

## 제2장: 딥러닝의 구조

**Soyoung Park**

Pusan National University  
Department of Statistics

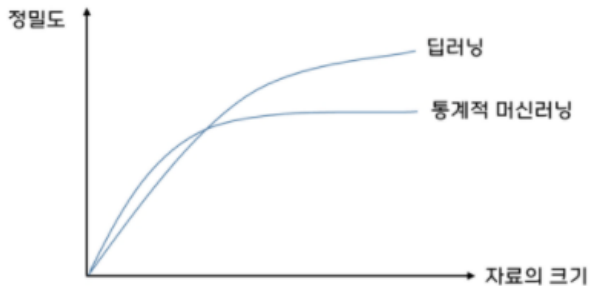
**`soyoung@pusan.ac.kr`**

# 통계적 머신러닝과 딥러닝

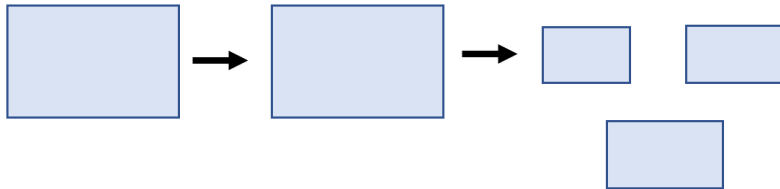
# 통계적 머신러닝과 딥러닝

구분	통계적 머신러닝	딥러닝
데이터의 크기		
분석자료 형태		
강점을 갖는 자료		
특성변수		
특성변수의 정규화 및 표준화		
모형		
최적화		
해석여부		
하드웨어		
실행요구시간		

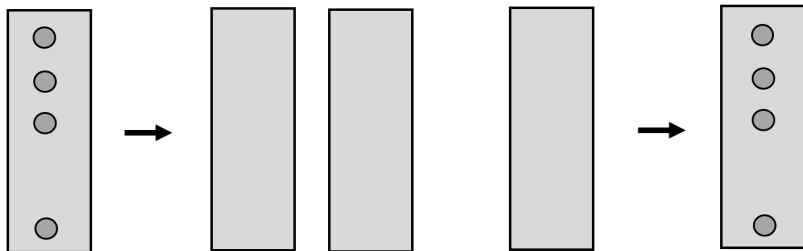
# 통계적 머신러닝과 딥러닝



# 머신러닝의 구조



# 딥러닝의 구조



# 선형결합과 비선형결합

# 딥러닝, 머신러닝의 4가지 영역

## 지도학습 (supervised learning)

- 특성변수와 이에 대응하는 해답을 입력데이터로 사용,
- 특성변수로 해답을 맞추는 예측모형을 만듦
- ex) 개, 고양이 사진 분류, 객체인식, 번역 및 통역 등

## 비지도학습 (unsupervised learning)

- 특성변수는 있으나 이에 대응하는 해답이 없는 학습방법
- ex) 트윗이나 구매 내역에 따른 고객 분류, 흐릿한 영상(이미지)의 복원 등



# 딥러닝, 머신러닝의 4가지 영역

## 스스로 지도학습 (self-supervised learning)

- 특성변수와 특성변수를 이용하여 만든 해답이 입력 자료로 사용되는 학습방법
- 해답에 사람이 관여하지 않음
- ex) 모네그림 재현, 여러 사람의 이미지로부터 새로운 사람의 이미지를 생성,
- ex) 현재의 영상이나 문장으로 다음 영상 또는 단어를 예측하는 모형으로부터 새로운 영화를 만들거나 소설 작성 등

## 강화학습 (reinforcement learning)

- 주어진 환경에서 최대의 보상을 받는 행동(action)을 학습
- ex) 게임에서 최대의 점수를 내는 게임전략을 만듦, 주식 등의 투자전략

# 딥러닝의 개념

## Basic concepts

- 전체데이터=학습(training)+검증(validation)+시험(testing)데이터
- 학습데이터를  $k$ 개로 분할(partition)하고, 이를 batch라고 한다.
- 학습데이터의 샘플은 여러개로 묶여서 학습 되는데 이 샘플의 묶음을 배치(batch)라고 함.
- 데이터를 batch로 나누는 이유는, 트레이닝 데이터를 통째로 신경망에 넣으면 비효율적인 리소스 사용으로 학습 시간이 오래 걸리기 때문
- 모든 학습 데이터를 1회 학습하는 것을 1에포크(epoch)라고 함.
- 1 epoch 에 이용되는 훈련 데이터는 여러 개의 batch로 분할됨.

<sup>1</sup><https://blog.naver.com/youseok0/221572211885>

# 데이터의 분할

# 딥러닝의 분석절차

# 딥러닝의 분석절차

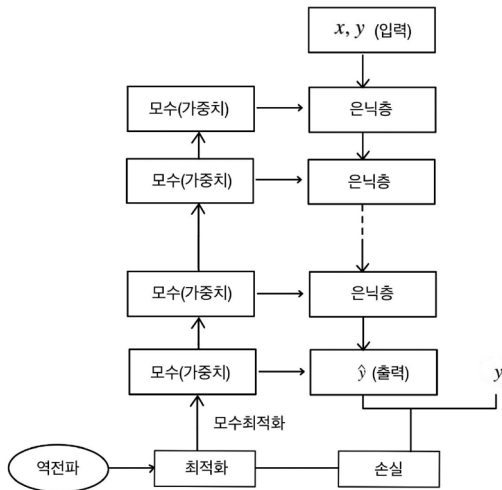


그림 2-1 딥러닝의 개념

# 딥러닝의 순서

- 1 딥러닝 모형 구축
- 2 딥러닝 모형의 모수에 임의의 값 할당
- 3 학습데이터로부터 한 batch의  $X$ 와  $y$ 를 임의 추출 (일반적으로 batch는 16 또는 32의 표본)
- 4  $X$ 를 딥러닝 모형에 입력하여  $y$ 를 예측 후, 손실함수와 정밀도 계산
- 5 모수에 대해 손실함수의 미분값을 계산한 후, 손실함수 미분값의 음의 방향으로 모수값 최신화(역전파)
- 6 2-5를  $k$ 개의 모든 batch에 대해 반복  $\rightarrow$  1-epoch 완성
- 7 2-6을 반복하여 모형의 정밀도를 원하는 수준만큼 올리고, validation set을 통해 초모수 조절 또는 모수에 제한 조건을 부여하여 모델 튜닝
- 8 testing set을 이용하여 모형의 overfitting을 점검
- 9 과대적합 발생시, 모수줄이기, 모수의 규제화, dropout등의 방법으로 해결

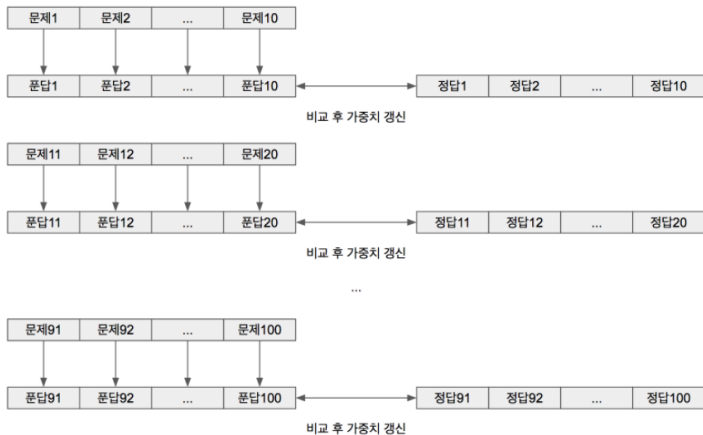
# Example

문제지	문제1	문제2	문제3	문제4	문제5	문제6	...	문제100
해답지	정답1	정답2	정답3	정답4	정답5	정답6	...	정답100

- $x$ : 입력데이터 - 100문항의 문제지
- $y$ : 라벨값 - 100문항의 답
- batch size: 몇 문항을 풀고 해답을 맞출지를 의미함
- 배치사이즈가 100문항이라면, 전체 100문항을 다 풀고 해답을 맞춰보는것 → 해답을 맞추고 나면 오차를 줄이기 위해 '역전파'를 이용하여 알고리즘의 가중치가 갱신됨.

## batch size

- 배치사이즈 = 10





## batch size

- 배치사이즈 = 1
- 배치사이즈가 작을수록 가중치 갱신이 자주 일어남.

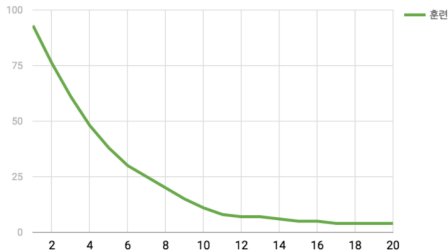


# batch size

- 배치사이즈 = 전체 훈련 데이터 수
  - 1 epoch (전체 데이터를 한번다보고)마다 가중치와 편향을 수정.
  - 일반적으로 안정된 학습이 진행되고 다른 두 종류의 학습에 비해 속도는 빠르지만 국소 최적해에 빠지기 쉬움.
- 배치사이즈 = 1
  - 개별 샘플마다 가중치와 편향을 수정.
  - 안정성이 떨어지지만 국소 최적해에 빠지는 것을 방지할 수 있음.
- 배치사이즈 = 최적의 수 (Mini batch)
  - 훈련데이터를 작은 그룹(batch)로 분할하고 그룹마다 가중치와 편향을 수정.
  - 무작위로 배치를 선택함 → 상대적으로 국소 최적해에 잘 빠지지 않음

# epoch

- epoch는 모의고사 1회분을 몇번이나 반복해서 풀어보는지를 말함.
- 에포크가 20이면 모의고사 1회분을 20번 푸는것을 말함. 같은 문제집을 여러번 풀면서 반복적으로 가중치를 갱신하고 모델이 학습됨.
- 딥러닝에서 epoch는 전체 트레이닝 셋이 신경망을 통과한 횟수 의미, 즉 1-epoch는 전체 트레이닝 셋이 하나의 신경망에 적용되어 순전파와 역전파를 통해 신경망을 한 번 통과했다는 것을 의미함.



- x축이 에포크, y축이 틀린 개수

# epoch

- 에포크를 늘린다고 무조건 좋을까?
  - 하나의 문제집만 계속 학습하면 오히려 역효과가 발생할 수 있음, 연습한 문제는 잘 풀더라도 새로운 문제는 잘 풀지 못함 → overfitting 문제
  - 에포크 증가에 따른 loss의 정도를 보면서, overfitting조짐이 보이면 학습을 중단해야함.

## 통계적모형과 오차항

- 통계적 모형: 최소한의 모수로 모형을 만들어 데이터의 특성을 파악
  - 오차항을 최소화하는 방법
- 딥러닝에서는 오차를 최대로 줄인다고 무조건 좋은것이 아님
  - 딥러닝에서는 오차항이 일반화를 가능케하는 중요한 성질이 있기 때문
  - 오차항을 줄여 모형의 임의성을 제한하면 → 과대적합발생 → 일반화실패
  - (1) 데이터 증가, (2) 모수의 규제화, (3) drop out 등으로 모형의 임의성 부여

# iteration

- Iteration은 1-epoch를 마치는데 필요한 미니배치 갯수를 의미
- 1-epoch를 마치는데 필요한 파라미터 업데이트 횟수
- 각 미니 배치마다 파라미터 업데이터가 한번씩 진행되므로 iteration은 파라미터 업데이트 횟수이자 미니배치 갯수
  - 예를 들어, 700개의 데이터를 100개씩 7개의 미니배치로 나누었을 때,
  - 1-epoch를 위해서는 7-iteration이 필요하며 7번의 파라미터 업데이트가 진행됨

# 데이터의 실수화

- 데이터는 실수화되어야함
- 데이터의 형태가 문장이나 사람과 동물같은 이미지같은 비정형 데이터의 경우에도 실수로의 전환이 필요.
- 자료의 벡터화(vectorization)이라고도 함
- ex) One-Hot Encoding
  - 3개의 클래스 A,B,C
  - A를 (1,0,0), B를 (0,1,0), C를 (0,0,1)로 벡터화

# One-Hot Encoding

- 단어 집합의 크기를 벡터차원으로
- 표현하고 싶은 단어의 인덱스에 1의 값을 부여, 다른 인덱스에는 0을 부여하는 단어의 벡터표현 방식
- 희소표현 (Sparse Representation), 희소 벡터(Sparse Vector)
- 각 단어에 고유한 인덱스 부여 (정수 인코딩)
- 단어 간 유사도 계산 불가

Label Encoding			One Hot Encoding			
Food Name	Categorical #	Calories				
Apple	1	95	Apple	Chicken	Broccoli	Calories
Apple	1	95	1	0	0	95
Chicken	2	231	0	1	0	231
Broccoli	3	50	0	0	1	50





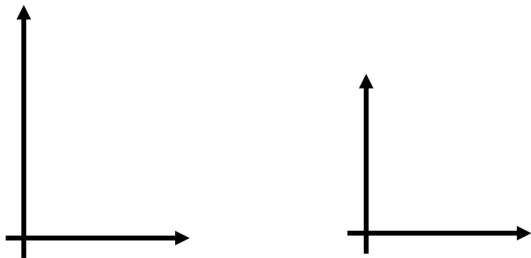
## 데이터의 정규화(standardization)

- 한 변수의 절대값이 크고, 다른 한 변수의 절대값이 작은 경우 변수의 중요도에 관계없이, 절대값이 큰 변수가 활성화 함수와 손실 함수의 변동에 민감하게 반응하는 결과를 가져다 줌
- 이는 손실 함수를 최소화하는 최적화에 실패하는 주요한 원인이 됨
- 특성 변수들의 분산이 다른 경우에도 동일한 이유로 문제가 됨
- → 이를 해결하기 위해, 평균이 0, 분산이 1이 되도록 정규화 변환 실시!

$$\frac{x_j - \bar{x}}{sd(x_j)}$$

## 데이터의 정규화(standardization)

- 머신러닝에서 학습에 사용되는 경사하강법(Gradient descent)의 예시



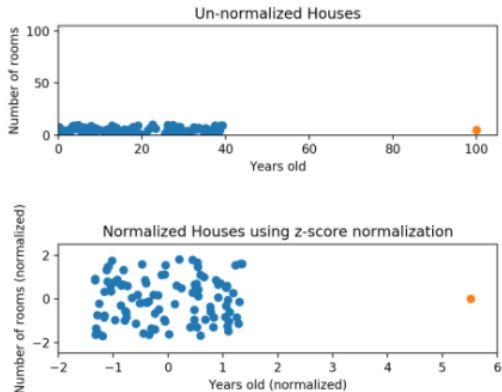
- 두 변수의 스케일이 다른 경우, 등치선도가 타원형을 그림
- 학습률이 동일하다는 가정하에 X1의 갱신 속도가 X2보다 빠름
- 비교적 많은 반복과정을 거쳐야 최적 해를 구할수 있음
- 오른쪽그림처럼 정규화를 거치면 X1과 X2의 갱신 속도가 동일해지며 더 빠르게 최적 해를 구할 수 있음.

## 데이터의 표준화(standardization)

- 딥러닝에서는 비선형변환인 활성화함수(activation function)을 사용하여 대부분의 활성화함수는 0-1 사이의 값을 출력한다.
- 이러한 이유로 특성변수를 0-1사이 값으로 변환하며 이를 특성변수의 표준화(standardization)라고 한다.
- 표준화는 손실함수의 최적화에 긍정적인 역할을 함.

$$\frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$$

# 데이터의 표준화(standardization)



<sup>1</sup><https://blog.naver.com/leeheung9/222421468148>

## 데이터에 결측치는 No!

- 딥러닝에 사용되는 특성변수에는 결측치가 없어야 함
- 일반적으로 결측치는 평균, 중위수로 대체하지만 딥러닝에서는 0으로 대체하는것이 가장 안전함
- 모수추정에서 사용하는 역전파에서 특성변수값 0은 모수의 최신화 (update)에 기여하지 않기 때문
- 시험데이터에 결측치가 있지만, 학습데이터에는 없다면 학습데이터의 일부를 0으로 대체해야함 (0으로 대체하는 특성변수는 결측이 발생한 특성변수여야함)

# 데이터의 형태

## 텐서(tensor)

- 데이터는 텐서(행렬의 일반화)로 입력되고, 출력됨
- 0차원 텐서: 스칼라
- 1차원 텐서(1D 텐서): 벡터
- 2차원 텐서(2D 텐서): 행렬
- 3차원 텐서(3D 텐서): 3개의 축으로 구성된 데이터
- 4차원 텐서(4D 텐서): 4개의 축으로 구성된 데이터

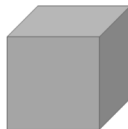
# tensor



vector



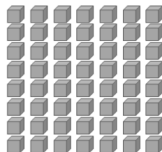
matrix



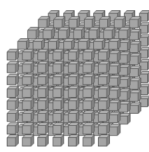
3d-tensor



4d-tensor



5d-tensor



6d-tensor

<sup>1</sup><https://rekt77.tistory.com/102>

# tensor



# 데이터의 형태

## 텐서(tensor)

- 텐서는 데이터를 담는 container라고 볼 수 있음
- 벡터데이터: (samples, features)
- 시계열데이터: (samples, timesteps, features)
- 이미지: (samples, height, width, channels) or (samples, channels, height, width)
- 동영상: (samples, frames, height, width, channels) or (samples, frames, channels, height, width)

## 2D텐서

- 2D텐서 데이터 (표본수=  $n$ , 특성변수수=  $p$ )로 표현
- ex) 100명의 사람 각각에 대한 연령, 성별, 수입 자료이면 (100,3)인 2D텐서
- 표본수를 생략하고 (3,)로 입력하기도 함

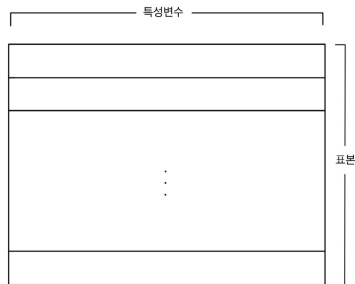


그림 2-2 2D텐서 데이터

## 3D텐서

- 3D텐서 데이터 (표본수 =  $n$ , 시간스텝 =  $k$ , 특성변수수 =  $p$ )로 표현
- 시간의 순서나 단어의 순서가 자료분석에 중요한 정보를 가진 경우
- ex) 1분마다 특정 주식의 가격과 1분동안 최대가격, 최소가격이 관측되면 하루 거래 시간 기준으로 390분 동안의 데이터를 관측하고, 이를 100일간 모았다.
- 표본수 100, 시간스텝 390, 특성변수 수 3: (100, 390, 3)
- 자료의 시간 순서가 중요한 자료!

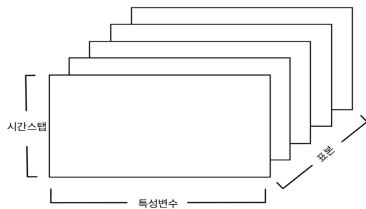


그림 2-3 3D텐서 데이터

## 3D텐서

- 문장을 이해하는데 단어의 순서는 매우 중요한 역할!
- 100,000개의 문서에서 가장 많이 사용되는 단어 5,000개 추출
- 각 문서 최초 10,000개의 단어에 대해 one-hot encoding
- 100,000개의 표본수, 10,000개의 시간스텝, 5000개의 특성변수:

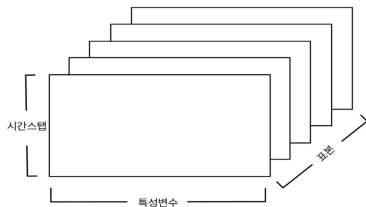


그림 2-3 3D텐서 데이터

# 4D텐서

- 4D텐서 데이터 (표본수 =  $n$ , 채널수 =  $c$ , 높이 =  $h$ , 너비 =  $w$ ) 또는 (표본수 =  $n$ , 높이 =  $h$ , 너비 =  $w$ , 채널수 =  $c$ )로 표현
- 이미지데이터로 대표됨
- ex)  $78 \times 78$  픽셀 컬러 이미지가 5000장 있으면 (5000, 78, 78, 3)
- 흑백이미지라면 (5000, 78, 78, 1)

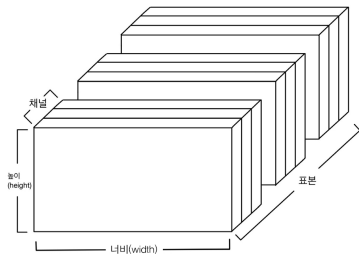
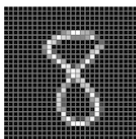


그림 2-4 4D텐서 데이터

MNIST image

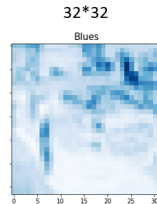
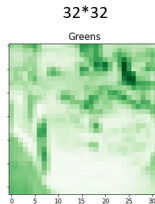
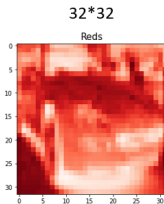
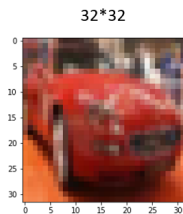


28 x 28  
784 pixels

[illegible]

<sup>1</sup>[https://ml4a.github.io/ml4a/neural\\_networks/](https://ml4a.github.io/ml4a/neural_networks/)

# CIFAR10 image



## 텐서별 신경망 특징

- 2D텐서로 저장된 벡터는 보통 fully connected layer나 densely connected layer로 처리
- 3D텐서로 저장된 시퀀스데이터는 보통 recurrent layer로 처리
- 4D텐서로 저장된 이미지 데이터는 Conv2D layer로 처리됨



# 데이터의 필수 가정

# 데이터의 필수 가정

- MLP의 1D텐서인 특성변수들은 서로 다른 표본에서 서로간에 독립
- CNN에서 3D텐서인 특성변수들은 서로 다른 표본에서 서로간에 독립
- RNN에서 하나의 표본에 2D텐서인 특성변수는 서로 다른 표본에서 서로간에 독립이며 추가적으로 시간스텝별로 주어진 특성변수는 최소한 정상성을 만족해야함
  - RNN에서 시계열 자료를 분석할 때, 이 정상성 만족여부를 분석 이전에 점검
  - 정상성을 만족하지 못 할 경우, 자료의 변환을 통해 정상성 만족하도록
  - (예외) 자연어자료에서는 정상성 체크 필요없음

## 데이터의 필수 가정

- 특성변수와 결합된 모수는 첨자(ex, i)가 걸려있지 않음
- 모수가 표본에 의존하지 않는다는것!
- 각 표본이 특성변수의 선형결합에 동일한 기여를 한다는 의미
- 표본이 같은 분포에서 나온 서로간에 독립이라는 의미
- 딥러닝 모형의 일반화에 있어 매우 중요한 요소