DeepIntoDeep

Machine Learning and MLP

발표자: 김성찬

Machine Learning and MLP

Seongchan Kim

Artificial Intelligence in Korea University(AIKU)

Department of Computer Science and Engineering, Korea University

발표자 소개

- 이름 - 김성찬
- 경력
  - 고려대학교 정보대학 컴퓨터학과 20학번
  - CVLAB 학부인턴 2022.01 ~ 2023.12
  - CVLAB 학석사 2024.01 ~
  - AIKU 0 ~ 1기 학회장
  - AIKU DeepIntoDeep 0 ~ 3기 강사
- 관심분야
  - NeRF
  - Video
  - Multimodal
  - 배구

Contents

Introduction

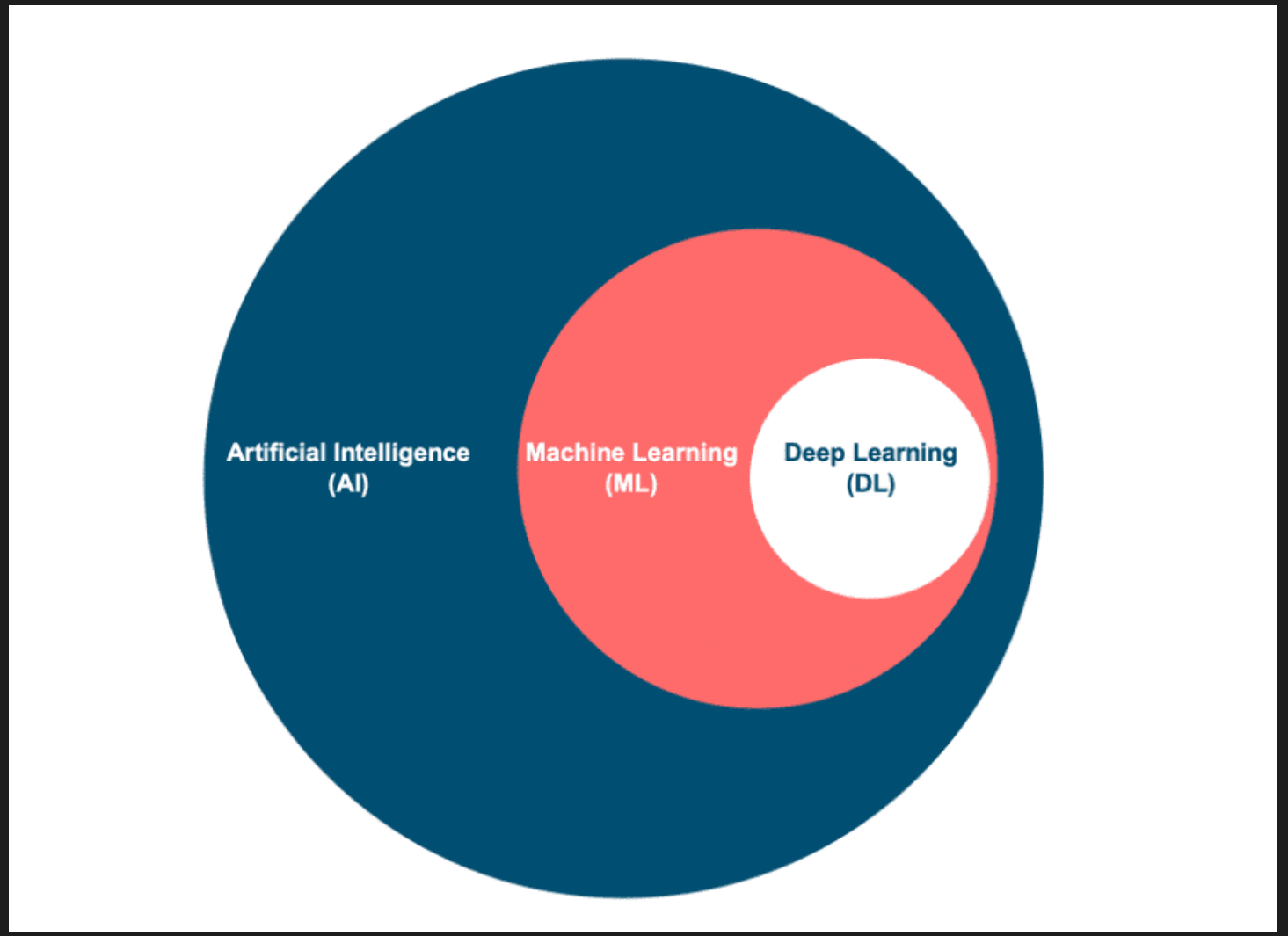
What is Artificial Intelligence?

지능의 요소

- 현상
- 현상에 대한 인식과 분석
- 시행착오와 학습
  - 인식
  - 판단 또는 결정
  - 상호작용

다가갈 수 없는 완전을 향해 다가가는 과정

AI, ML, DL



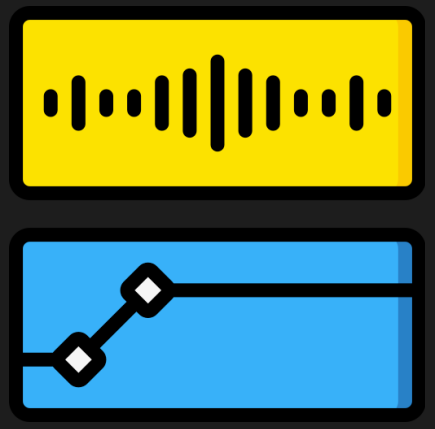
https://www.phdata.io/blog/data-science-terms-you-should-know-the-difference-between-ai-ml-and-dl/

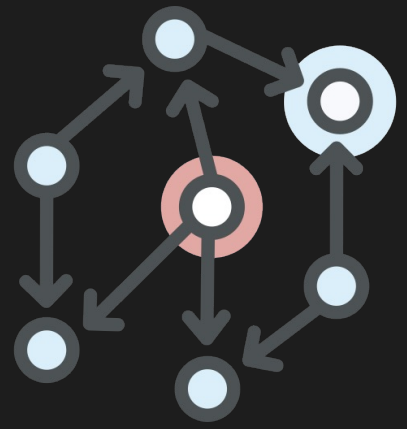
Machine Learning Key Components

- Data
- Model
- Objective
- Optimization Algorithm

Machine Learning Key Components

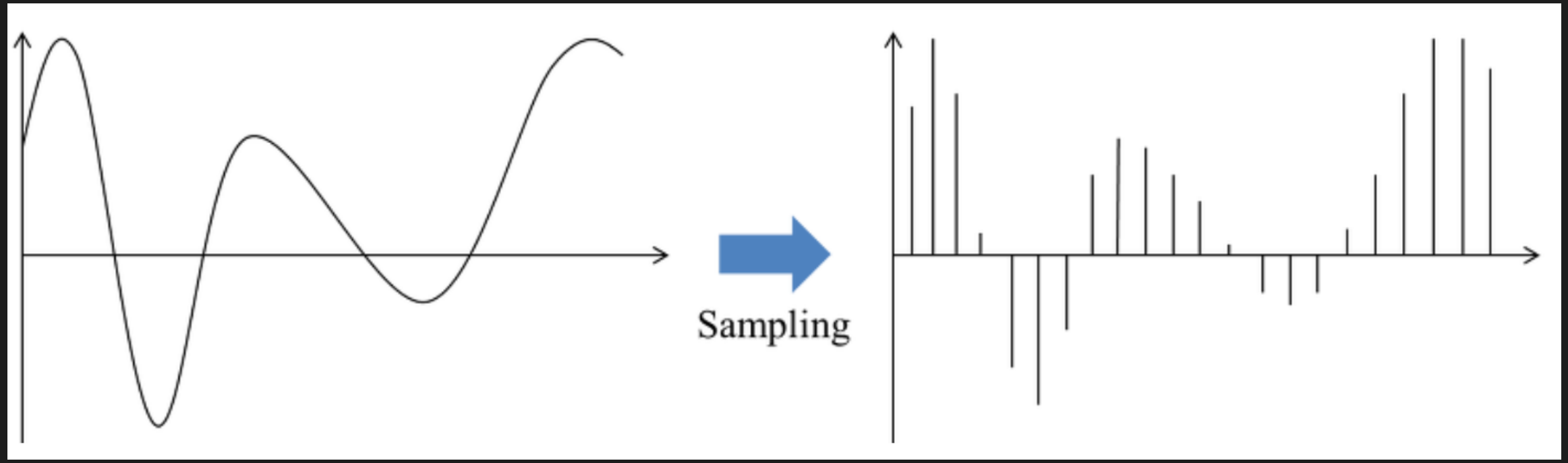
Data





Machine Learning Key Components

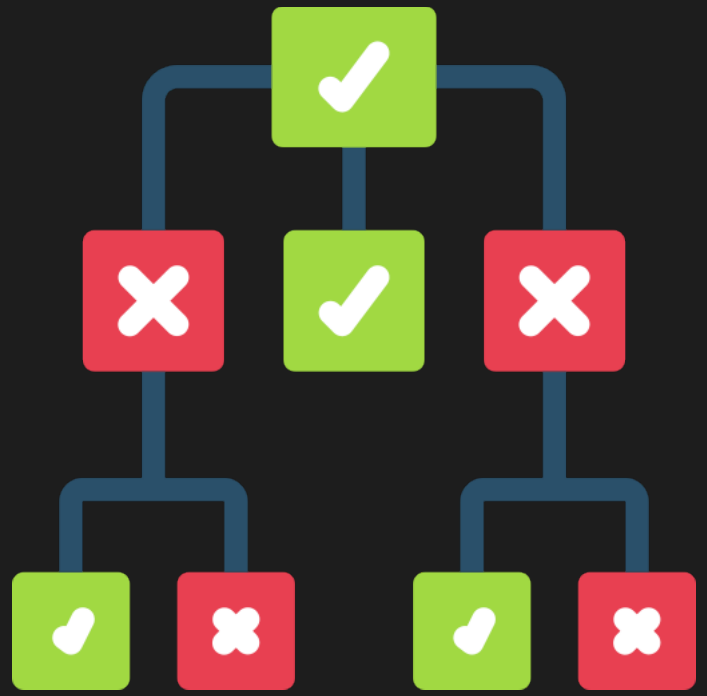
Data

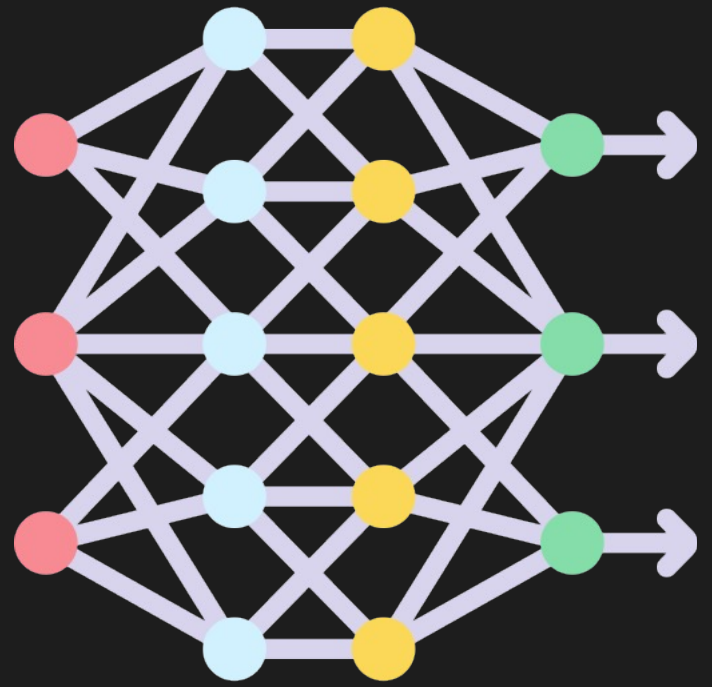


https://untitledtblog.tistory.com/120

Machine Learning Key Components

Model





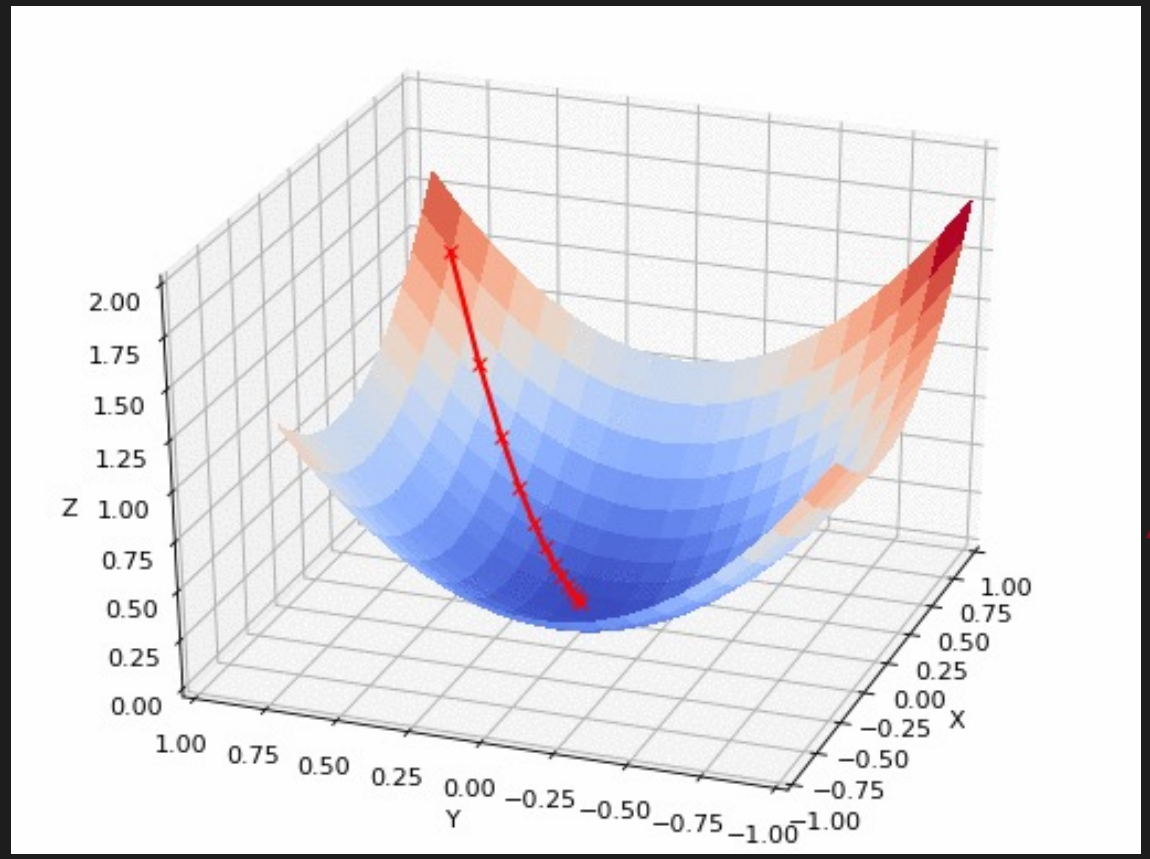
Machine Learning Key Components

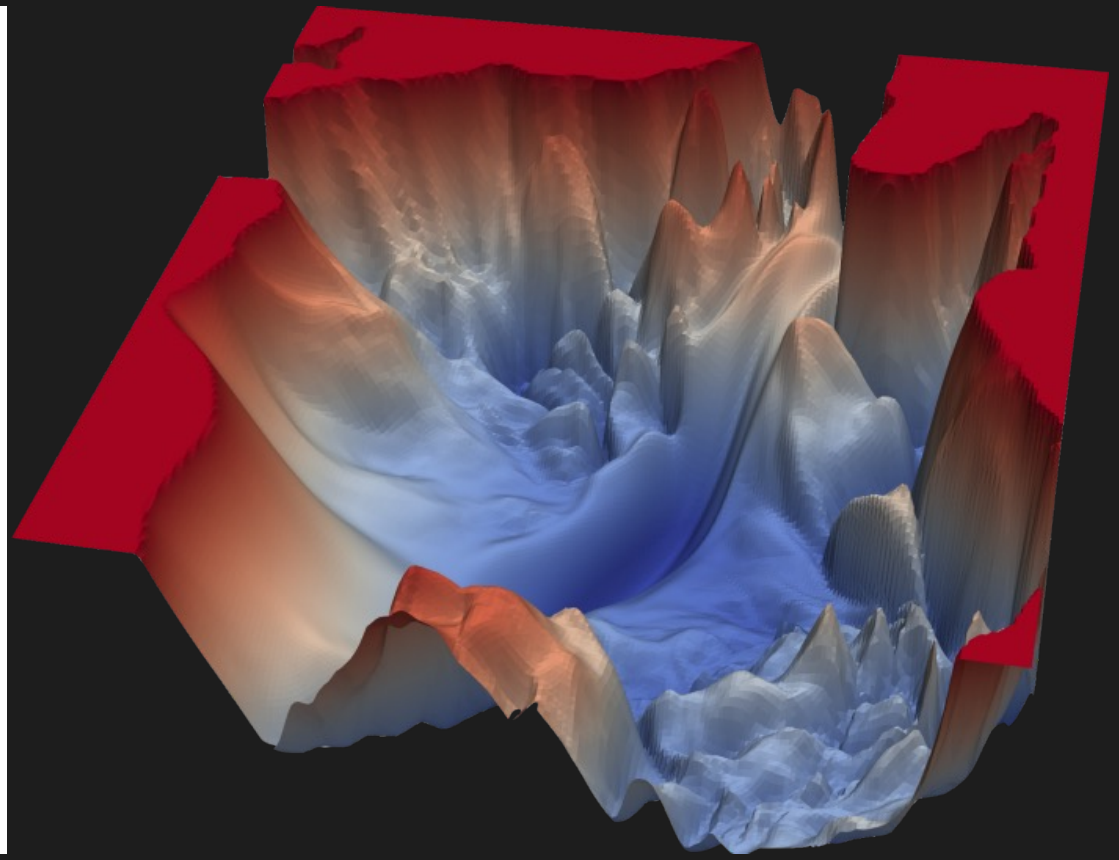
Objective

- 우리가 원하는 것 : 모델의 예측 값이 실제 값과 가까워지는 것
  - 즉, 차이 (Loss)를 최소화하고 싶다.
- Objective Function >= Cost Function >= Loss Function
  - Objective Function : 학습을 통해 최적화시키고자 하는 목표
  - Cost Function : 모든 입력 데이터에 대한 오차
  - Loss Function : 입력 데이터에 대한 예측 값과 실제 값 사이의 차이 (뒤에서 더 자세히)

Machine Learning Key Components

Optimization Algorithm





https://machine-learning.paperspace.com/wiki/gradient-descent

https://only-wanna.tistory.com/entry/%EB%AA%A9%EC%A0%81-%ED%95%A8%EC%88%98%EC%99%80-%EC%B5%9C%EC%A0%81%ED%99%94

Machine Learning Key Components

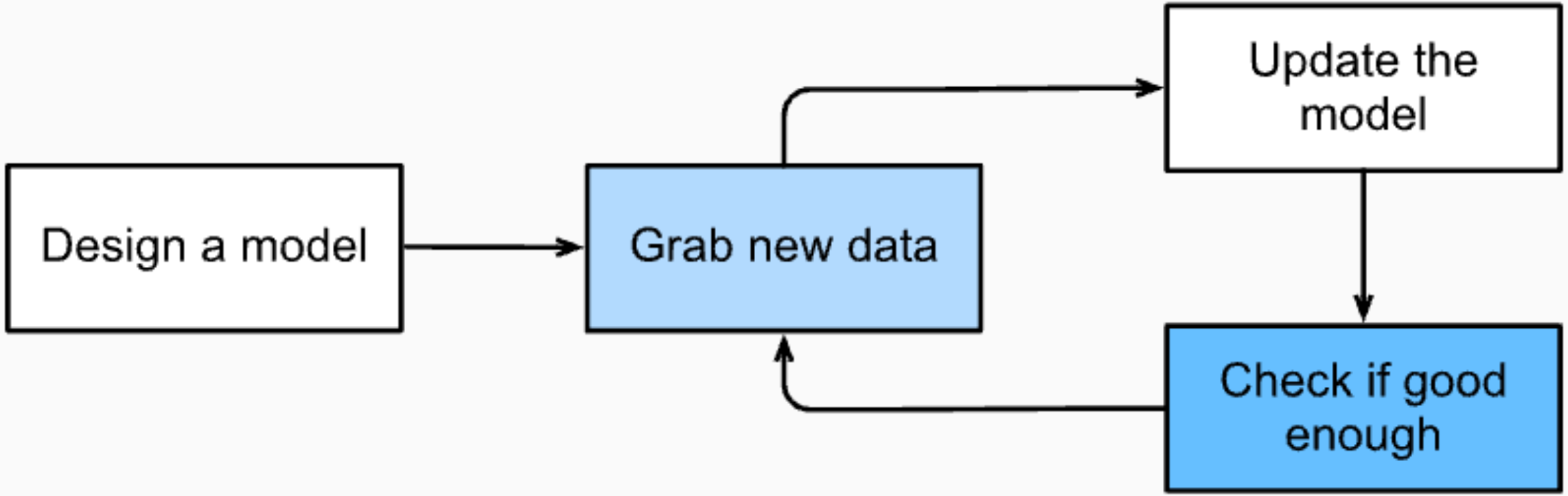


Fig. 1.1.2 A typical training process.

https://d2l.ai/chapter_introduction/index.html

What is “Good” Model?

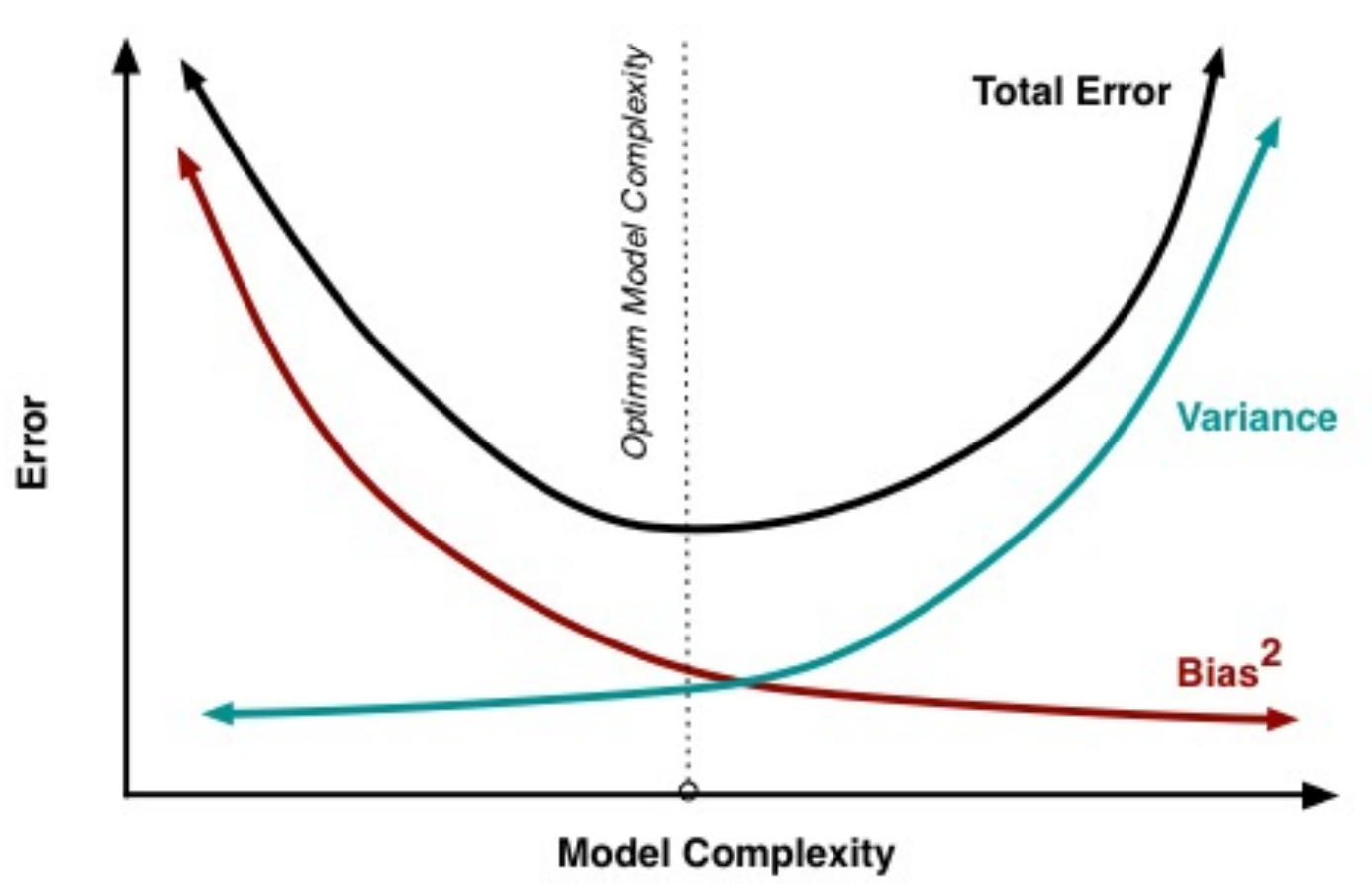
- Bias-Variance Tradeoff and Model Complexity

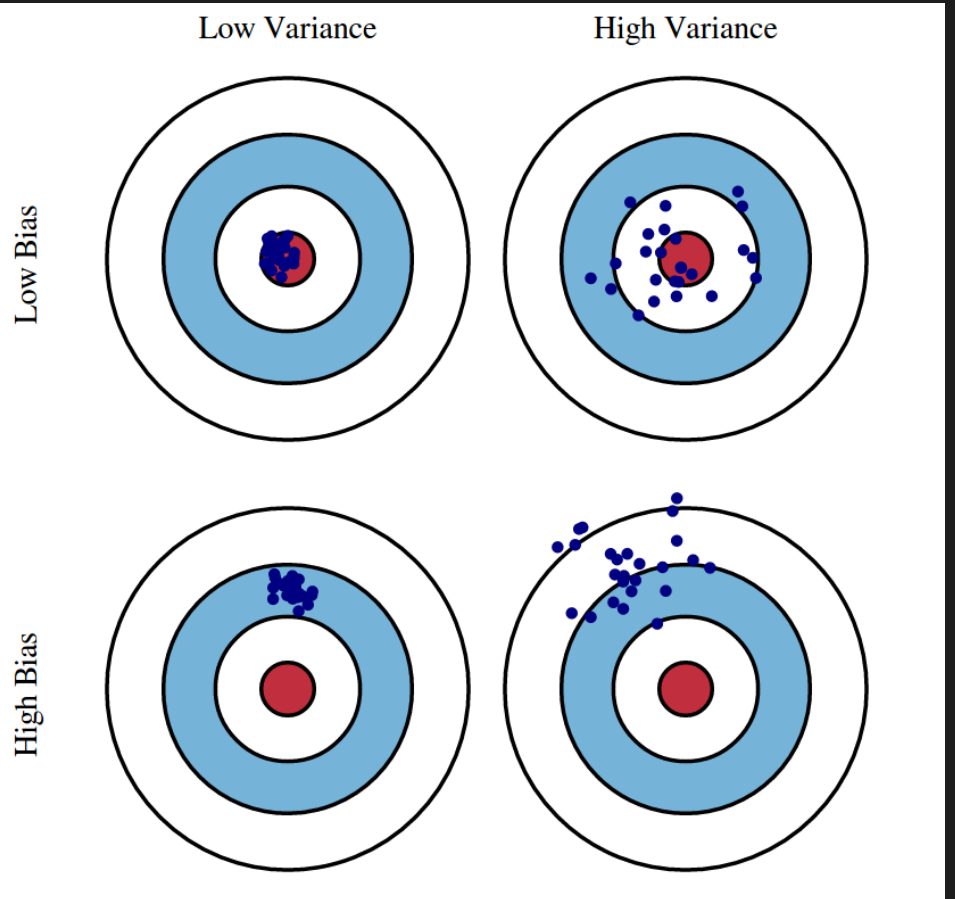
$$\underbrace{E_{\mathbf{x},y,D}\left[\left(h_D(\mathbf{x})-y\right)^2\right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x})-\bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y}\left[\left(\bar{y}(\mathbf{x})-y\right)^2\right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x})-\bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

What is “Good” Model?

- Bias-Variance Tradeoff and Model Complexity

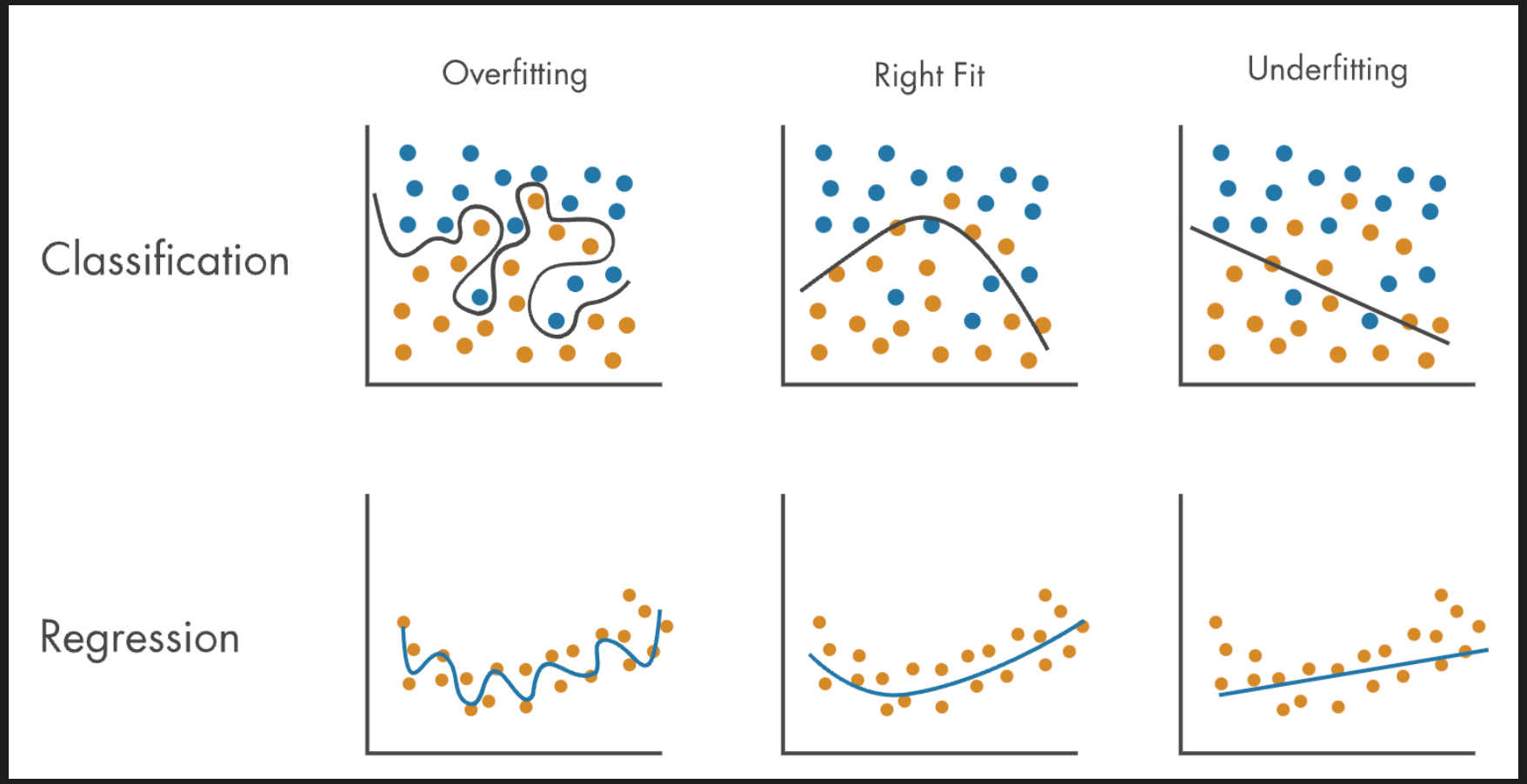




https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

What is “Good” Model?

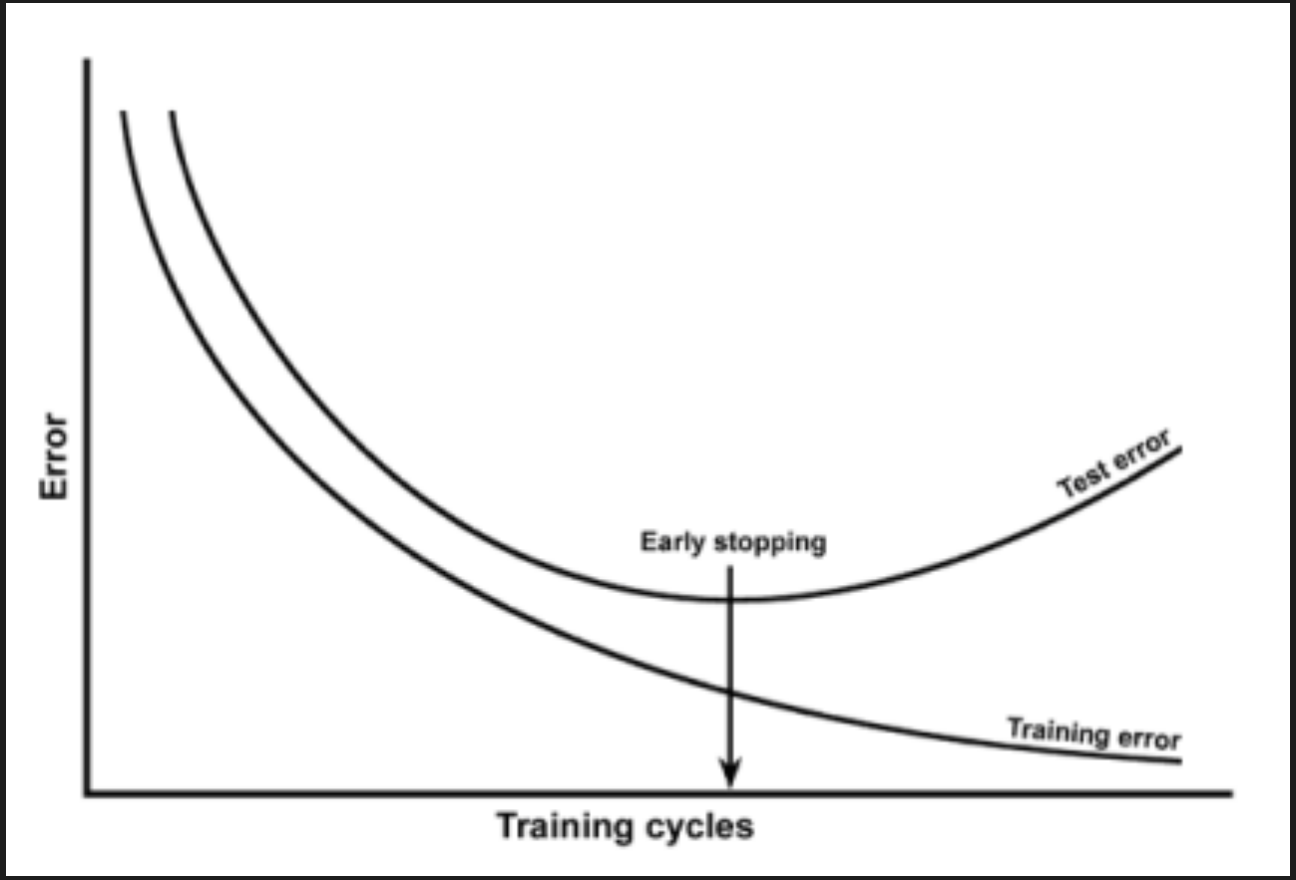
- Overfitting, Underfitting



https://kr.mathworks.com/discovery/overfitting.html

What is “Good” Model?

- Overfitting, Underfitting

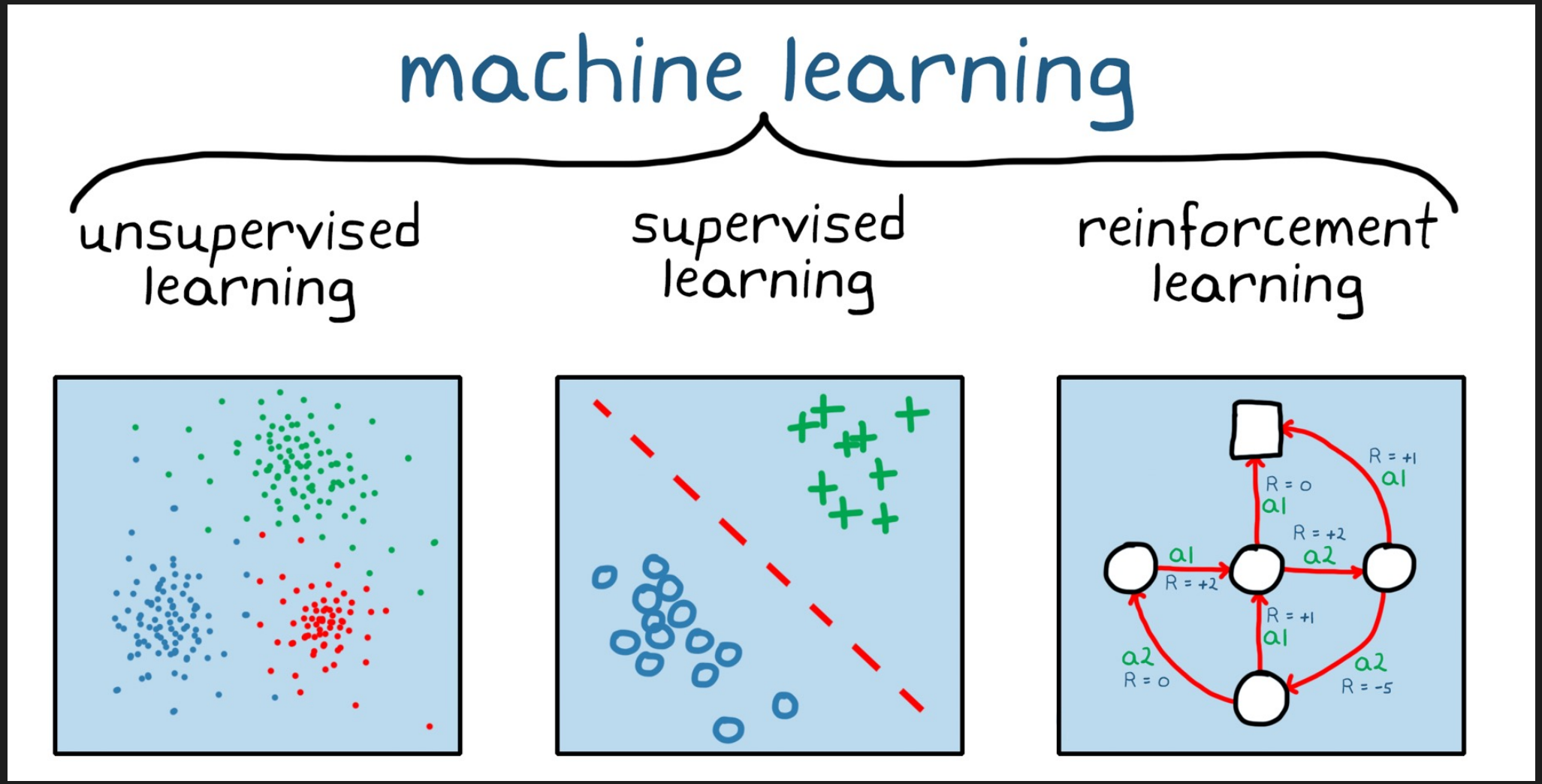


https://untitledtblog.tistory.com/68

What is “Good” Model?

Troubleshooting은 DeepIntoDeep 4회차에서…

Machine Learning Problems

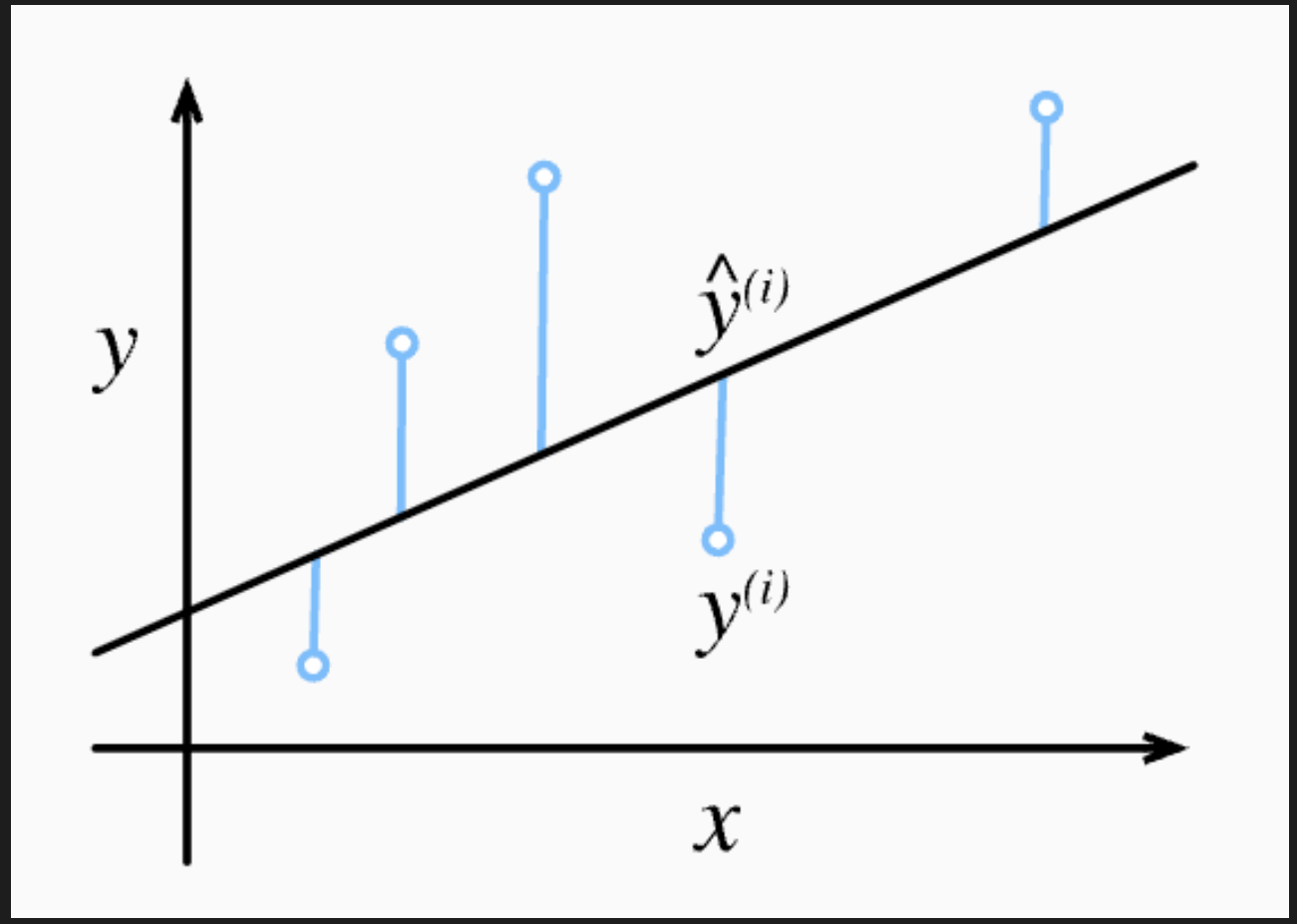


https://kr.mathworks.com/discovery/reinforcement-learning.html

Deep Learning Basics

Model

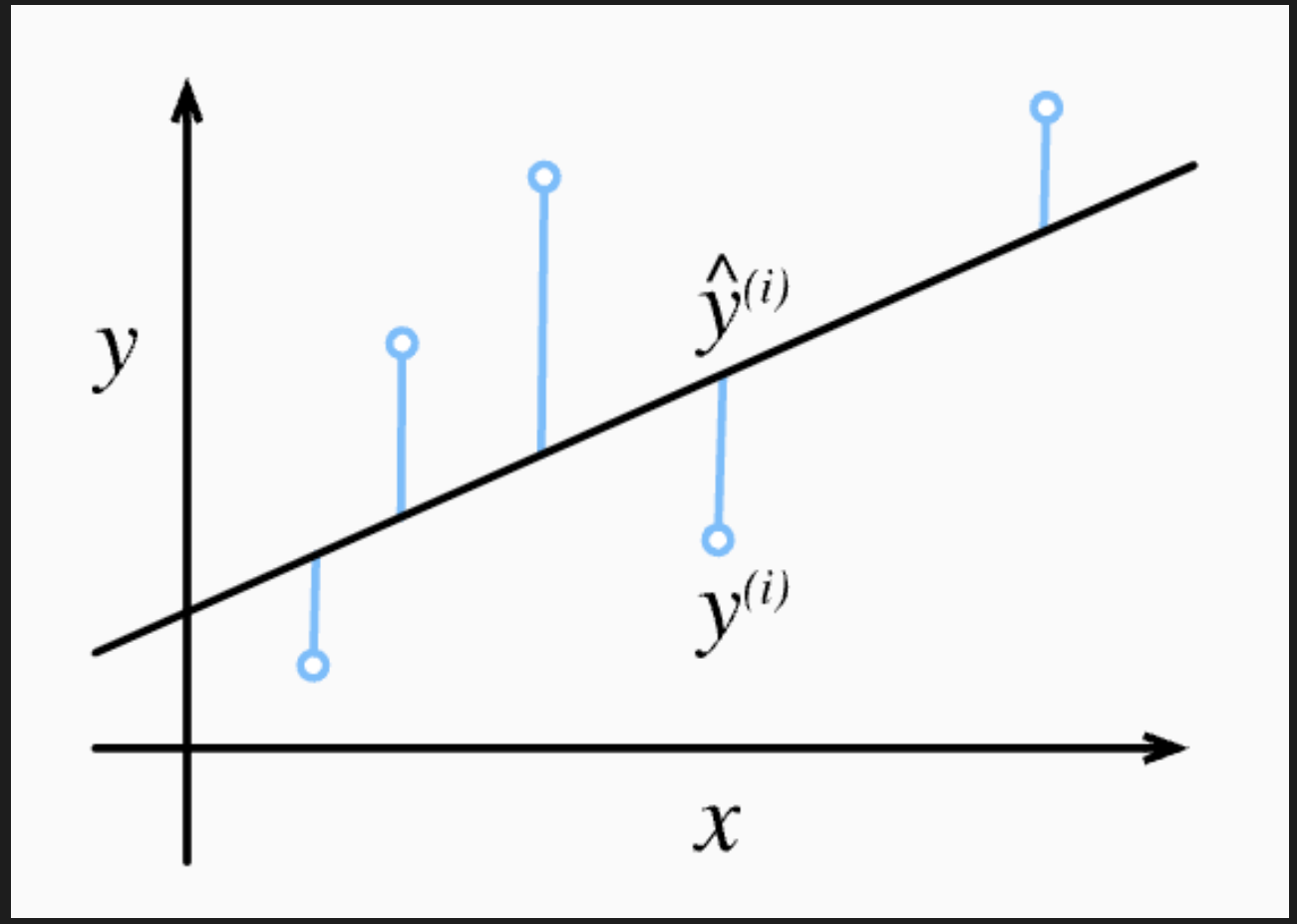
Linear Model



https://d2l.ai/chapter_linear-regression/linear-regression.html#loss-function

Model

Linear Model



$$\hat{y} = w_1 x_1 + \cdots + w_d x_d + b.$$

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b.$$

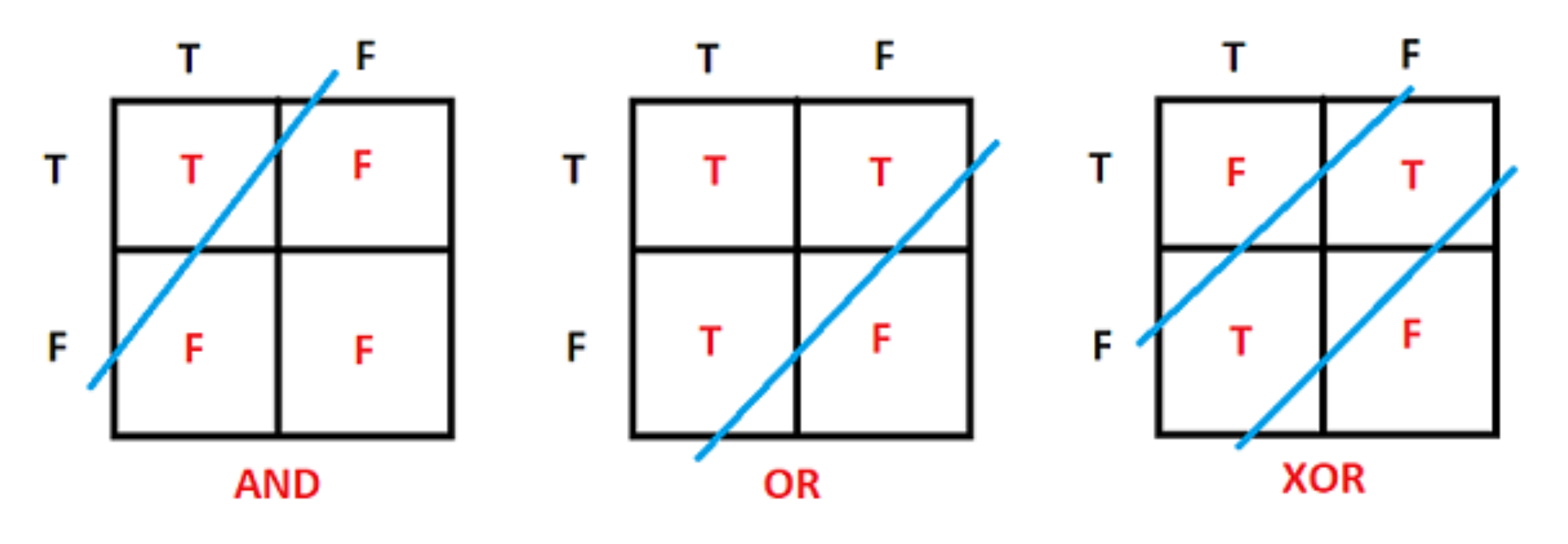
$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$$

https://d2l.ai/chapter_linear-regression/linear-regression.html#loss-function

Model

Linear Model

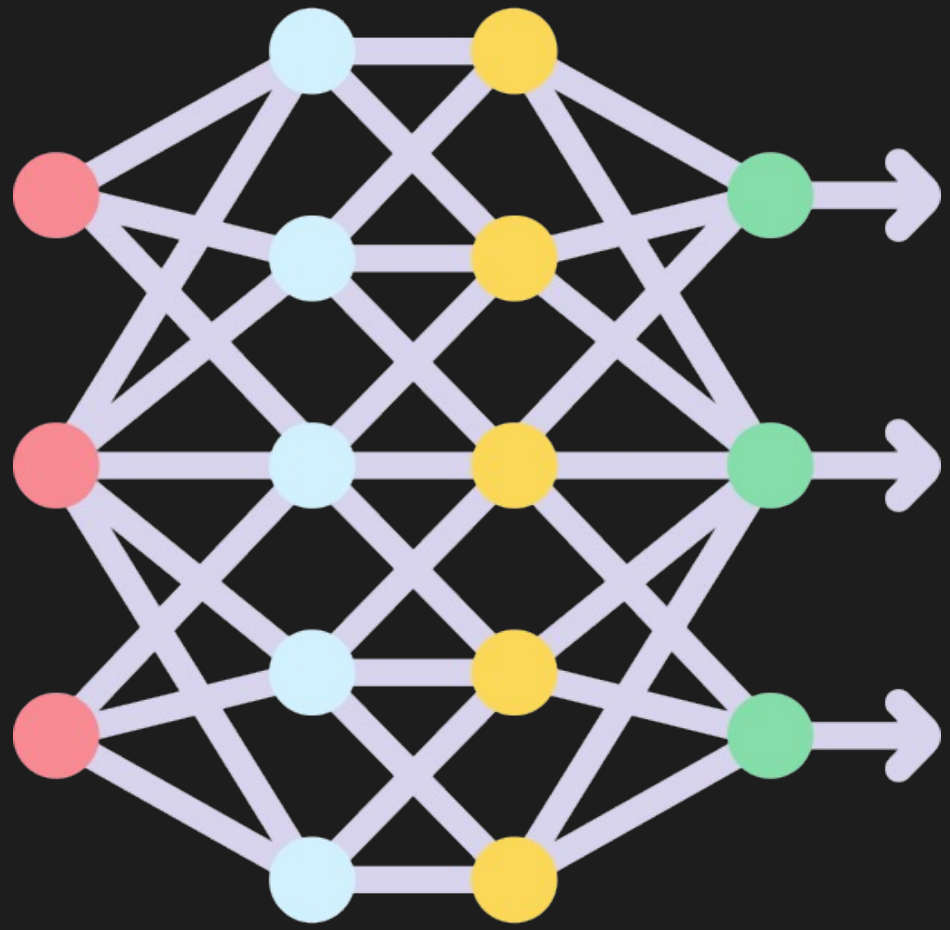
- XOR Problem



https://velog.io/@jyunxx/DL-XOR-Problem

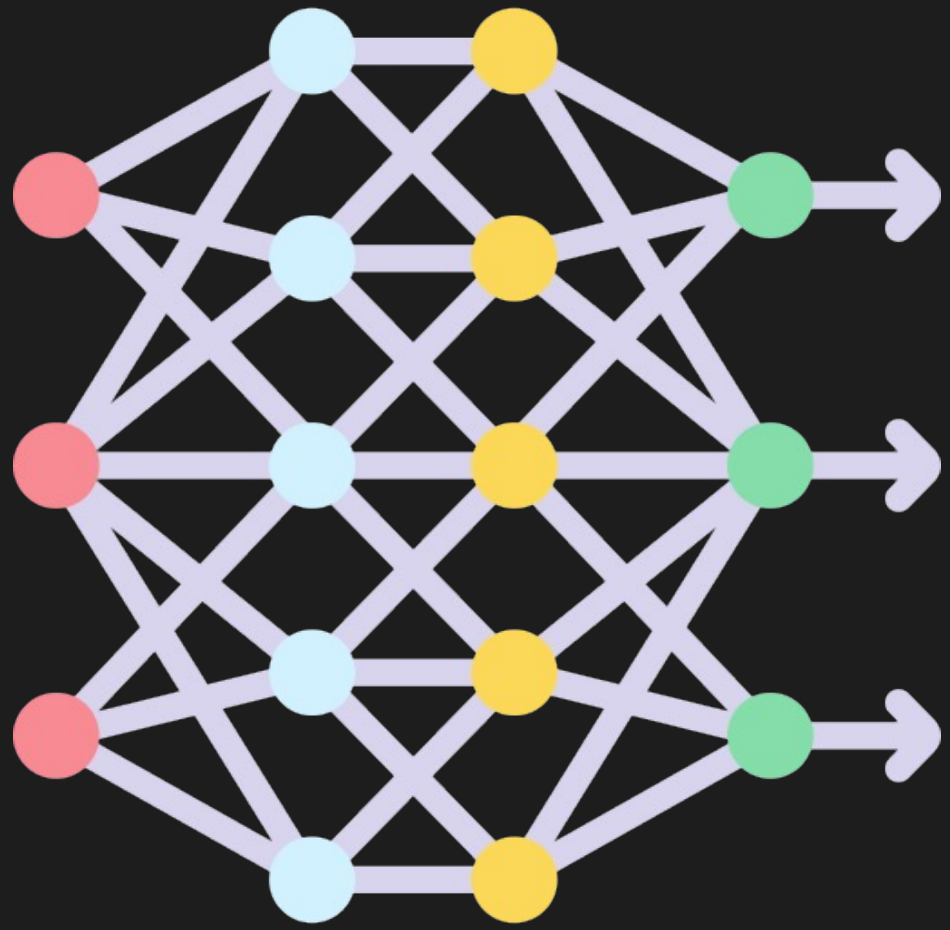
Model

Linear Model



Model

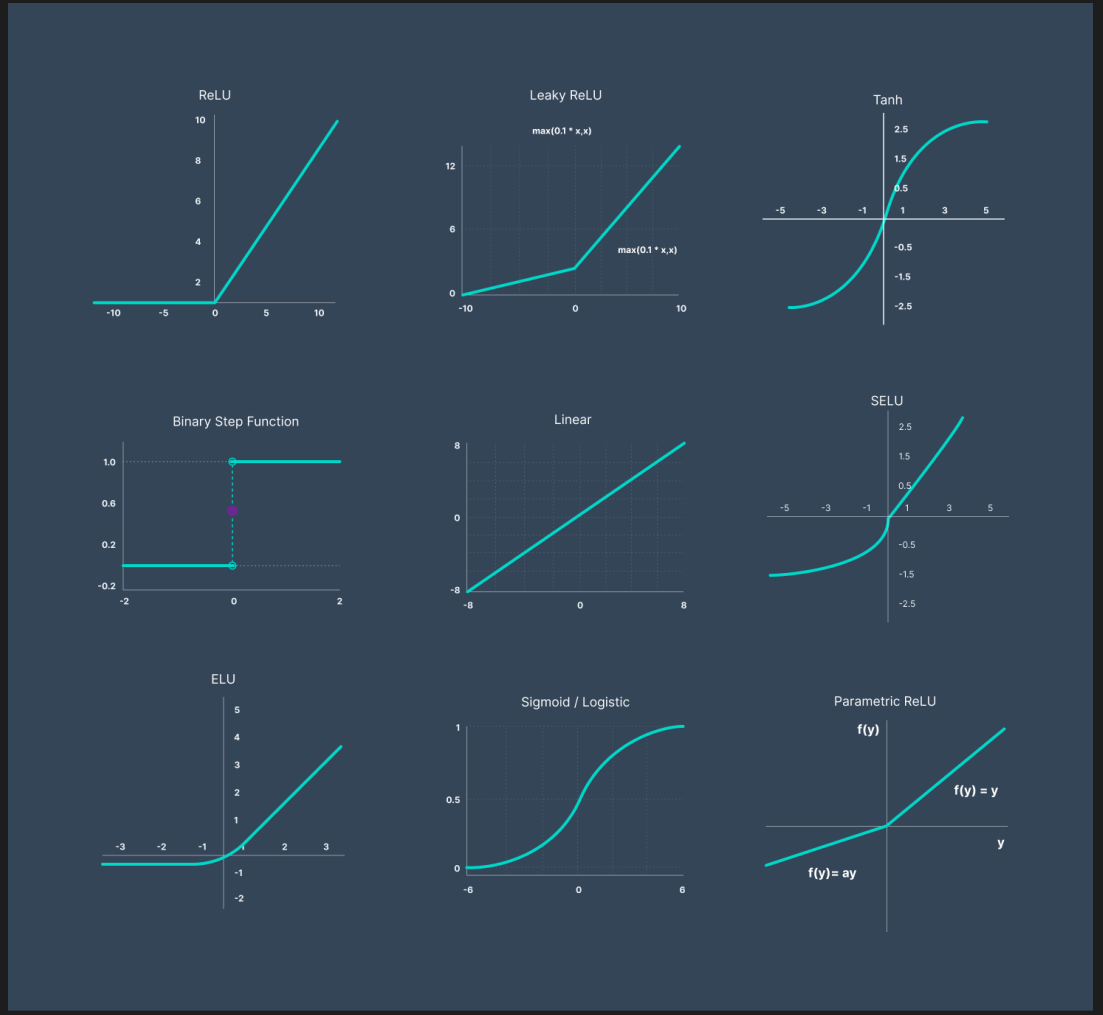
Multi Layer Perceptron



Model

Multi Layer Perceptron

- Nonlinearity
- 서로 다른 두 행렬을 곱하면 새로운 행렬이 된다.



https://www.v7labs.com/blog/neural-networks-activation-functions

Model

Multi Layer Perceptron

- MLP로 모든 함수를 근사할 수 있다. (물론 이론적으로 잘 학습시켜야 함.)

Universal approximation theorem

6 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

This article **may be too technical for most readers to understand**. Please help improve it to make it understandable to non-experts, without removing the technical details. *(July 2023)* (*Learn how and when to remove this template message*)

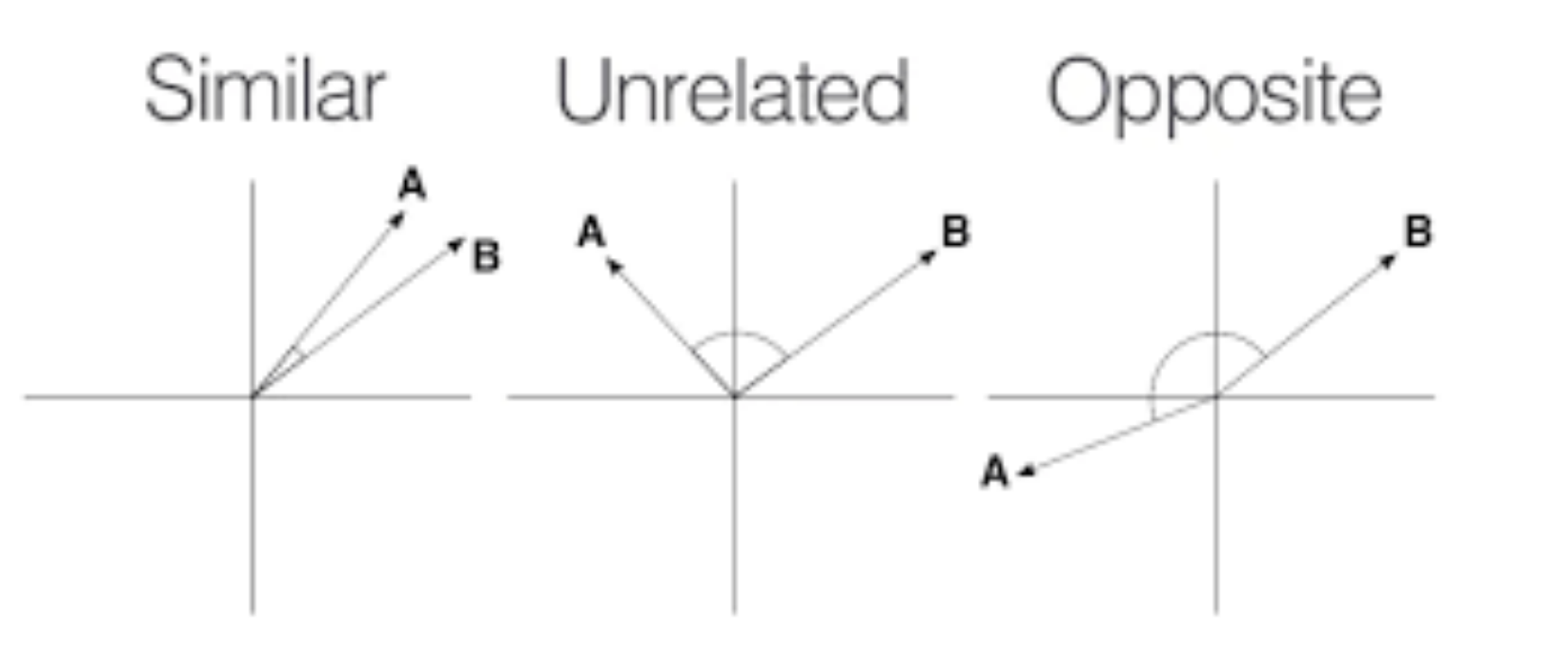
Artificial neural networks are combinations of multiple simple mathematical functions that implement more complicated functions from (typically) real-valued vectors to real-valued vectors. The spaces of multivariate functions that can be implemented by a network are determined by the structure of the network, the set of simple functions, and its multiplicative parameters. A great deal of theoretical work has gone into characterizing these function spaces.

https://en.wikipedia.org/wiki/Universal_approximation_theorem

Model

Encode and Decode

- Embedding Vectors and Features



https://medium.com/@milana.shxanukova15/cosine-distance-and-cosine-similarity-a5da0e4d9ded

Data and Loss

Regression

- MSE (Mean Squared Error)
- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)

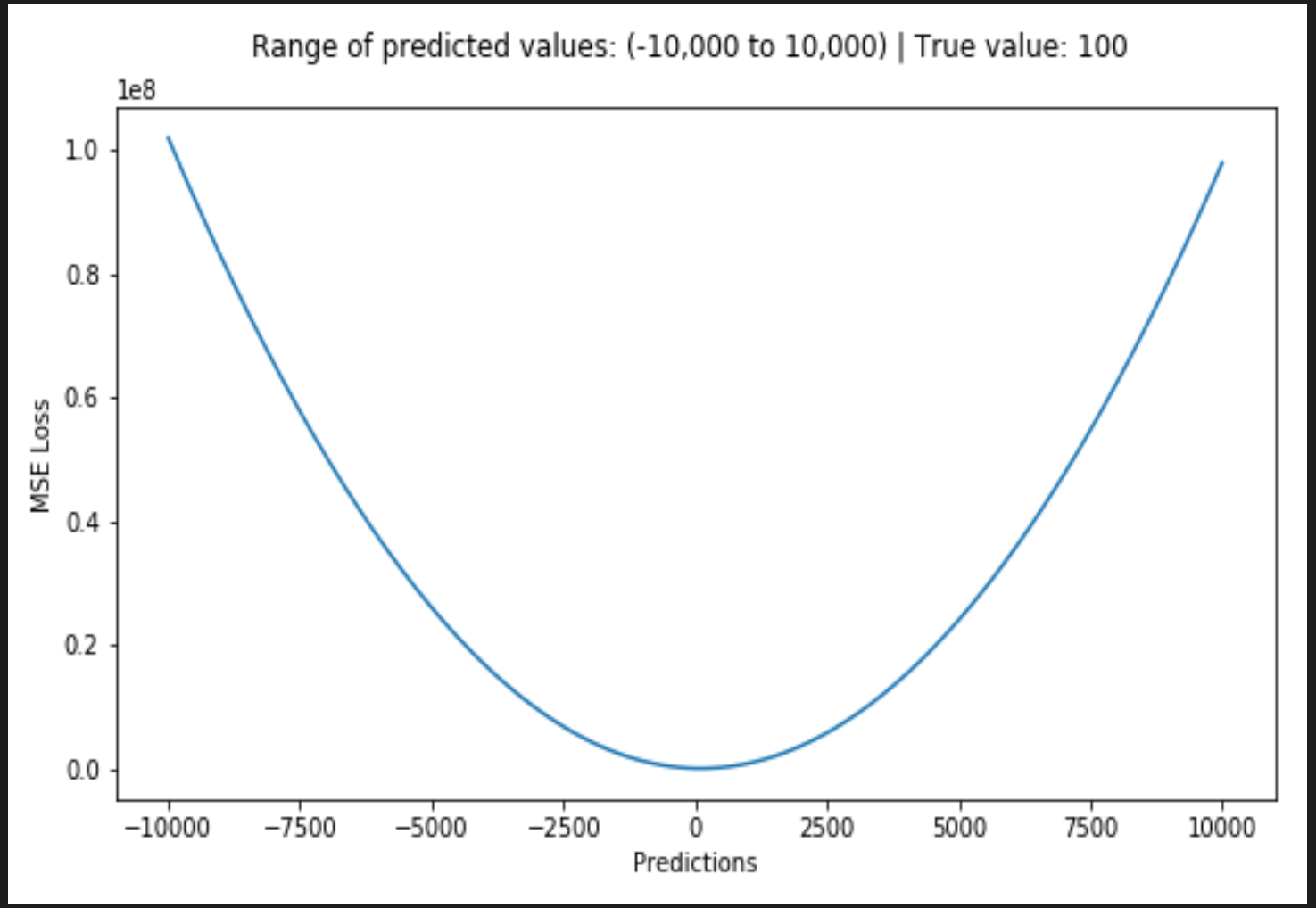
Classification

- Cross-Entropy Loss
- Binary Cross-Entropy Loss (Sigmoid + CE Loss)
- Category Cross-Entropy Loss (Sigmoid + CE Loss)

Data and Loss

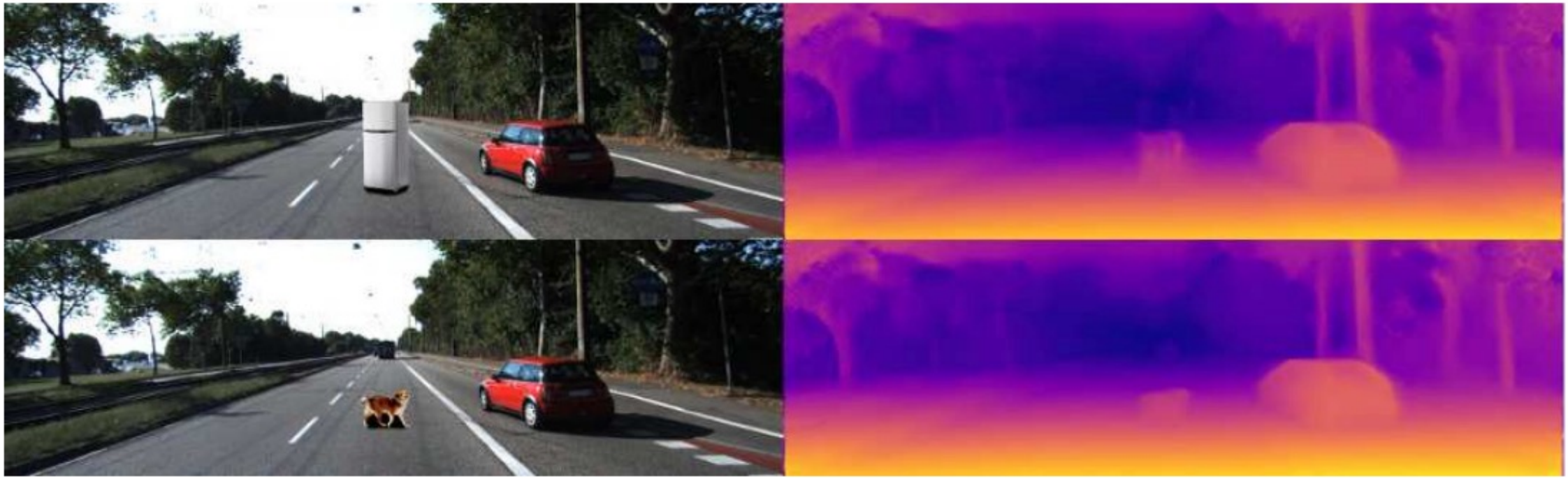
MSE Loss

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$



Data and Loss

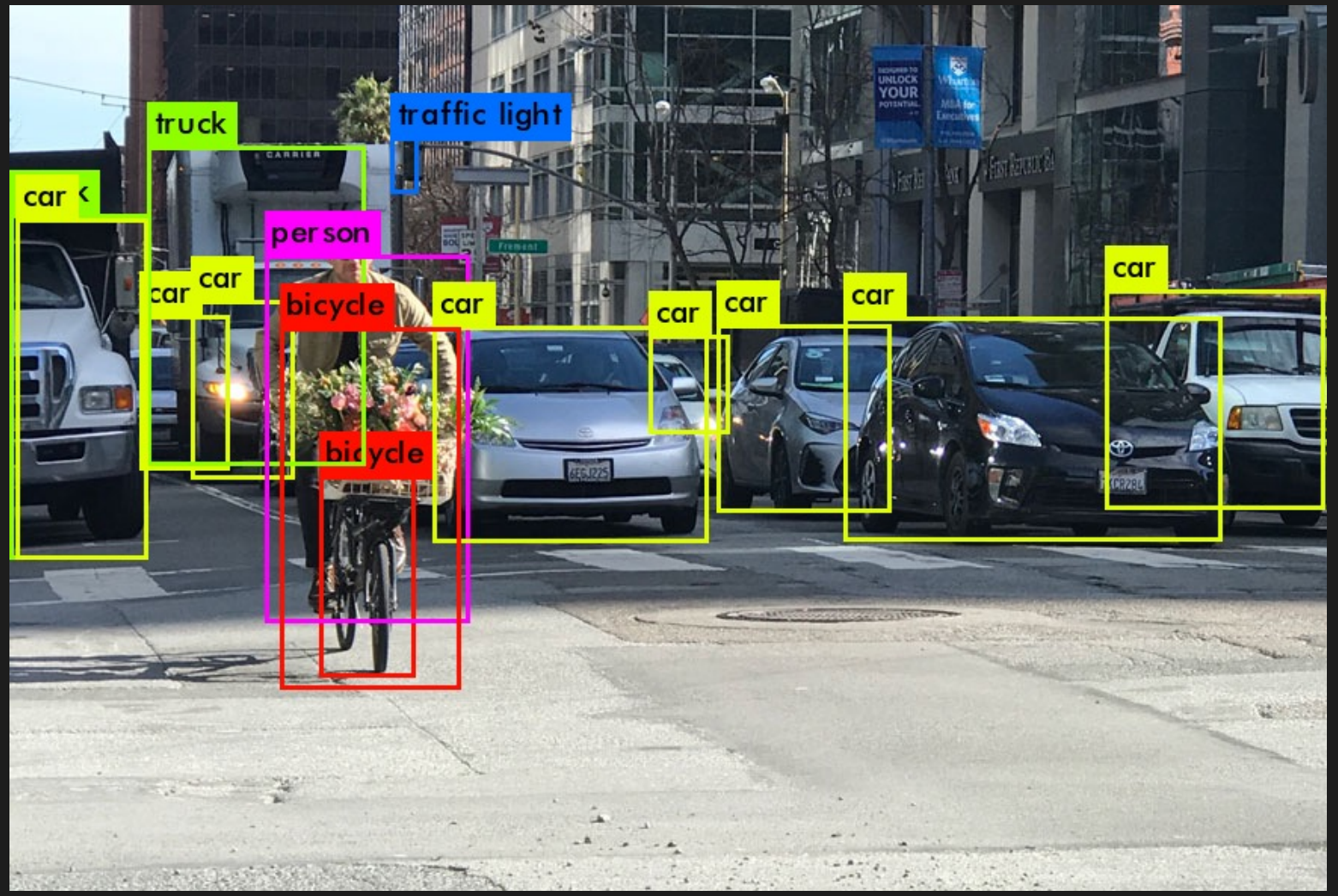
MSE Loss



https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1

Data and Loss

MSE Loss



https://itsjb13.medium.com/building-an-advanced-object-detection-application-for-autonomous-vehicles-yolov7-intel-pytorch-478ee5cedd39

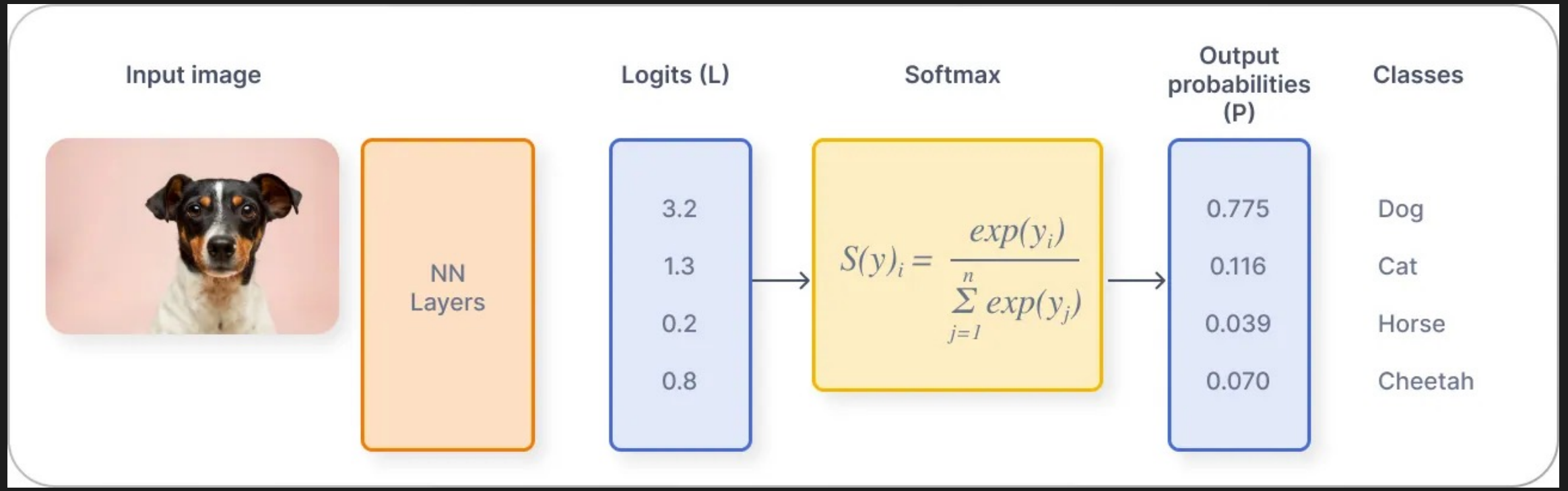
Data and Loss

MSE Loss

https://gaussian37.github.io/vision-concept-optical_flow/

Data and Loss

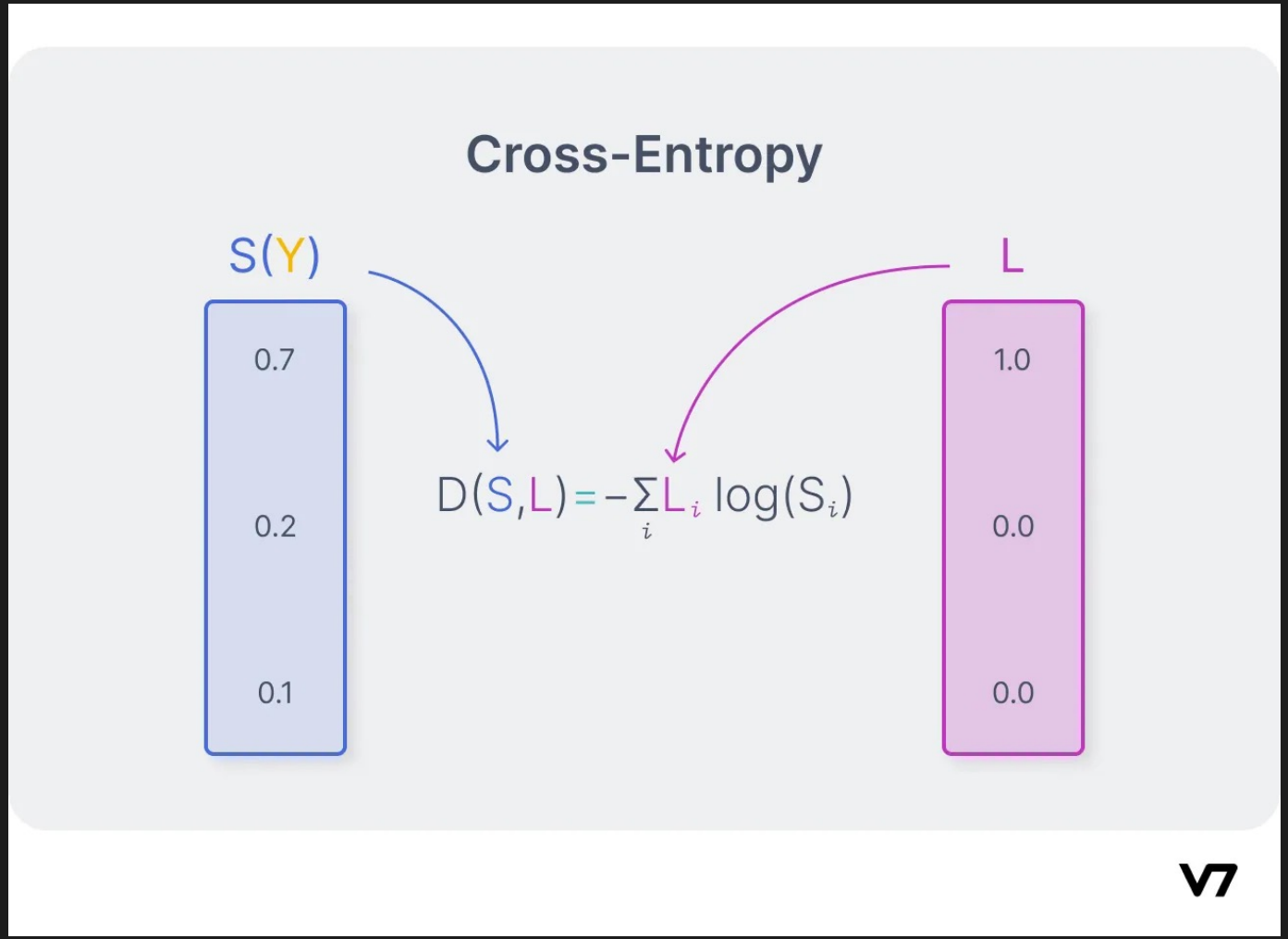
CrossEntropy Loss



https://www.v7labs.com/blog/cross-entropy-loss-guide

Data and Loss

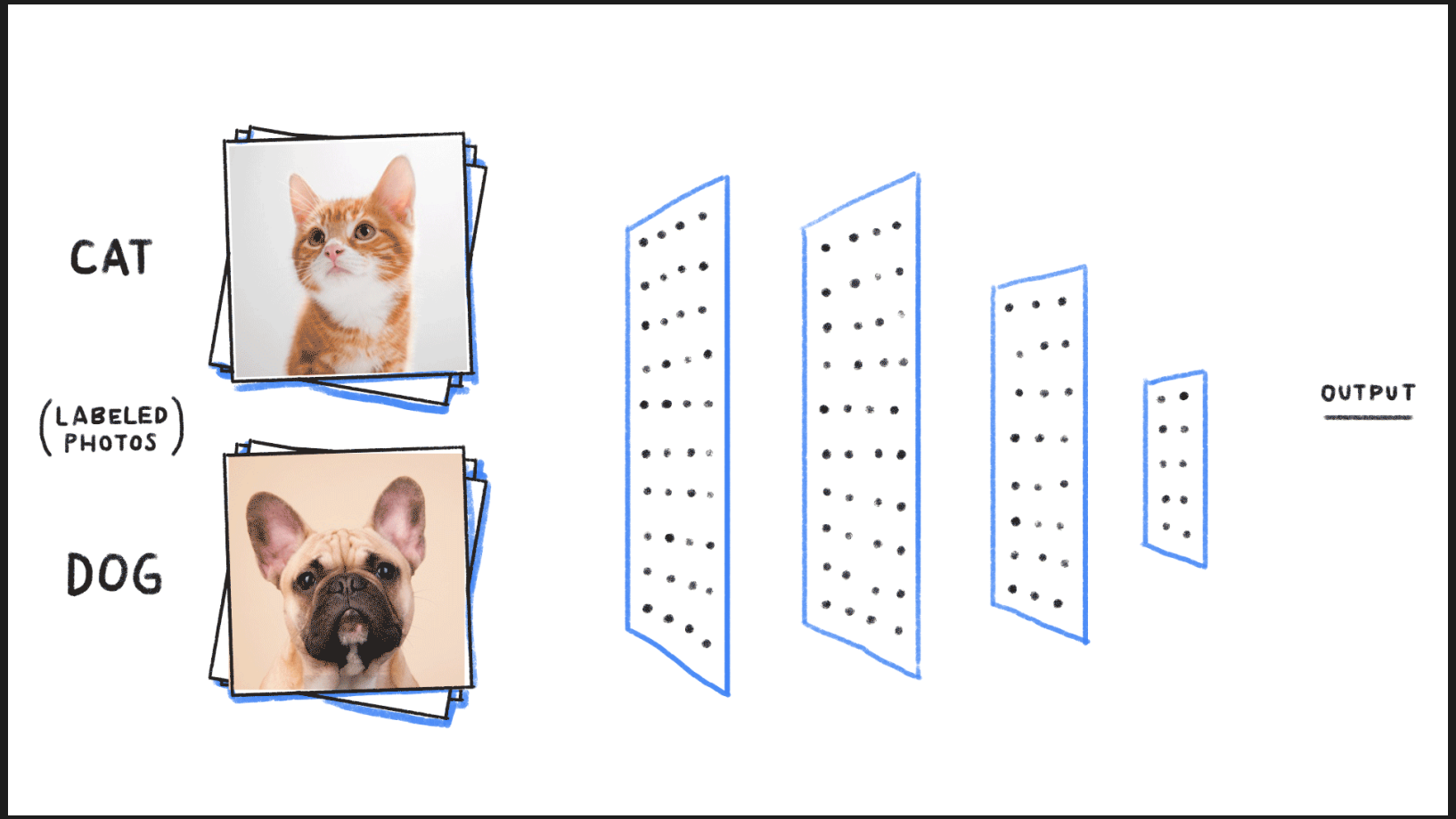
CrossEntropy Loss



https://www.v7labs.com/blog/cross-entropy-loss-guide

Data and Loss

CrossEntropy Loss



https://www.kaggle.com/code/arbazkhan971/image-classification-using-cnn-94-accuracy

Data and Loss

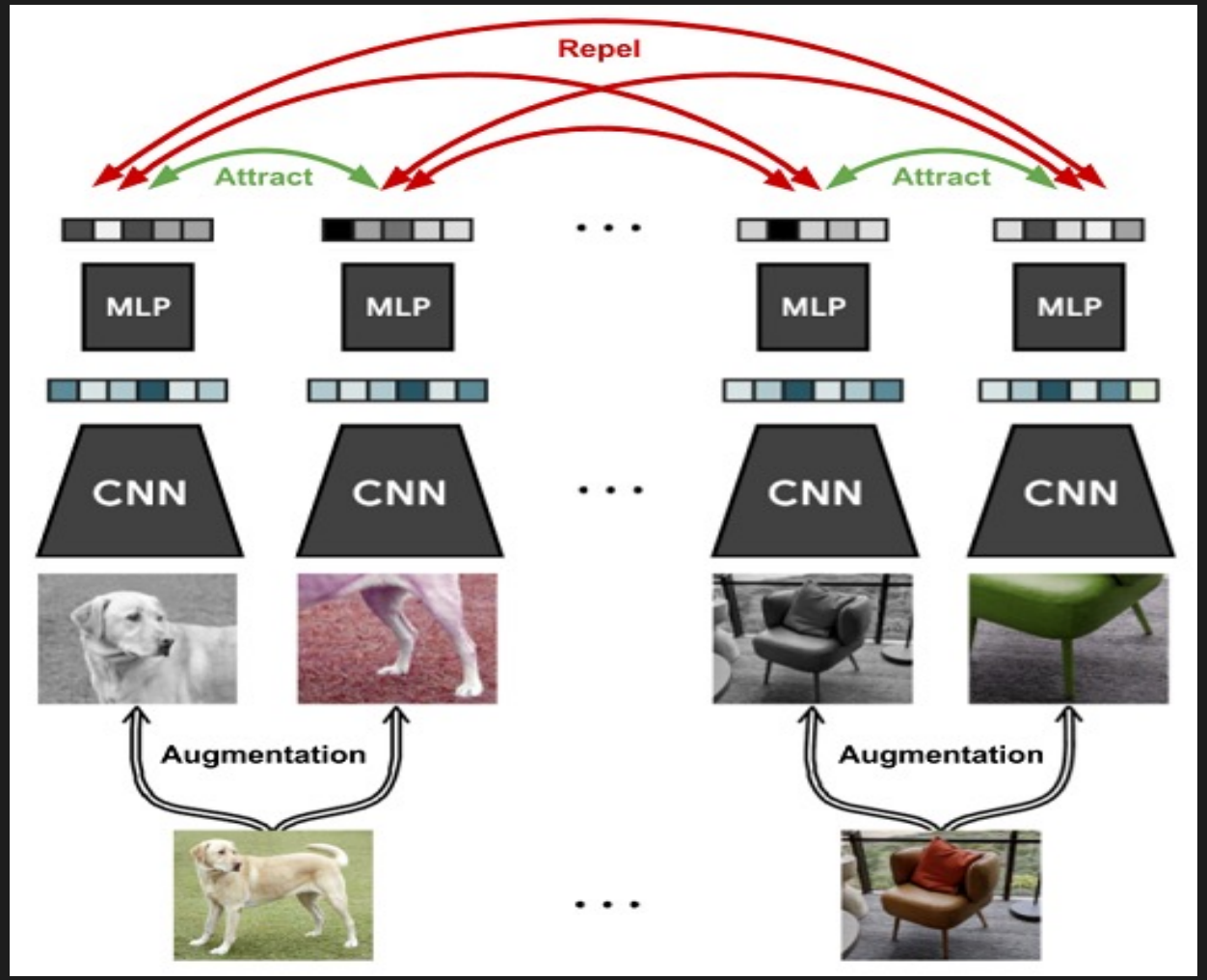
CrossEntropy Loss

https://segment-anything.com/

Data and Loss

CosineEmbedding Loss

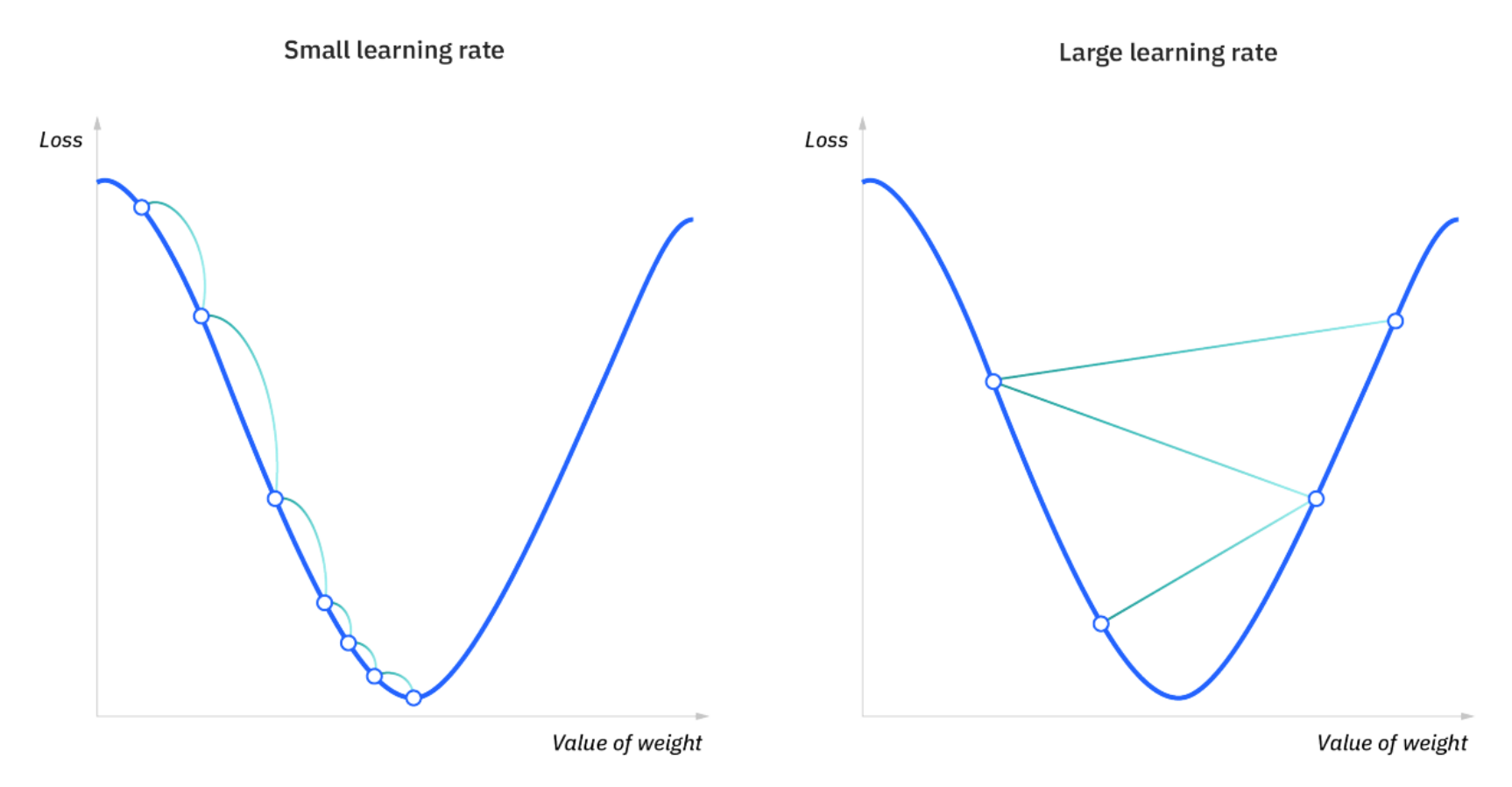
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



https://sh-tsang.medium.com/review-simclr-a-simple-framework-for-contrastive-learning-of-visual-representations-5de42ba0bc66

Learning Algorithm

Gradient Descent



https://www.ibm.com/topics/gradient-descent

Learning Algorithm

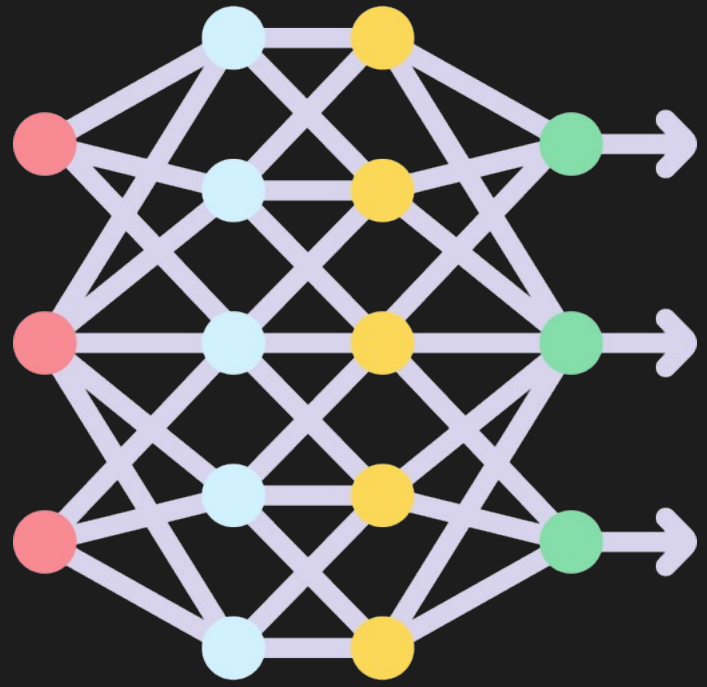
Backpropagation and chain-rule

```python
import torch

a = torch.tensor([1, 2, 3], requires_grad=True, dtype=torch.float32)
b = torch.tensor([4, 5, 6], requires_grad=True, dtype=torch.float32)

c = torch.matmul(a, b)

c.backward()

print(a.grad) # tensor([4., 5., 6.])
print(b.grad) # tensor([1., 2., 3.])
```

Learning Algorithm

Backpropagation and chain-rule

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



Learning Algorithm

Backpropagation and chain-rule

z1 = x + a

z2 = z1 * b

z3 = z2 + c

dz3/dz2 = 1

dz2/dz1 = b

dz1/dx = 1

dz1/dx = b

```python
import torch

x = torch.tensor([1], requires_grad=True, dtype=torch.float)

a = torch.tensor([3], requires_grad=True, dtype=torch.float)
b = torch.tensor([2], requires_grad=True, dtype=torch.float)
c = torch.tensor([3], requires_grad=True, dtype=torch.float)

z = x + a
z = z * b
z = z + c
z.backward()

print(x.grad) # tensor([2.])
```

Learning Algorithm

Define and Run, Define by Run

- PyTorch – Define by Run
  - Forward 연산과 함께 그래프가 생성된다.
  - 유연하다.
- Tensorflow, JAX – Define and Run
  - 컴파일을 통해 필요한 연산을 정의한 이후에 연산을 한다.
  - 빠르다.

Discussion

[개인 질문] Neural Network는 깊으면 깊을수록 좋을까?

[팀 질문] 울퉁불퉁한 landscape를 가진 loss function을 학 습할 때, local minimum를 벗어나려면 어떻게 하면 좋을까?

감사합니다.