛집"])
Out[9]:
[('서울특별시', 0.6868484020233154),
 ('여의도', 0.6540044546127319),
 ('서울시', 0.637624979019165),
 ('인사동', 0.6320654153823853),
 ('청담동', 0.6317439675331116),
 ('강남구', 0.6309967041015625),
 ('호의동', 0.6214122772216797),
 ('신촌', 0.6208873987197876),
 ('충천', 0.6188673377037048),
 ('해운대', 0.6161919832229614)]
```

```
print(doc_vectorizer.most_similar('공포/Noun'))
# => [('서스펜스/Noun', 0.5669919848442078),
#      ('미스터리/Noun', 0.5669919848442078),
#      ('스릴러/Noun', 0.5669919848442078),
#      ('장르/Noun', 0.4822016954421997),
#      ('판타지/Noun', 0.4395076632499695),
#      ('무게/Noun', 0.4077949523925781),
#      ('호러/Noun', 0.4026390314102173),
#      ('환타지/Noun', 0.4003834724426295),
#      ('멜로/Noun', 0.39946430921554565),
#      ('공포영화/Noun', 0.3899948000907898),
#      ('^/Punctuation', 0.3852730989456177),
#      ('~/Punctuation', 0.3797937035560608)]

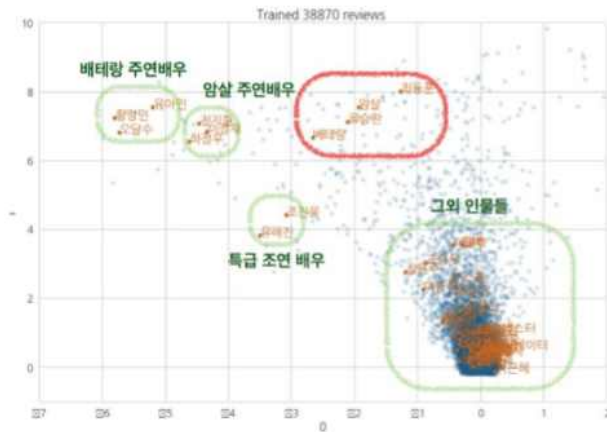
print(doc_vectorizer.most_similar('ㅋㅋ/KoreanParticle'))
# => [('ㅎㅎ/KoreanParticle', 0.5768033862113953),
#      ('~/KoreanParticle', 0.4822016954421997),
#      ('!!!/Punctuation', 0.4395076632499695),
#      ('!!!/Punctuation', 0.4077949523925781),
#      ('!/Punctuation', 0.4026390314102173),
#      ('~/Punctuation', 0.4003834724426295),
#      ('~/Punctuation', 0.39946430921554565),
#      ('!/Punctuation', 0.3899948000907898),
#      ('^/Punctuation', 0.3852730989456177),
#      ('~/Punctuation', 0.3797937035560608)]
```

- 학습 데이터가 특정 주제의 데이터 일 경우, word2vec을 통해 제품의 특징을 잡아낼 수 있지 않을까?

1. 활용

3. 추천 시스템

❖ 유사도를 통한 추천 시스템



- 유사도로 벡터화 되는 word2vec를 이용해 추천 알고리즘(CF, FP-Growth) 보완?



- 정확한 제품명을 알지 못해도 클릭한 상품의 이미지와 유사한 이미지(CNN)를 이용한 추천 시스템 활용?

1. 활용

3. 추천 시스템

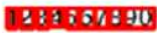
❖ 오픈소스

Software	Description	Language	URL
Apache Mahout	Machine learning library includes collaborative filtering	Java	http://mahout.apache.org
Cofi	Collaborative filtering library	Java	http://www.nongun.org/cofi
Crab	Components to create recommender systems	Python	https://github.com/muricoca/crab
easyrec	Recommender for Web pages	Java	http://easyrec.org
LensKit	Collaborative filtering algorithms from GroupLens Research	Java	http://lenskit.grouplens.org
MyMediaLite	Recommender system algorithms	C#/Mono	http://mloss.org/software/view/282
SVDFeature	Toolkit for featurebased matrix factorization	C++	http://mloss.org/software/view/333
Vogo PHP LIB	Collaborative filtering engine for personalizing web sites	PHP	http://sourceforge.net/projects/vogoo

1. 활용

4. 이미지 내 문자 인식

❖ OpenCV - python

숫자	1 2 3 4 5 6 7 8 9 0		
영어	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz HELLO		
한글	가 나 다 라 마 바 사 아 자 차 카 타 파 하 가 가 거 겨 고 구 규 그 기 강 낭 당 량 망 방 상 양 장 창 캉 탕 광 향 엔 테 이 터		

- 이미지 전 처리로 인식률 개선
- SVM, NN를 이용해 문자 추출
- 오프라인 문서를 온라인 텍스트로 변환?
- 이미지 내 문자 번역 or 분석 ?

감 사 합 니 다