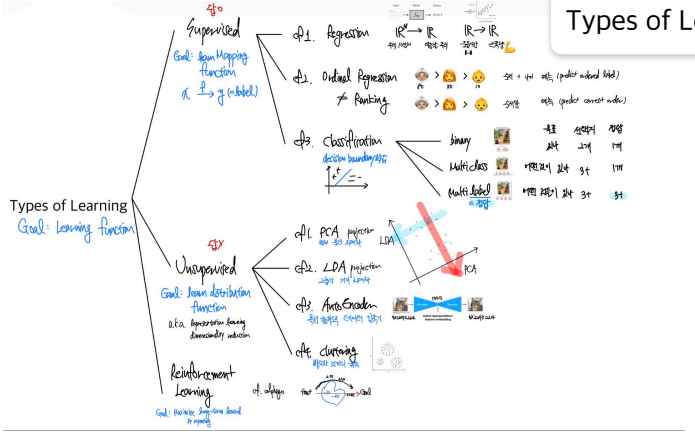
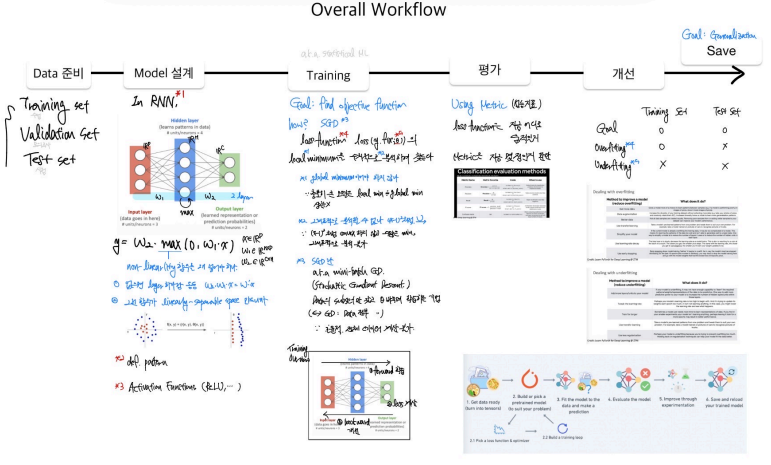
Lec1. Preliminary

Types of Learning

Linear Algebra and Numpy

Overall Workflow

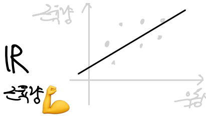




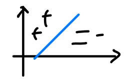
Types of Learning

Goal: Learning function

- 답O **Supervised**
  - Goal: learn Mapping function
  - $x \xrightarrow{f} y$ (=label)
  - ex1. Regression
    - $\mathbb{R}^N \to \mathbb{R}$ (주식 10일치 → 다음날 주식)
    - $\mathbb{R} \to \mathbb{R}$ (운동시간 → 근육량💪)



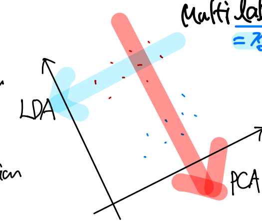
  - ex2. Ordinal Regression
    - 👵 > 👩 > 👶 (80, 30, 10) 순서 + 나이 예측 (predict ordered label)
    - ≠ Ranking: 👵 > 👩 > 👶 순서만 예측 (predict correct order)
  - ex3. Classification
    - decision boundary 학습



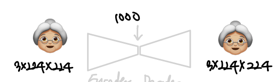
    - binary
    - Multiclass
    - Multi label = 정답

	목표	선택지	정답
binary (👵👶)	있나	2개	1개
Multiclass (👵👶)	어떤것이 있나	3+	1개
Multi label (👵👶)	어떤 것들이 있나	3+	3+

- 답X **Unsupervised**
  - Goal: learn distribution function
  - a.k.a. representation learning, dimensionality reduction
  - ex1. PCA projection — 전체 분산 최대화
  - ex2. LDA projection — 그룹간 거리 최대화

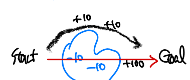


  - ex3. AutoEncoder — 복원 가능하도록 데이터 압축기



  - ex4. clustering — 비슷한 것끼리 묶음

- **Reinforcement Learning**
  - Goal: Maximize long-term Reward by repeating
  - ex. alphago



Overall Workflow

Data 준비 — Model 설계 — Training — 평가 — 개선 → Save

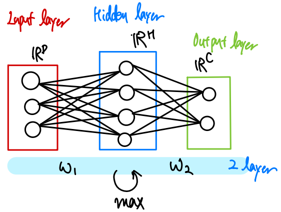
Data 준비

- Training set (수업)
- Validation set (모의고사)
- Test set (시험)

Model 설계

pick loss function, optimization / build training loop

In RNN.*1



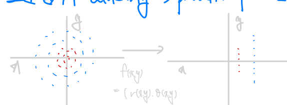
$$y = W_2 \cdot \max(0, W_1 \cdot x)$$

$x \in \mathbb{R}^P$, $W_1 \in \mathbb{R}^{H \times P}$, $W_2 \in \mathbb{R}^{C \times H}$

non-linearity 함수는 꼭 넣어야 한다.

① 없으면 layer 하나랑 동등 $W_2 \cdot W_1 \cdot x = W' \cdot x$

② 고차 함수가 linearly-separable space 만드니까



*2 def. pattern

*3 Activation Functions (ReLU, ⋯)

Training

a.k.a. statistical ML

Goal: find objective function

how? SGD*3

loss function*4 $loss(y, f(x; \theta))$*5 의 local minimum*1을 수치적으로*2 분석하여 찾는다

*1 global minimum이어야 하지 않나

∵ 충분히 큰 모델은 local min ≒ global min 상관X

*2 그래프적으로 분석할 수 없다 $(x-1)^2$처럼.

∵ $(x-1)^2$처럼 convex하지 않고 수많은 min, 그래프적으로 분석 불가

*3 SGD란

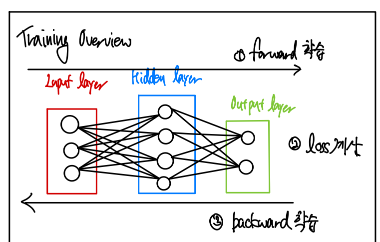
a.k.a mini-batch GD.

(Stochastic Gradient Descent.)

Data의 subset만 갖고 θ 바꾸며 학습하는 기법

(↔ GD : Data 전부 〃)

∵ 비효율적. 전체 데이터 계산 불가.



평가

Using Metric (성능지표)

loss function은 지금 얼마나 틀렸는지

Metric은 지금 몇 점인지 판단

cf. Accuracy, Precision, Recall, F1-Score, Confusion Matrix

개선

	Training Set	Test Set
Goal	O	O
Overfitting*4	O	X
Underfitting*5	X	X

*4 Overfitting Solution

- Get more data
- Data Augmentation
- Use transfer learning
- Simplify model
- Use learning rate decay
- Use early stopping

*5 Underfitting solution

- Add more layers/units
- Use less regularization
- Use transfer learning
- Train for longer
- Tweak the lr

(Matched)

Save

Goal: Generalization

부록

*1

p3 overall workflow > model

RNN이 항상 정답은 아님.
상황에 따라 더 유리한 모델 있음.
cf. RandomForest, NaiveBayes, SVM

*2

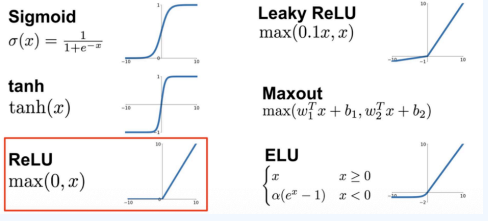
p3 overall workflow > model

아래 용어는 모두 비슷한 의미다.
pattern, embedding, weights,
feature represenation,
feature vectors

*3

p3 overall workflow > model

다음과 같은 activation function이 있다



*4

p3 overall workflow > Training

아래는 다양한 loss function이다.
MSE, MAE, CrossEntropy,
Negative log likelihood,
Connectionist temporal classification

*5

p3 overall workflow > Training

θ는 parameter이다
parameter는 함수에서 우리가 정할 수있는 값
optimization은 parameter를 바꾸는 작업

cf. $f(x;a,b,c) = ax^2+bx+c$

$\Rightarrow f(x;\theta) \quad \theta = [a,b,c]$