

The background of the slide is a close-up, slightly blurred image of a financial chart or ledger. A silver pen is positioned in the upper right corner, pointing towards a data point on the chart. The chart features a grid with dotted lines and some handwritten numbers, including '2.5' and '2.47'.

4가지 유즈 케이스를 활용한 시계열 분석

3RD SESSION
JAMES LEE

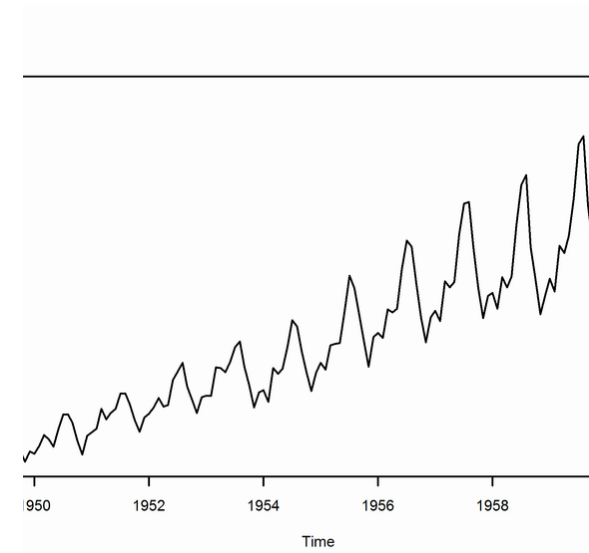
Recap:

시계열 데이터의 결측치

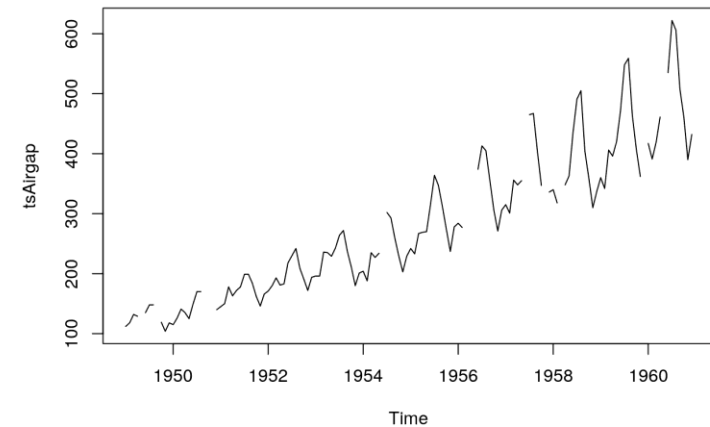
일반적인 데이터에 비해서도 결측치를 어떻게 처리할지는 매우 중요한 이슈다

하지만 시계열에서 결측치는 특히 중요하다

- 대부분의 시계열은 비정상이다
- 따라서, 시간에 따라 평균과 분산이 변한다
- 특정 시점 값의 소실은 그 시점의 평균과 분산의 왜곡을 가져오고 이는 분석 결과에 치명적인 영향을 미친다



AirPassenger data with missing values



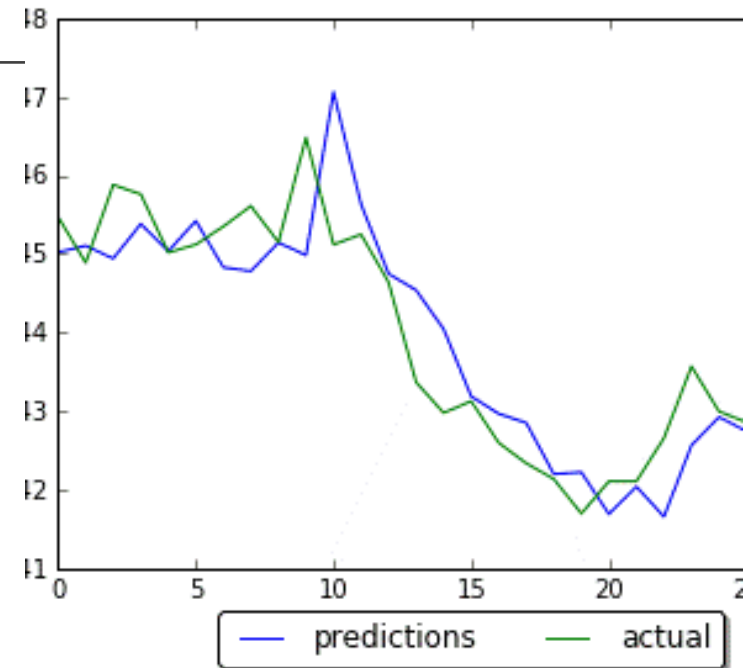
Recap: 시계열 데이터의 전처리 - 노이즈

노이즈는 특히 시계열 데이터 분석에 있어 치명적인 영향을 미치며

- 시간의 흐름에 따라 변화하는 통계적 특성을 강하게 왜곡

결측치와 마찬가지로 시간 순대로 기록되는 정보이기에 노이즈가 같이 기록될 가능성도 높고

일반적인 데이터와 노이즈 제거 방법도 다름



Recap: 시계열 데이터 교차검증

교차검증 (Cross-validation) 이란?

- 학습 (Train) 데이터와 검증 (Test) 데이터를 바꿔가면서 모델의 성능을 측정하는 방법
- 모델의 과최적 (Over-fitting)을 막기 위해 반드시 수행해야 하는 검증 중 하나

My model on training data



My model on test dataset



3주차 – 주가 시계열 딥러닝 적용

이론

- 머신러닝의 기초
- Supervised Learning, Un-Supervised Learning
- Discriminative model vs Generative model
- RNN, CNN의 개념

실습

- 간단한 딥러닝 모델들 구축
- 주가 차트 시계열 데이터에 RNN 및 CNN 적용하기

Layer (type)	Output Shape	Param #
input_21 (InputLayer)	[(None, 1008, 1)]	0
lstm_20 (LSTM)	(None, 10)	480
dropout_20 (Dropout)	(None, 10)	0
dense_20 (Dense)	(None, 1)	11
Total params: 491		
Trainable params: 491		
Non-trainable params: 0		

```
epoch #0
Train on 12800 samples
12800/12800 [=====] - 26s 2ms/sample - loss: 0.0190 - mse: 0.0190
Train on 12800 samples
12800/12800 [=====] - 25s 2ms/sample - loss: 0.0118 - mse: 0.0118
Train on 12800 samples
12800/12800 [=====] - 24s 2ms/sample - loss: 0.0142 - mse: 0.0142
Train on 12800 samples
12800/12800 [=====] - 24s 2ms/sample - loss: 0.0215 - mse: 0.0215
Train on 12800 samples
12800/12800 [=====] - 25s 2ms/sample - loss: 0.0087 - mse: 0.0087
Train on 12800 samples
12800/12800 [=====] - 24s 2ms/sample - loss: 0.0090 - mse: 0.0090
Train on 12800 samples
12800/12800 [=====] - 25s 2ms/sample - loss: 0.0074 - mse: 0.0074
Train on 12800 samples
12800/12800 [=====] - 25s 2ms/sample - loss: 0.0120 - mse: 0.0120
Train on 12800 samples
12800/12800 [=====] - 24s 2ms/sample - loss: 0.0124 - mse: 0.0124
Train on 12800 samples
12800/12800 [=====] - 22s 2ms/sample - loss: 0.0102 - mse: 0.0102
Train on 12800 samples
2688/12800 [====>.....] - ETA: 17s - loss: 0.0199 - mse: 0.0199
```

딥러닝의 개념

머신러닝의 기초

What is Machine Learning?

Explicit programming
(명시적 프로그래밍)

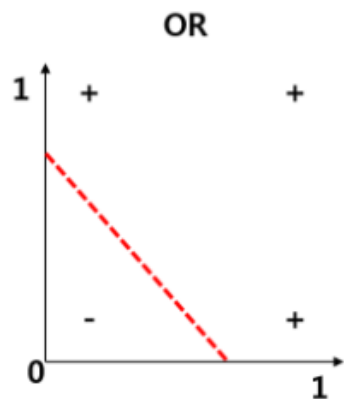
- 프로그래머가
프로그램의 모든 Rule을
지정해주는 프로그래밍
패러다임

```
1 class A{
2     boolean a;
3     char c;
4     byte b;
5     short s;
6     int i;
7     long l;
8     float f;
9     double d;
10    String y; //Class type variable
11    A z;      //Class type variable
12    public static void main(String[] args){
13        A a=new A();
14        char c1='\u0000'; //"\0000 is Equivalent to 0"
15        System.out.println(a.a);
16        System.out.println(a.c==c1);
17        System.out.println(a.b);
18        System.out.println(a.s);
19        System.out.println(a.i);
20        System.out.println(a.l);
21        System.out.println(a.f);
22        System.out.println(a.d);
23        System.out.println(a.y);
24        System.out.println(a.z);  }
```

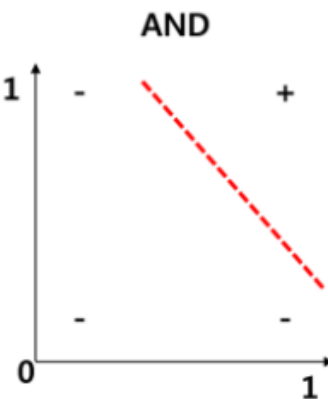
머신러닝의 기초

What is Machine Learning?

Explicit programming의 한계 – Classification 문제



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



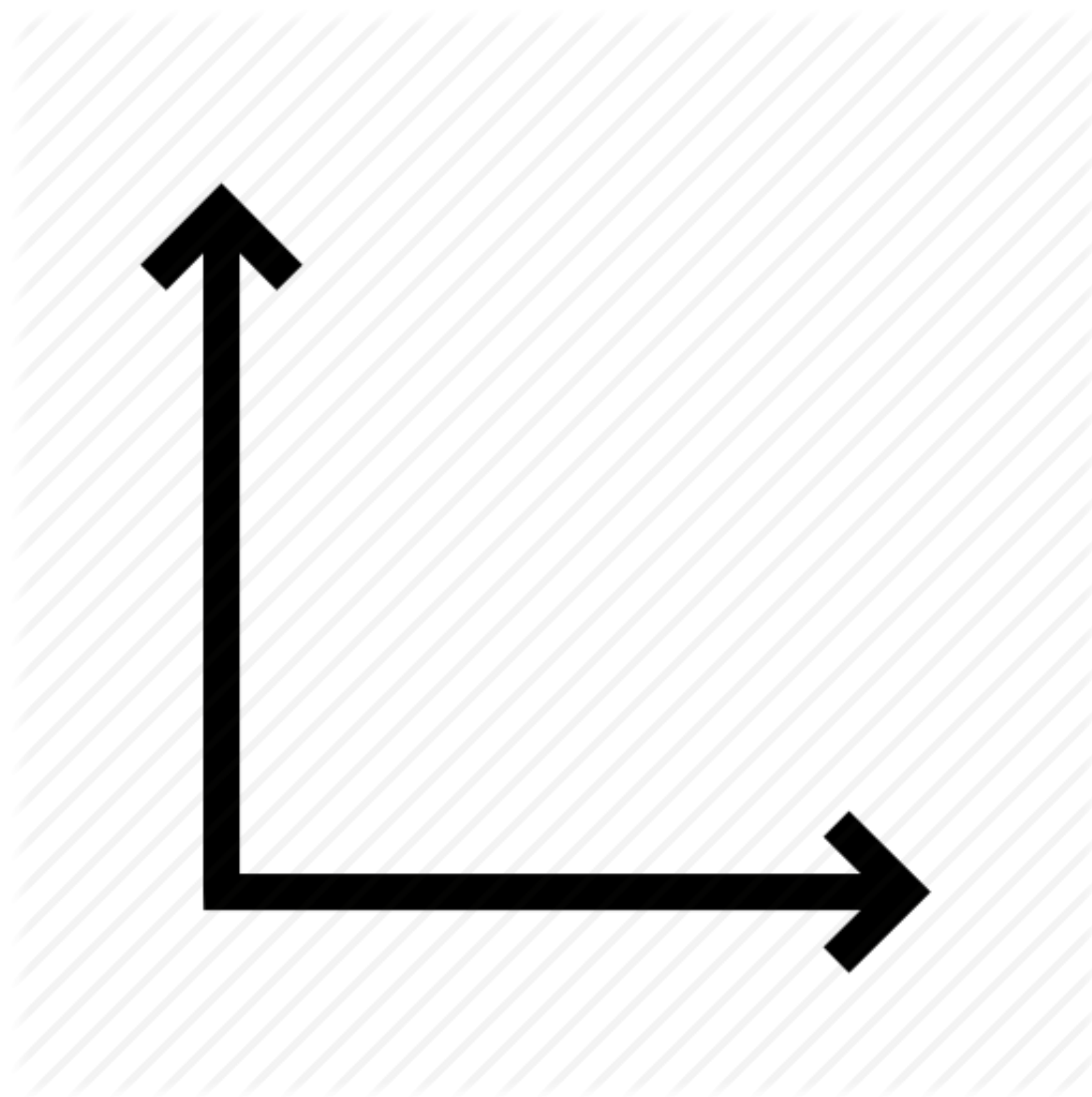
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

머신러닝의 기초

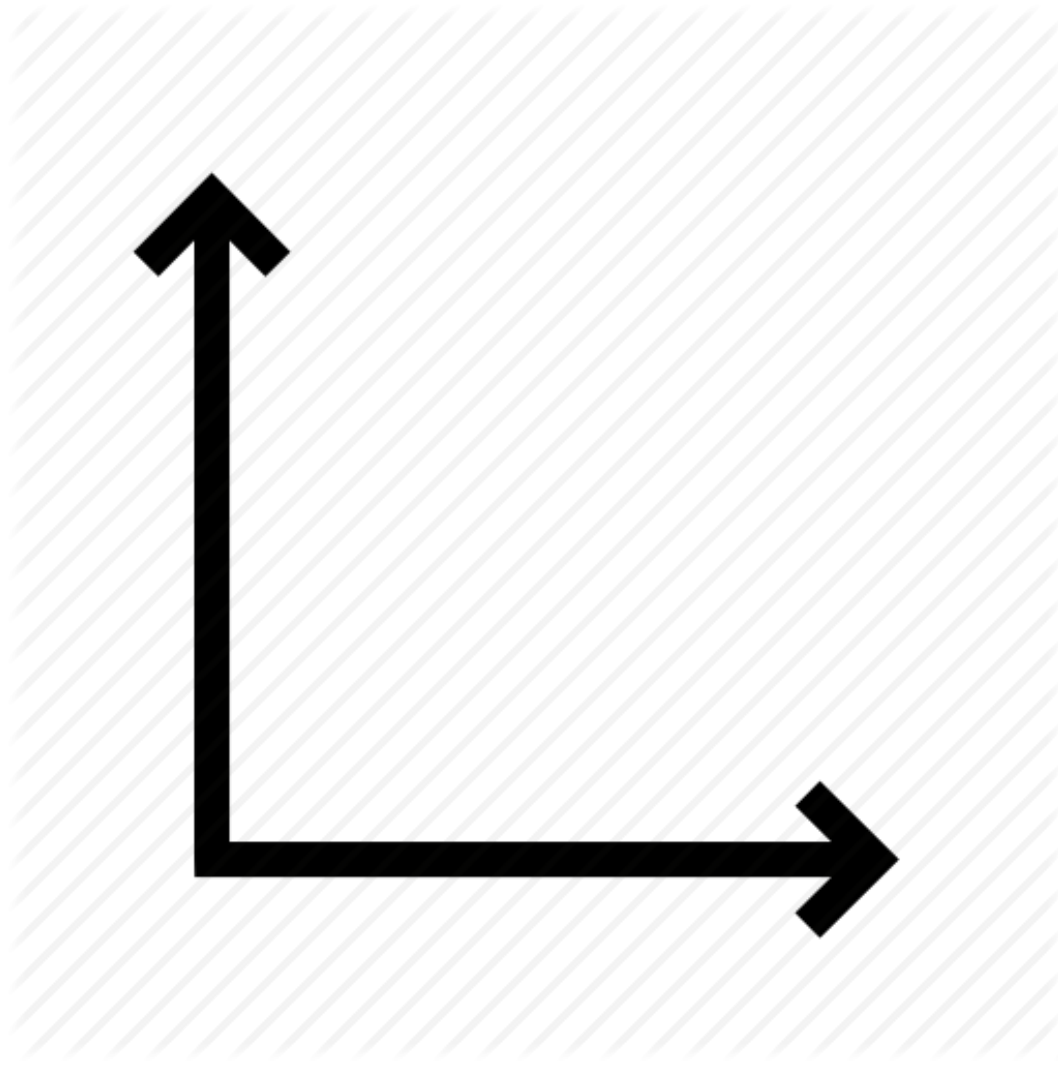
What is Machine Learning?

Explicit programming의
한계 – Classification 문제

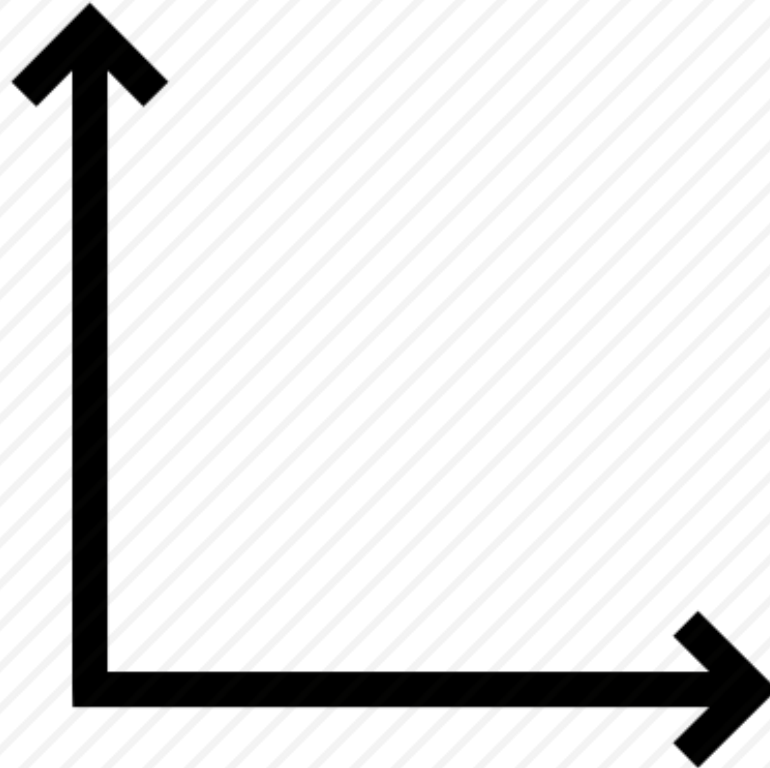
- XOR문제를 어떻게 해결
할 것인가?



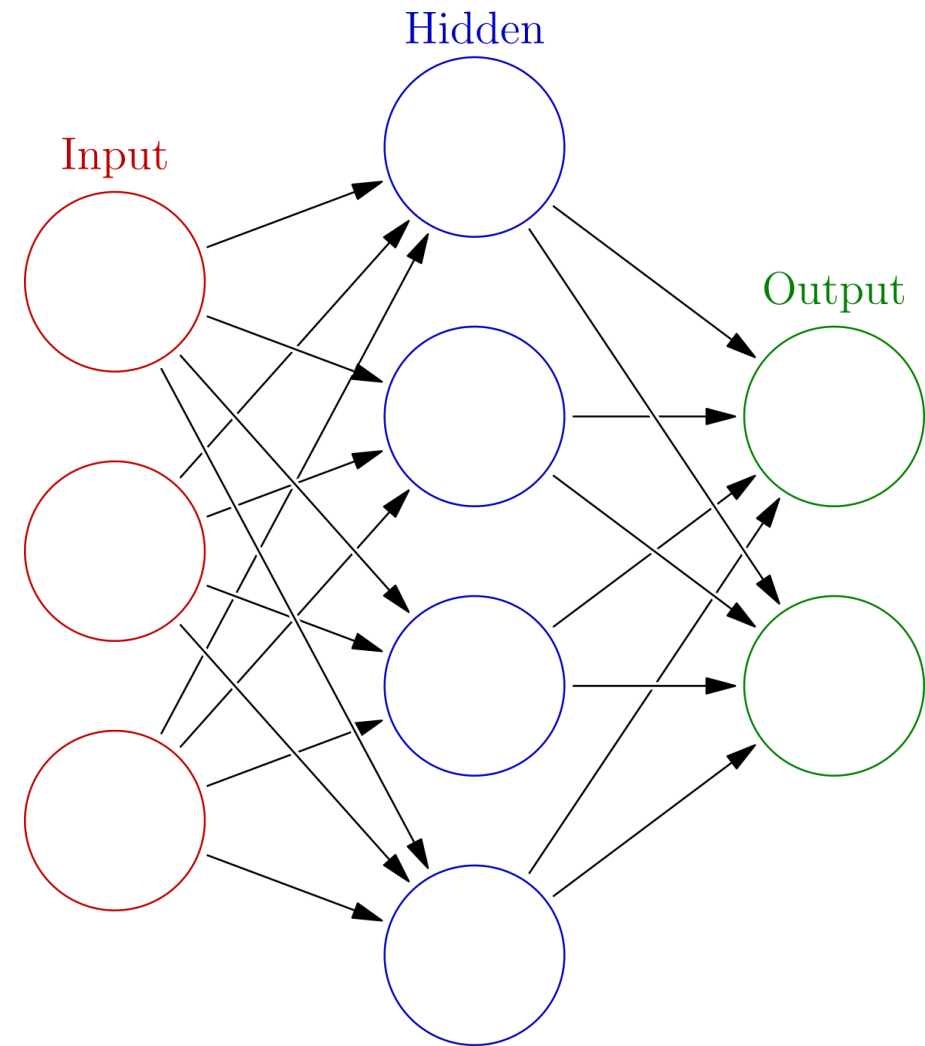
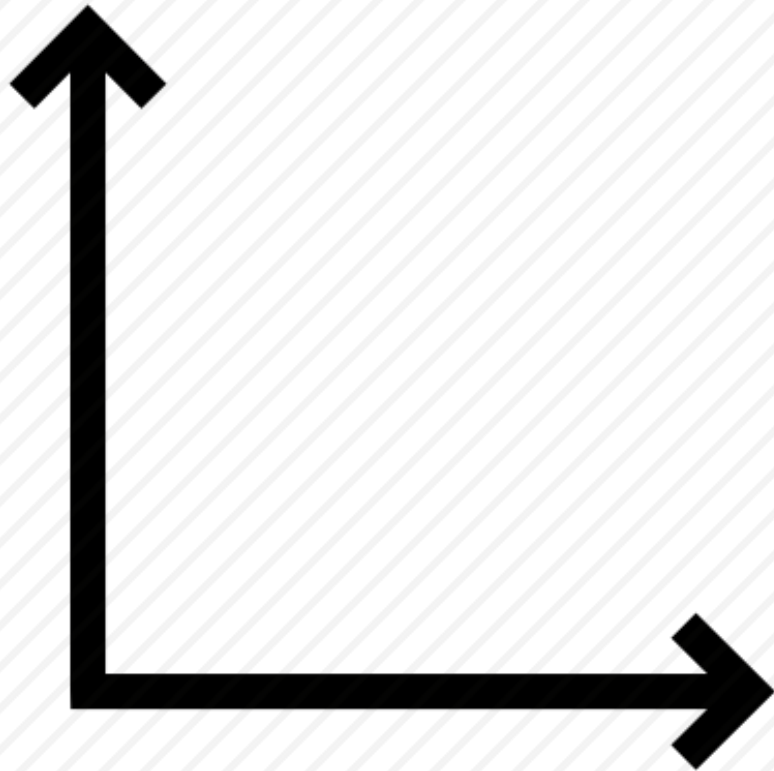
만약 우리에게 데이터가 더 있다면?



만약 우리에게 데이터가 더 있다면?
만약 데이터를 근사할 수 라도 있다면?

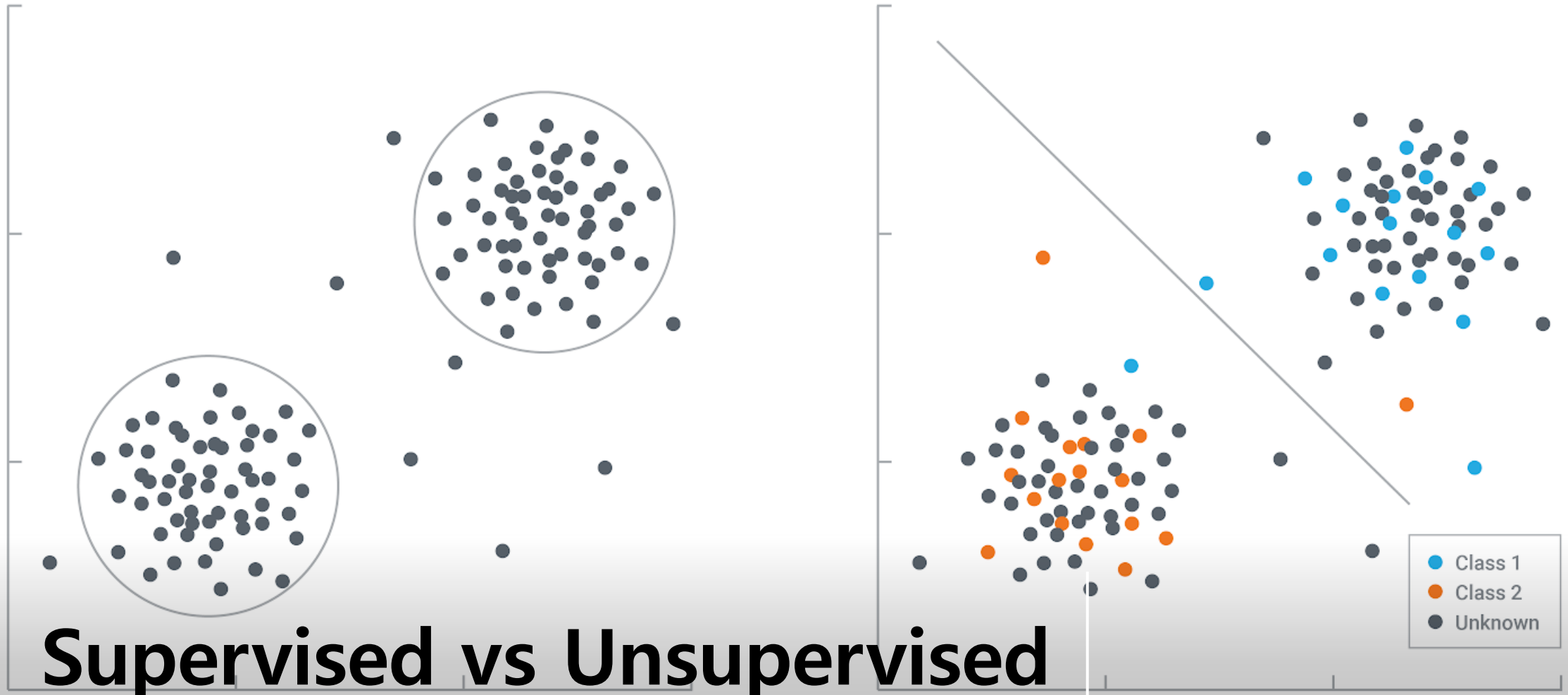


어디서 많이 보던 친숙한 그림



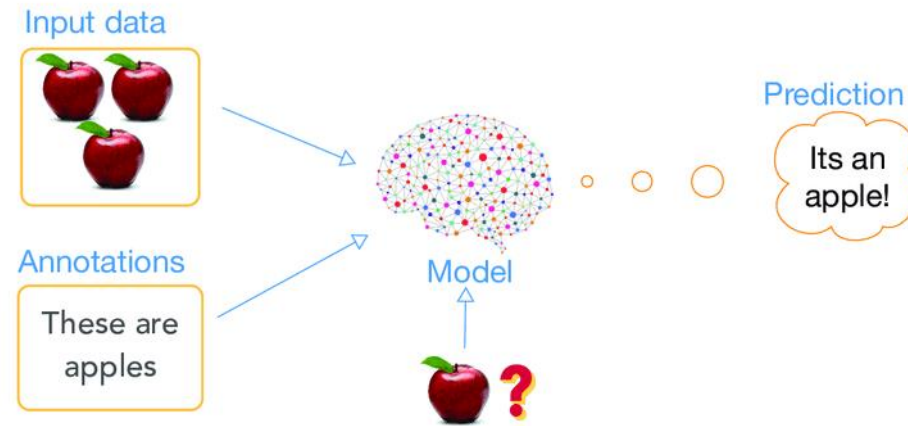
UNSUPERVISED

SUPERVISED

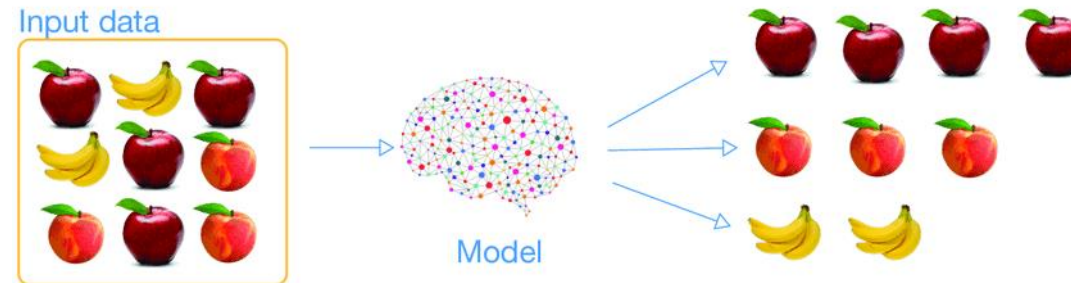


Supervised vs Unsupervised

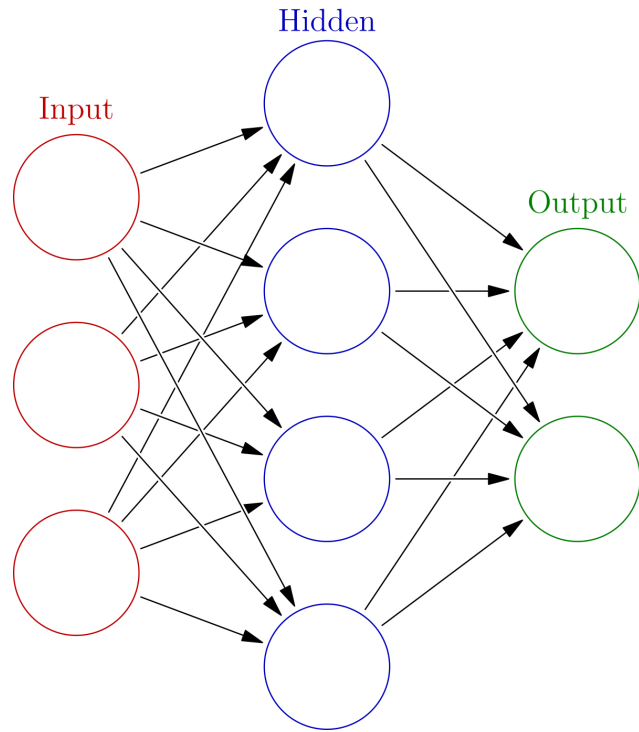
supervised learning

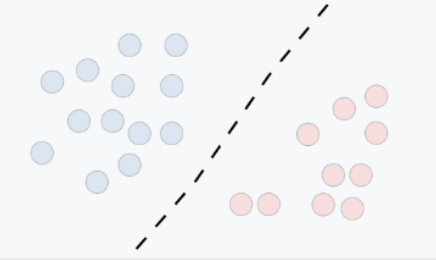
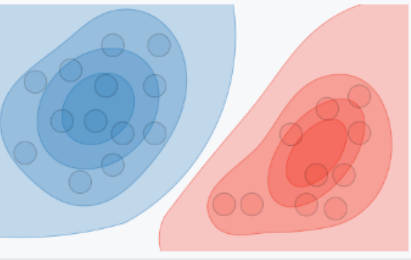


unsupervised learning



Supervised vs Unsupervised



What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Discriminative model vs Generative model

시계열과 머신러닝

시계열에 머신러닝을 적용할 때 생기는 문제

일반적인 데이터의 모델 입/출력을 수학적으로 표현하면?

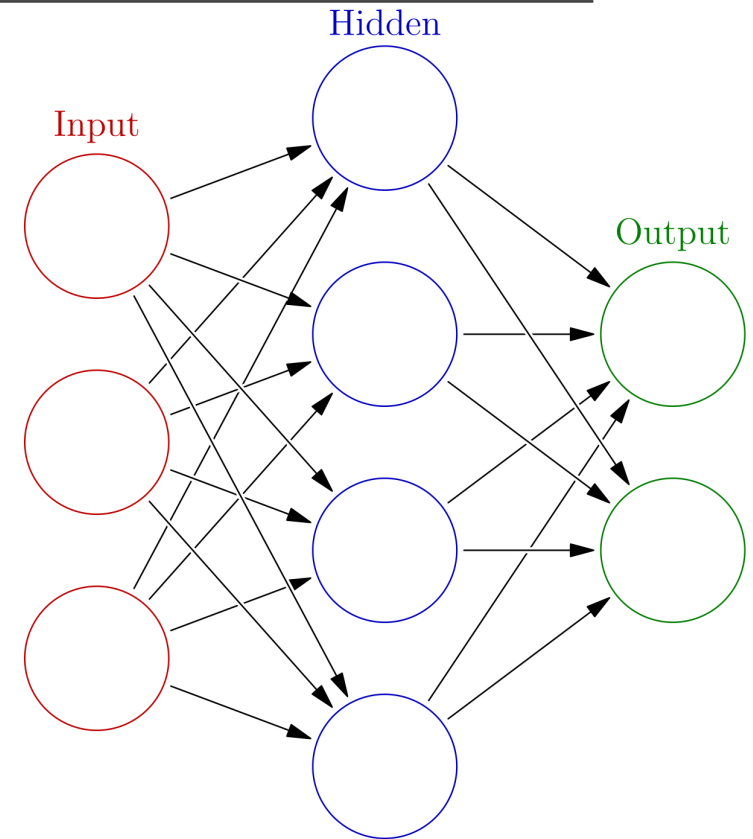
- $f(x) = y$

시계열 데이터의 모델 입/출력을 수학적으로 표현하면?

- $f(x) = y, \quad \text{when } t = 0$

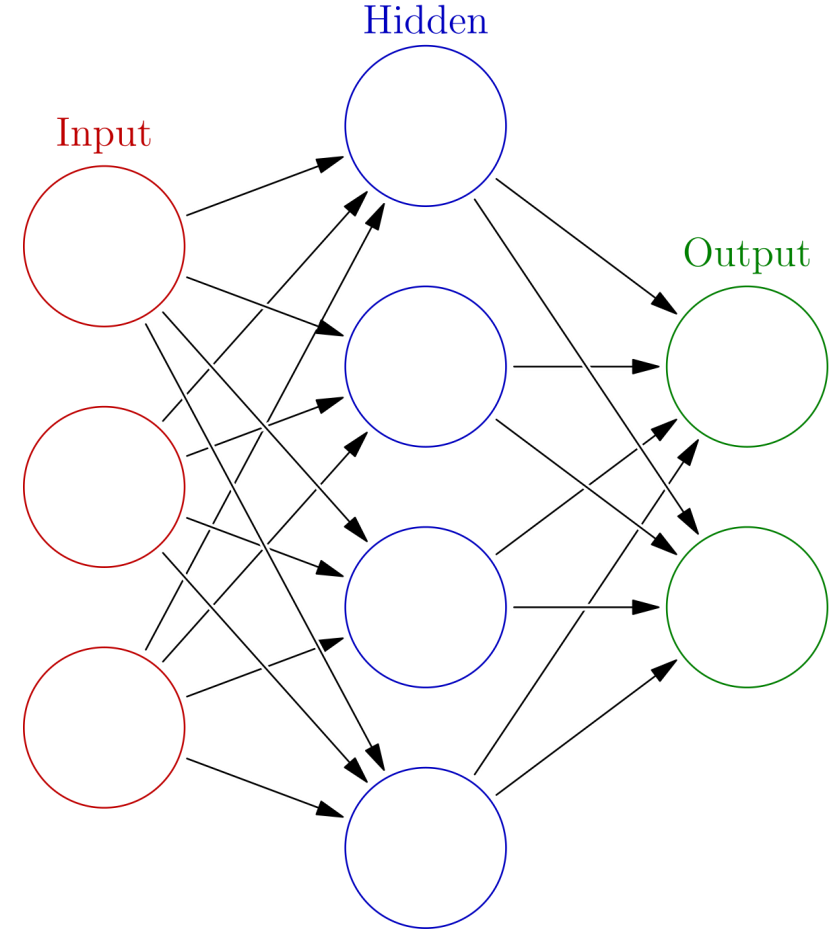
- $f(x) = z, \quad \text{when } t = 1$

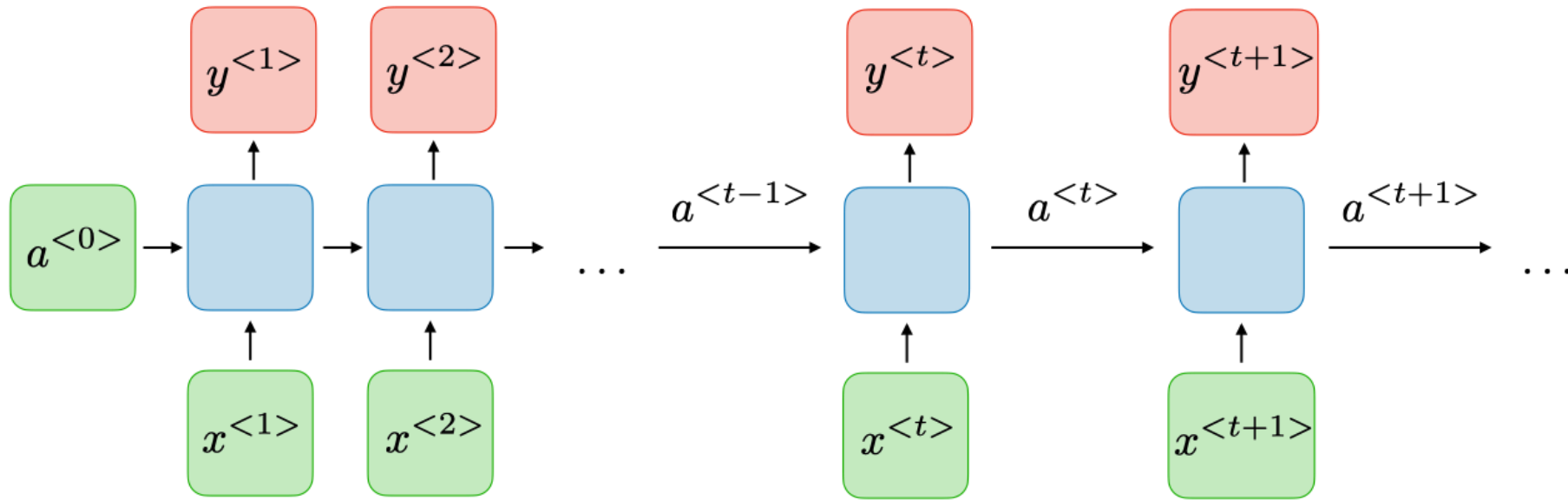
- “시간의 흐름에 따라서 관측치가 변한다”



시계열에 머신러닝을 적용할 때 생기는 문제

“시간”이라는 도메인을 어떻게
모델에 반영할까?

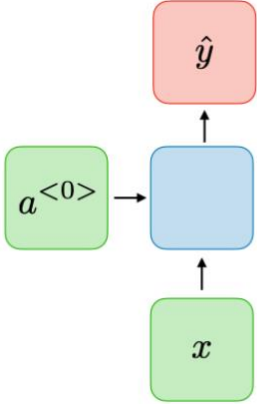
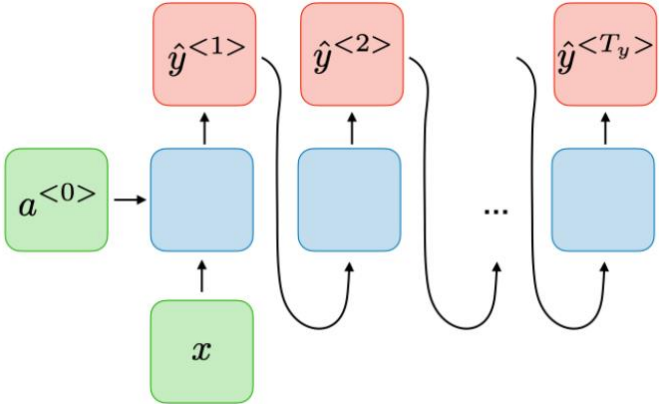




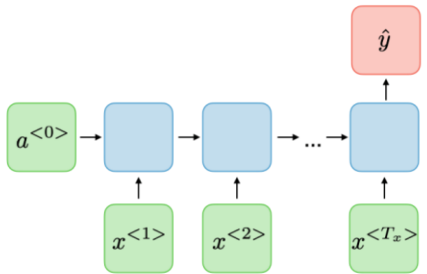
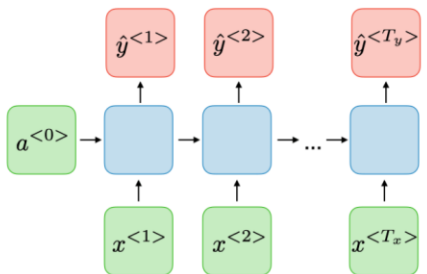
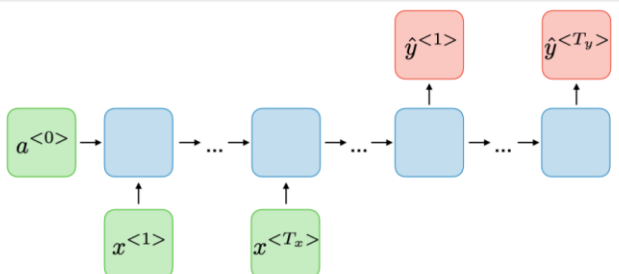
시계열에 머신러닝을 적용할 때 생기는 문제

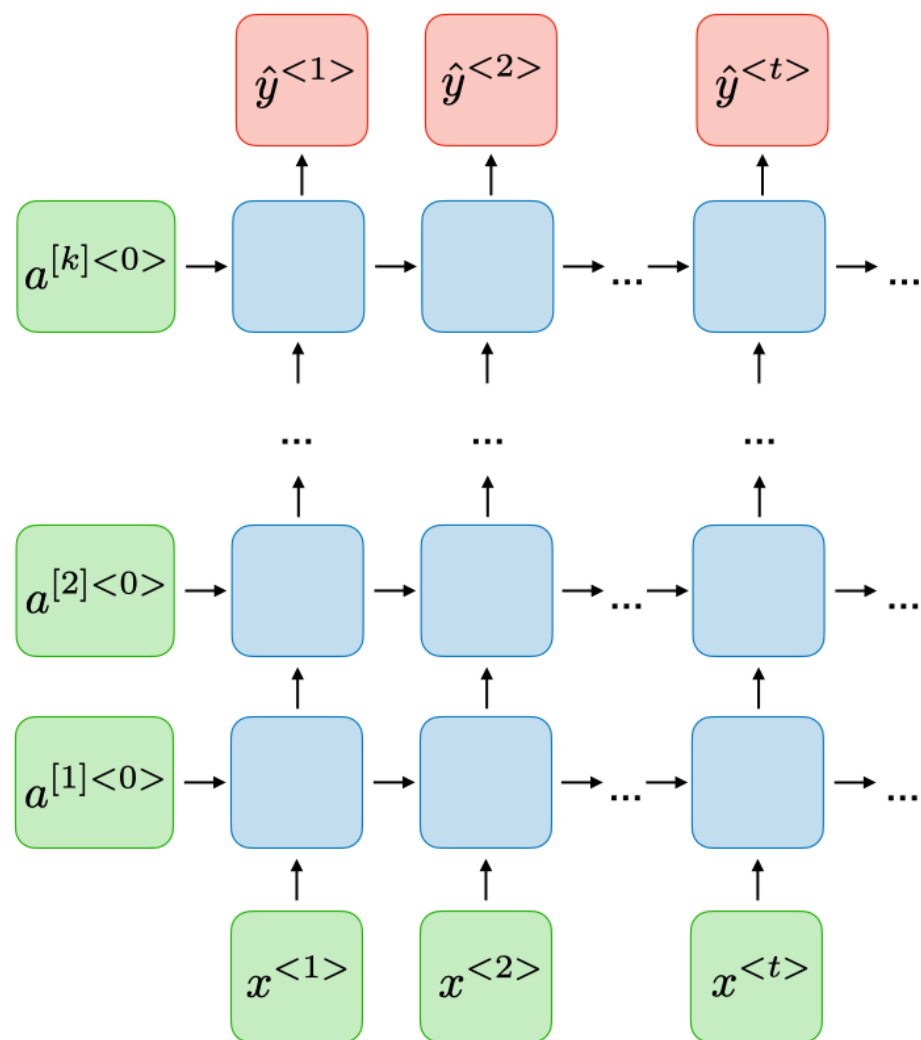
RNN의 개념

RNN의 종류

<p>One-to-one $T_x = T_y = 1$</p>		<p>Traditional neural network</p>
<p>One-to-many $T_x = 1, T_y > 1$</p>		<p>Music generation</p>

RNN의 종류

Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation



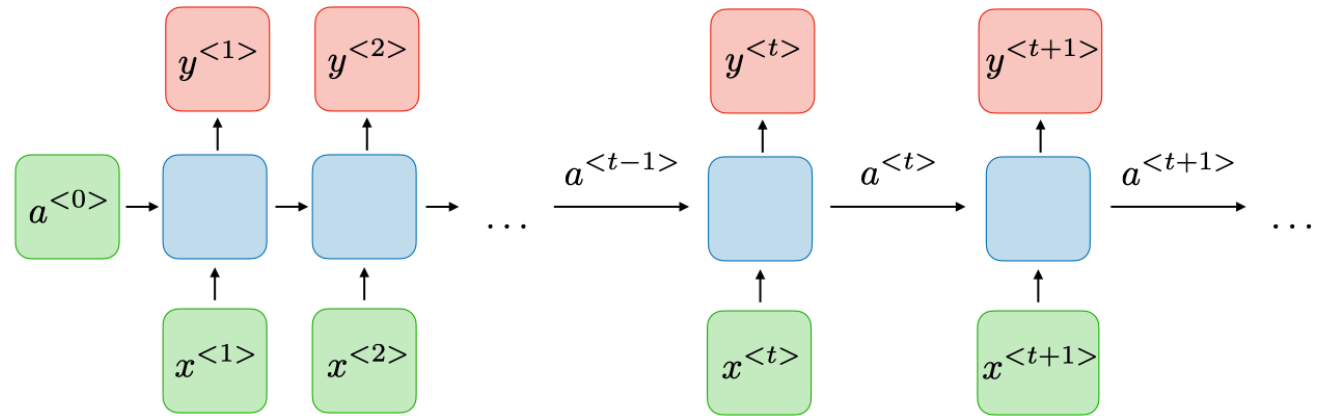
DRNN (Deep RNN)

RNN의 바리에이션

Vanila RNN

Bidirectional (양방향)

GRU, LSTM (장기 의존)



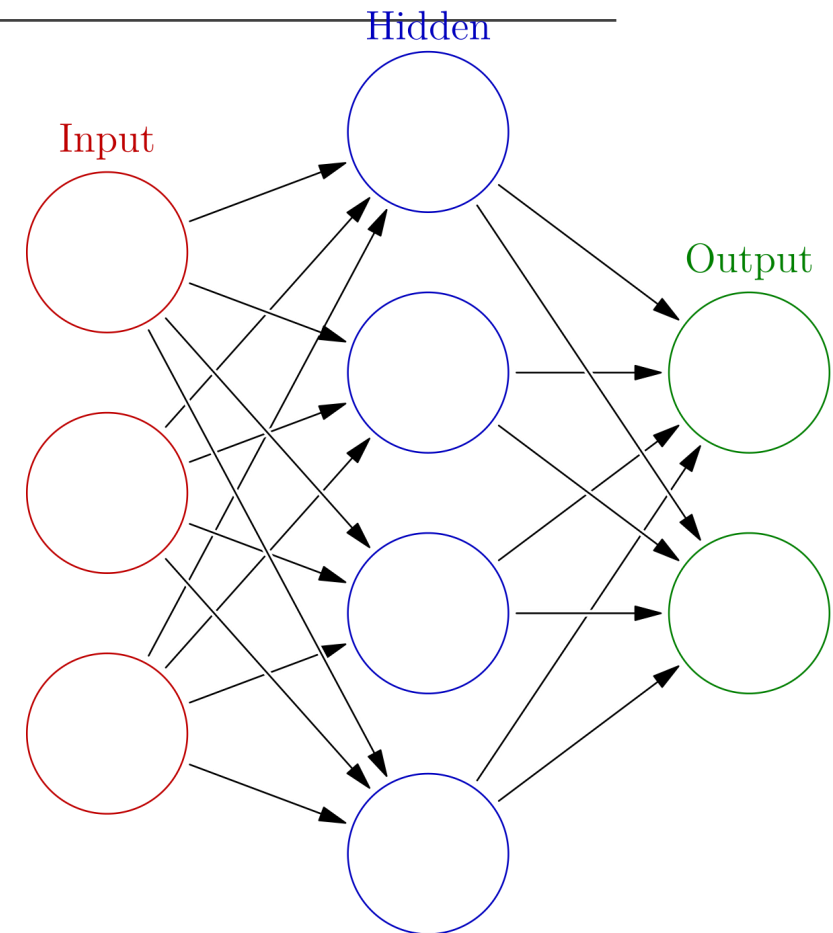
모든 RNN이 가진 공통점

- 특정 시점 T 의 값은 과거 / 미래 데이터의 출력에 의해 영향을 받는다
- 이 명제가 거짓인 데이터에는 절대 RNN 계열의 알고리즘을 무턱대고 적용해서는 안됩니다
- 예) 주가 데이터의 방향

CNN (Convolutional NN)

합성곱신경망 (CNN)

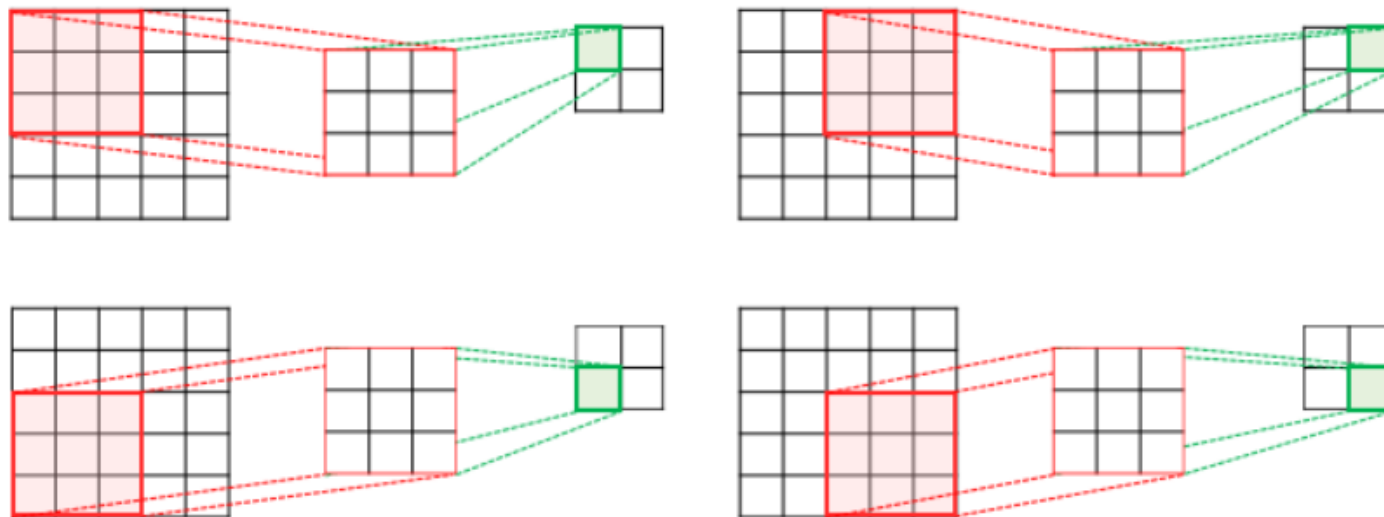
- 이미지를 인식하는 문제를 뉴럴넷으로 푼다면 어떻게 풀까?



CNN (Convolutional NN)

합성곱신경망 (CNN)

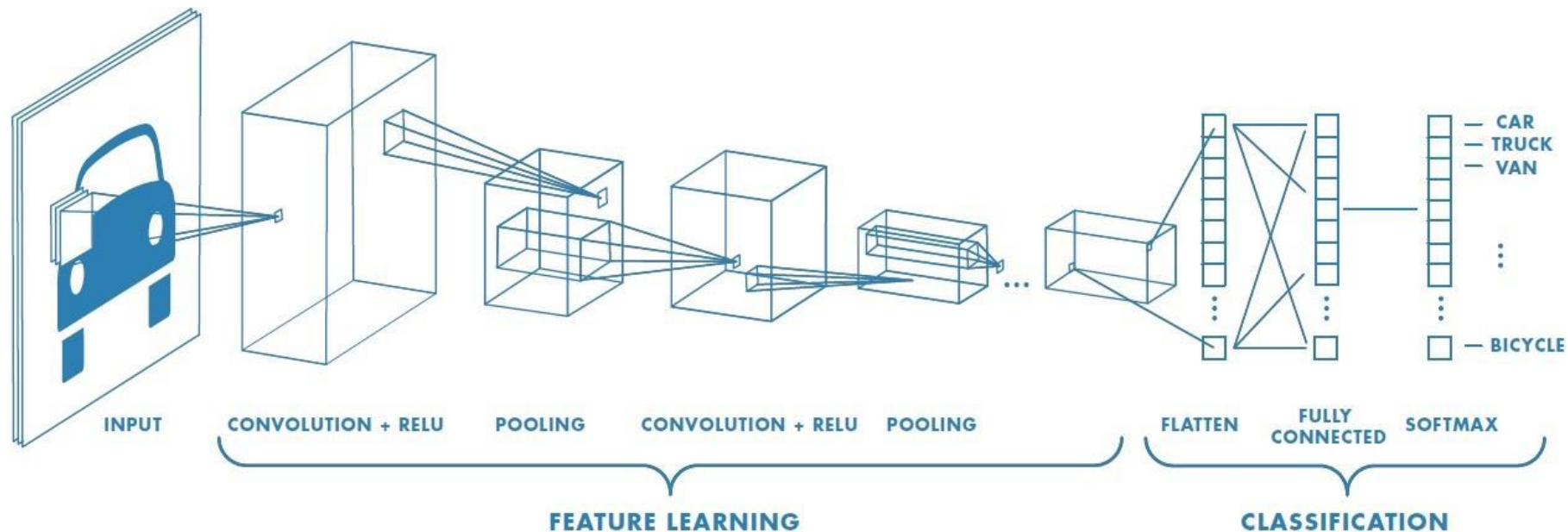
- 합성곱 (Convolution) 연산을 통해 입력값을 요약 표현하는 기법



CNN (Convolutional NN)

합성곱신경망 (CNN)

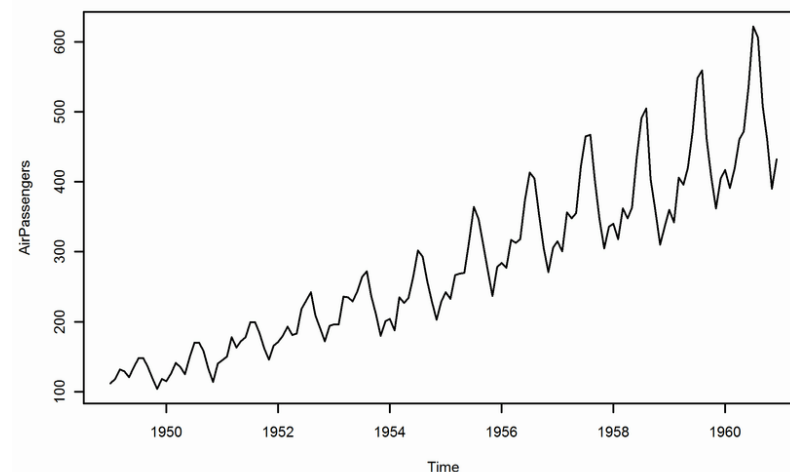
- 합성곱 (Convolution) 연산을 통해 입력값을 요약 표현하는 기법



CNN (Convolutional NN)

합성곱신경망 (CNN)

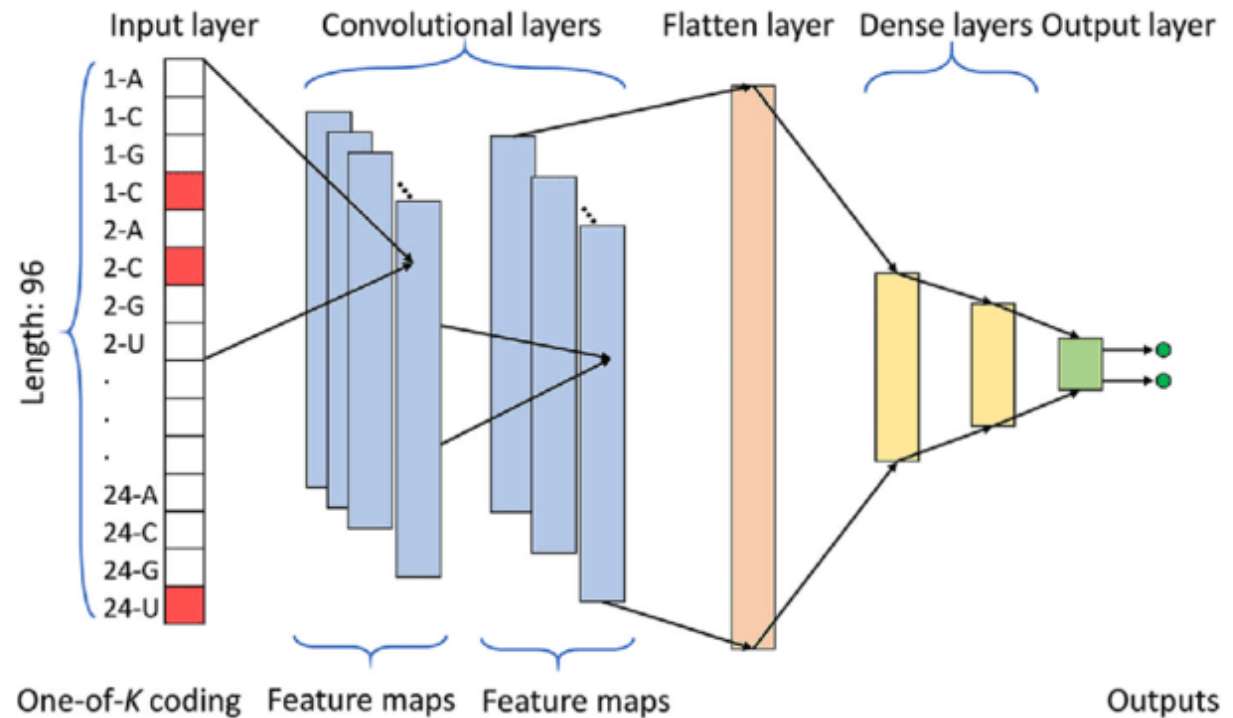
- CNN은 결국 데이터의 내제된 local한 특징을 추출하는 역할을 한다
- 시계열은 어떤가?
 - 시간의 흐름에 따라 **평균과 분산**이 변한다 => 로컬한 영역마다 서로 다른 통계적 특징을 갖는다
 - CNN을 통해 이러한 Local한 특징을 추출할 수 있지 않을까?



CNN (Convolutional NN)

합성곱신경망 (CNN)

- 1D CNN의 개념



Thank You!
