신입생 Deep Learning 기초 교육

1회: Deep Learning? DNN?

Multimodal Language Cognition Lab,
Kyungpook National University

2023.02.01

MLCL Kyungpook National University

Notice

- All students have to give a presentation with their materials
  - Presenting in 5-10 minutes
  - If no homework, you will get a disadvantage!
- You can choose Korean or English to make your presentation materials
- Summarize the two presentations you created each week for a lab meeting
- Show us what you learn as well as the results

일정표

일정	수업	장소	시간
2/1 (수)	Deep learning, DNN	테크노빌딩 508호	13:00
2/3 (금)	CNN		
2/6 (월)	Object Detection	테크노빌딩 211호	
2/8 (수)	RNN		
2/10 (금)	Seq2seq, Seq2seq with attention		
2/13 (월)	Transformer		
2/14 (화)	ASR		
2/15 (수)	ASR		
2/16 (목)	BERT		
2/17 (금)	GPT		

Multimodal Language Cognition Lab (MLCL)

정호영 (지도 교수님)

김준우 (박사과정)

- 음성 처리

도주성 (석박통합)

- 생체 신호 처리

윤은지 (석사과정)

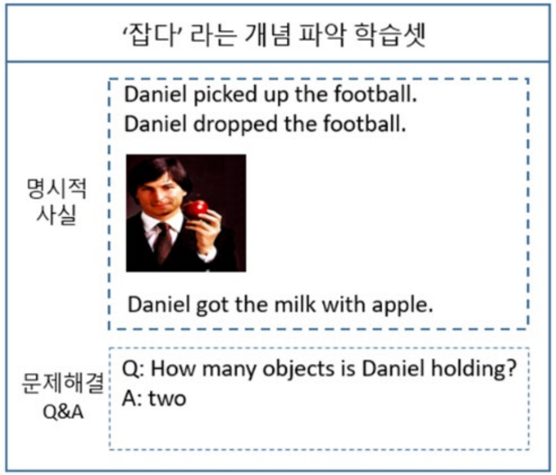
- 비디오 처리

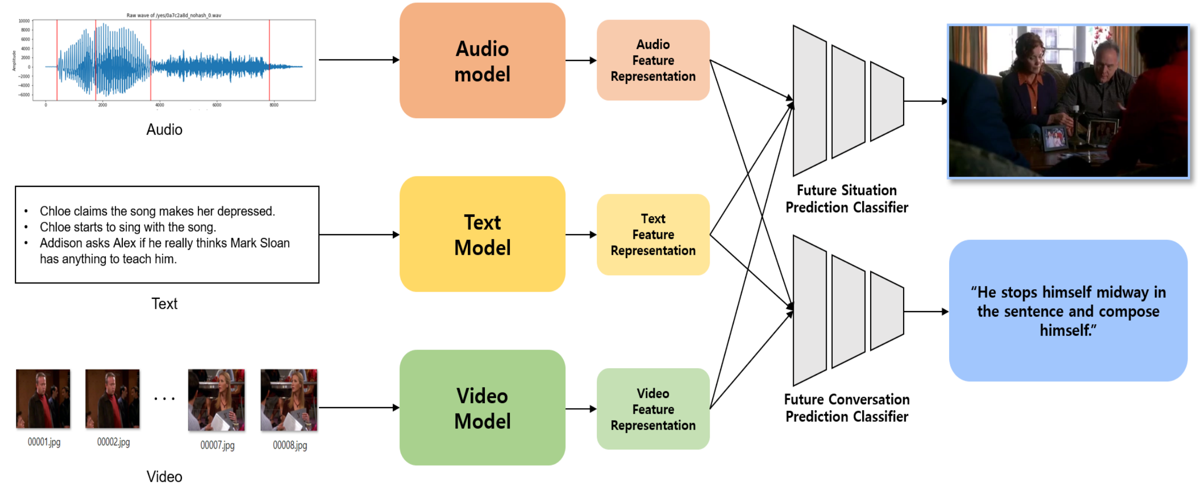
김수영 (석사과정)

- 자연어 처리

Multimodal Language Cognition Lab (MLCL)

- **Multimodal Language Processing**
  - 언어 인지 컴퓨팅 연구
  - 멀티모달 기억 모델 기반 개념 기저학습 연구
  - 멀티모달 이해 기반 미래 예측 기술 연구
- **Self-Determining Autonomous AI**
  - 자율성장형 인공지능 연구
- **Spoken Language Processing**
  - 자유발화 음성인식 연구
  - 음성데이터 증강 기술 연구

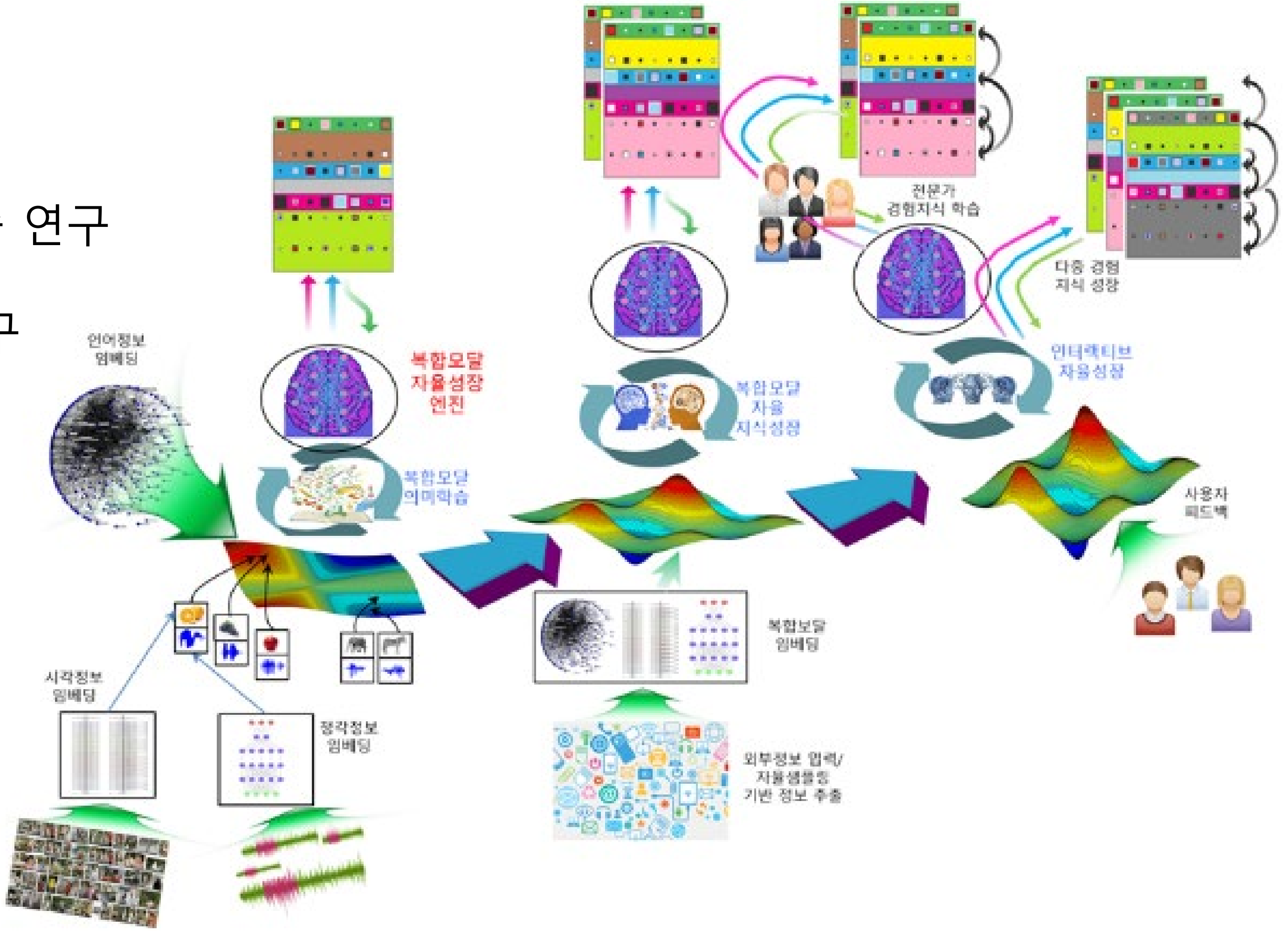




MLCL Kyungpook National University

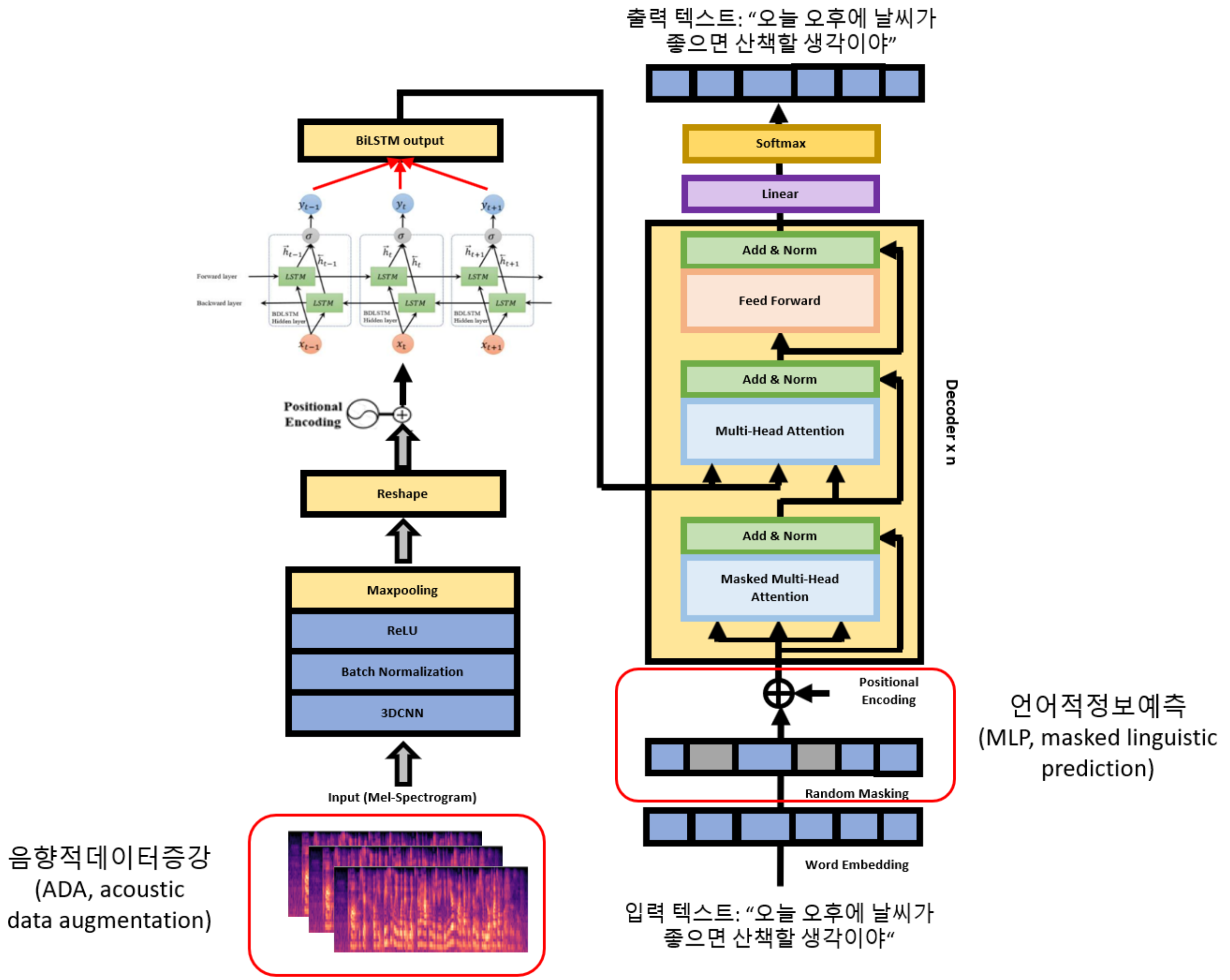
Multimodal Language Cognition Lab (MLCL)

- **Multimodal Language Processing**
  - 언어 인지 컴퓨팅 연구
  - 멀티모달 기억 모델 기반 개념 기저학습 연구
  - 멀티모달 이해 기반 미래 예측 기술 연구
- **Self-Determining Autonomous AI**
  - 자율성장형 인공지능 연구
- **Spoken Language Processing**
  - 자유발화 음성인식 연구
  - 음성데이터 증강 기술 연구



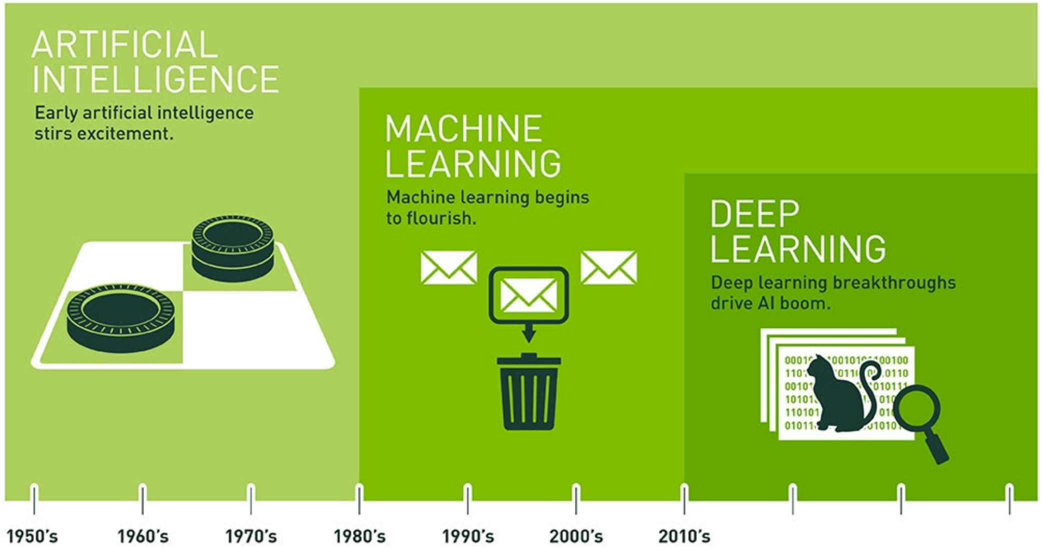
Multimodal Language Cognition Lab (MLCL)

- **Multimodal Language Processing**
  - 언어 인지 컴퓨팅 연구
  - 멀티모달 기억 모델 기반 개념 기저학습 연구
  - 멀티모달 이해 기반 미래 예측 기술 연구
- **Self-Determining Autonomous AI**
  - 자율성장형 인공지능 연구
- **Spoken Language Processing**
  - 자유발화 음성인식 연구
  - 음성데이터 증강 기술 연구

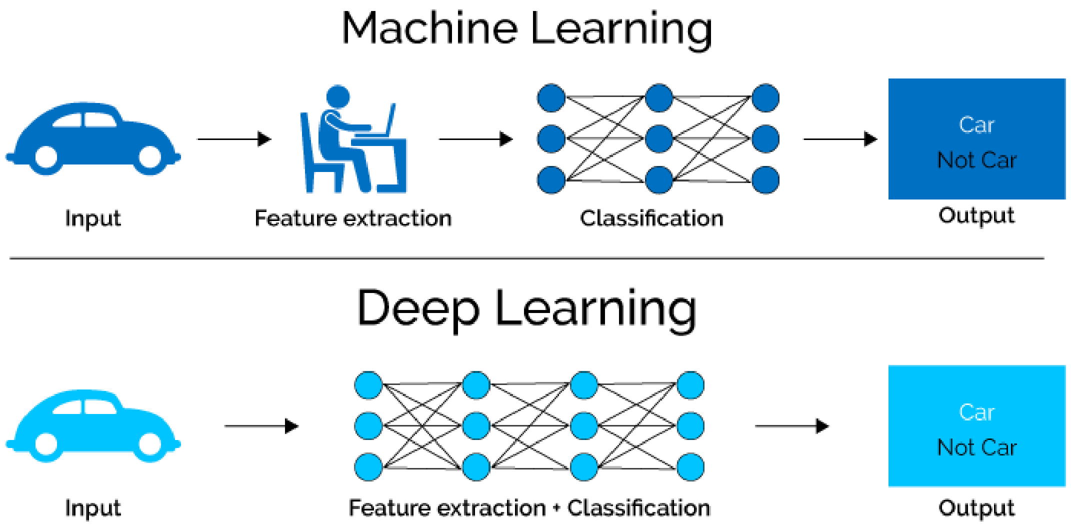


What is AI? Machine learning? Deep learning?

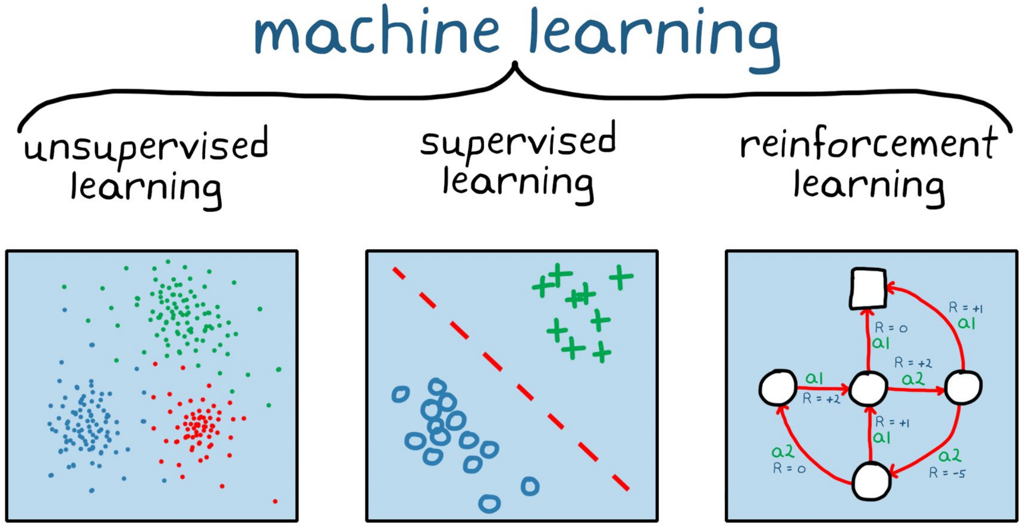
- **AI?** Human learning ability, reasoning ability, perception ability, and other artificially implemented computer programs or computer systems including them
- **Machine Learning?** Computer algorithms that automatically improve through experience
- **Deep learning?** As a method of machine learning, the model performs automatically from feature extraction to task execution

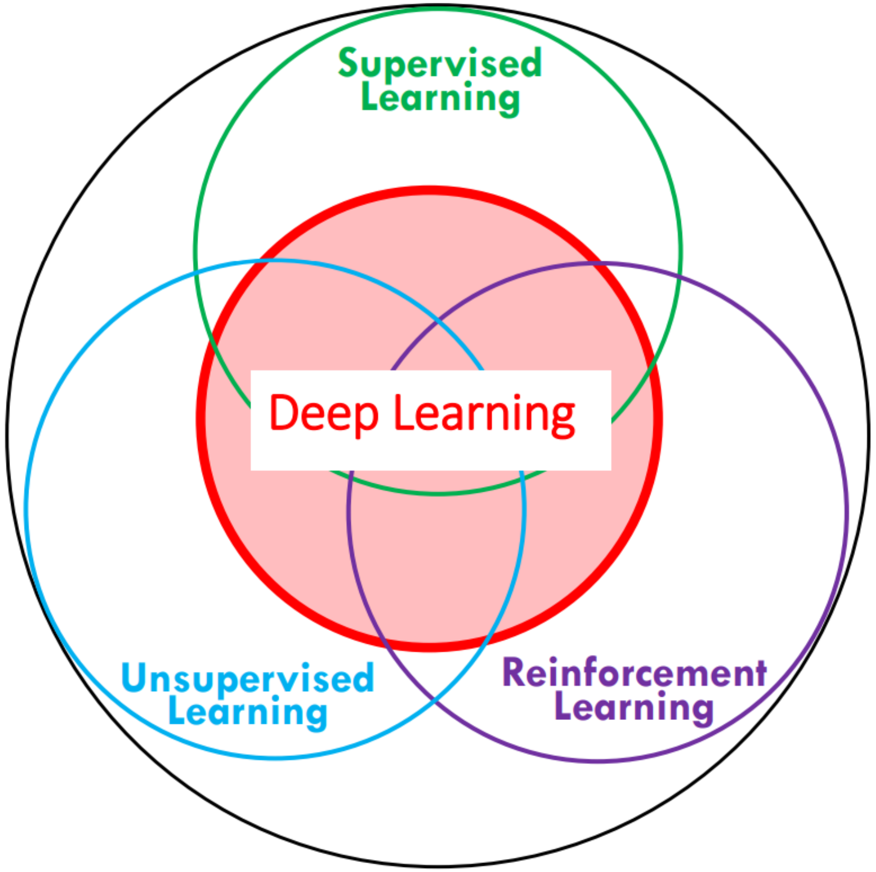


Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.



What is Deep learning?



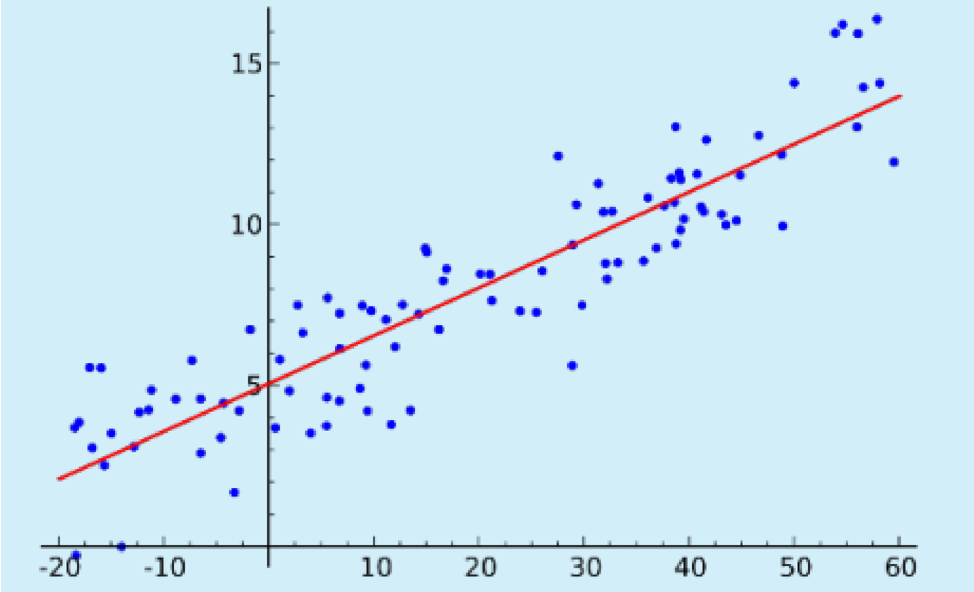


Supervised Learning

- Data: (x, y)
  - Where x is data, y is label

- Goal: Learn a function that maps $f: x \rightarrow y$

- Examples:
  - Regression
  - Classification
  - Object detection

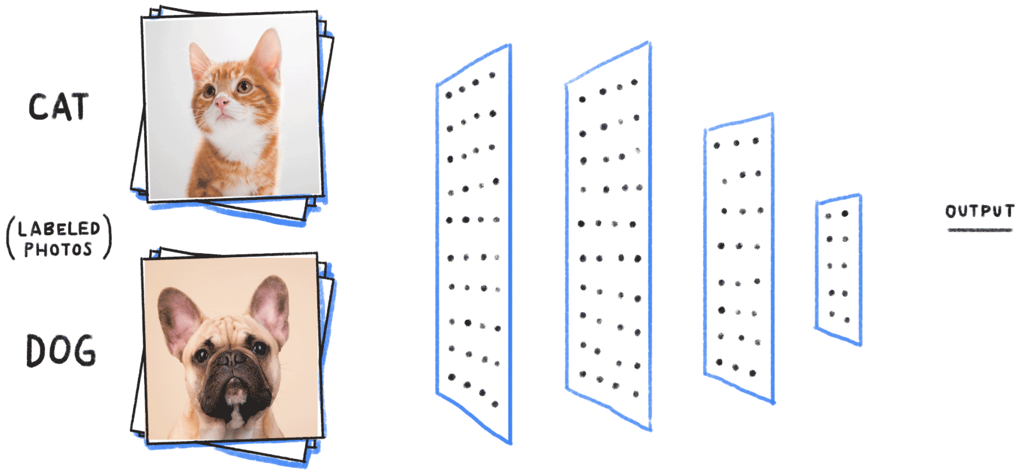


Supervised Learning

- Data: x
  - Just data

- Goal: Learn a function that maps $f: x \rightarrow y$

- Examples:
  - Regression
  - Classification
  - Object detection

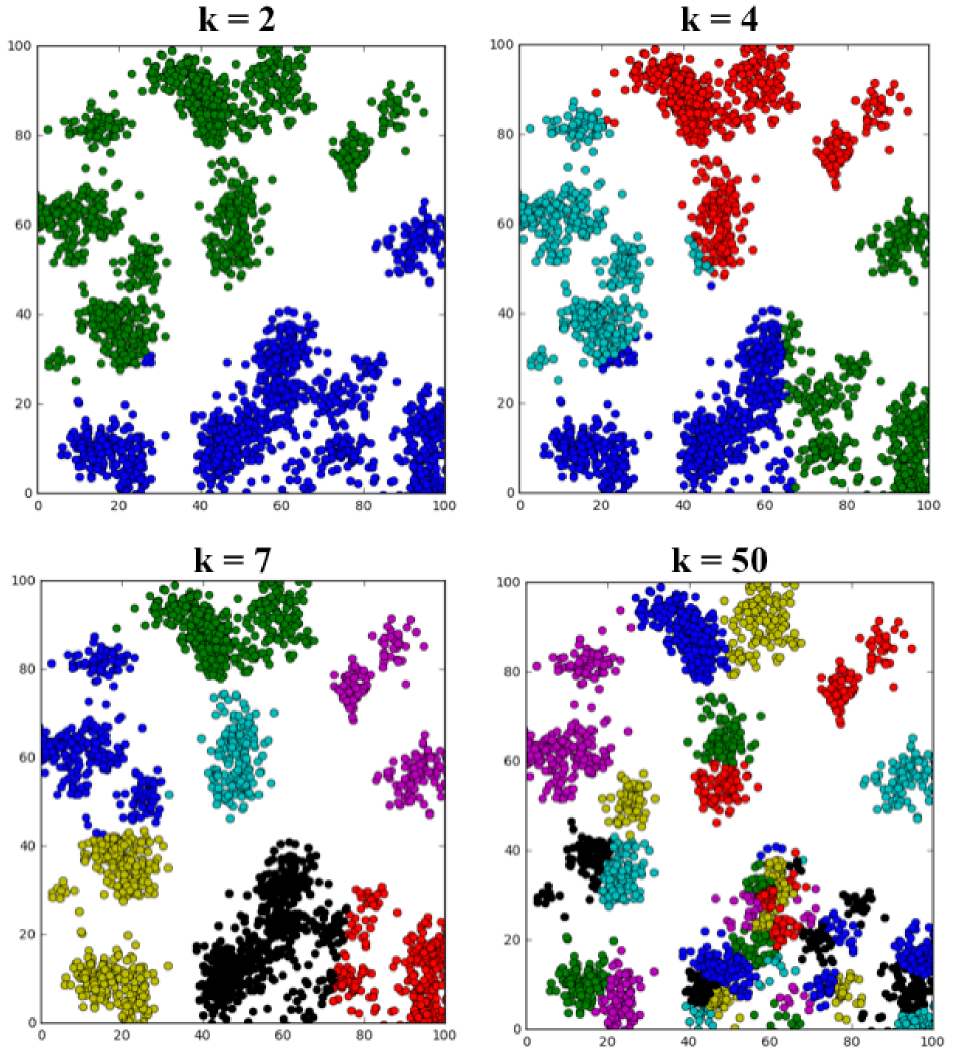


Supervised Learning

- Data: (x, y)
  - Where x is data, y is label
- Goal: Learn a function that maps $f: x \rightarrow y$
- Examples:
  - Regression
  - Classification
  - Object detection

Unsupervised Learning

- Data: x
  - Just data, no labels!
- Goal: Learn some underlying hidden structure of the data
- Examples:
  - Clustering
  - Dimensionality reduction
  - Feature learning
  - Generative model



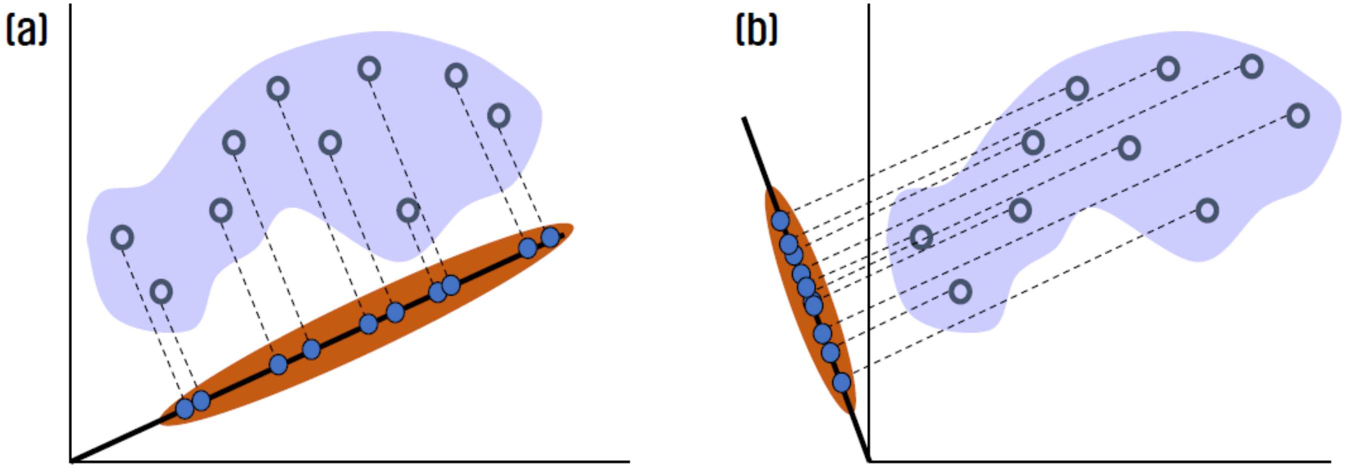
Unsupervised Learning

- Data: x
  - Just data, no labels!

- Goal: Learn some underlying hidden structure of the data

- Examples:
  - Clustering
  - Dimensionality reduction
  - Feature learning
  - Generative model

PCA

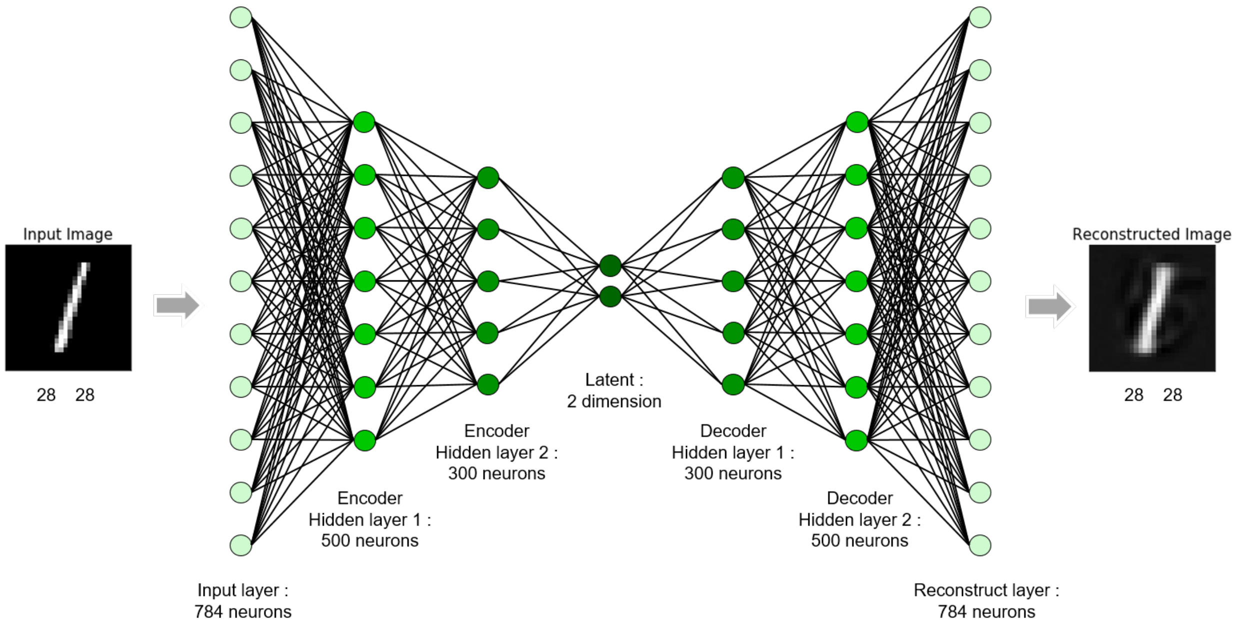


Unsupervised Learning

- Data: x
  - Just data, no labels!

- Goal: Learn some underlying hidden structure of the data

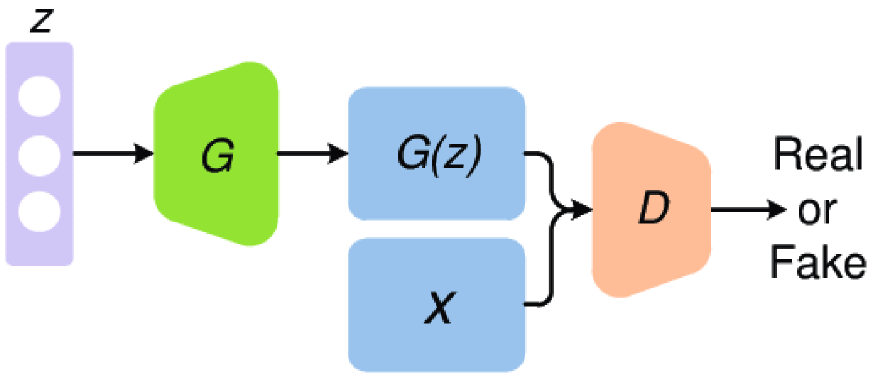
- Examples:
  - Clustering
  - Dimensionality reduction
  - Feature learning
  - Generative model

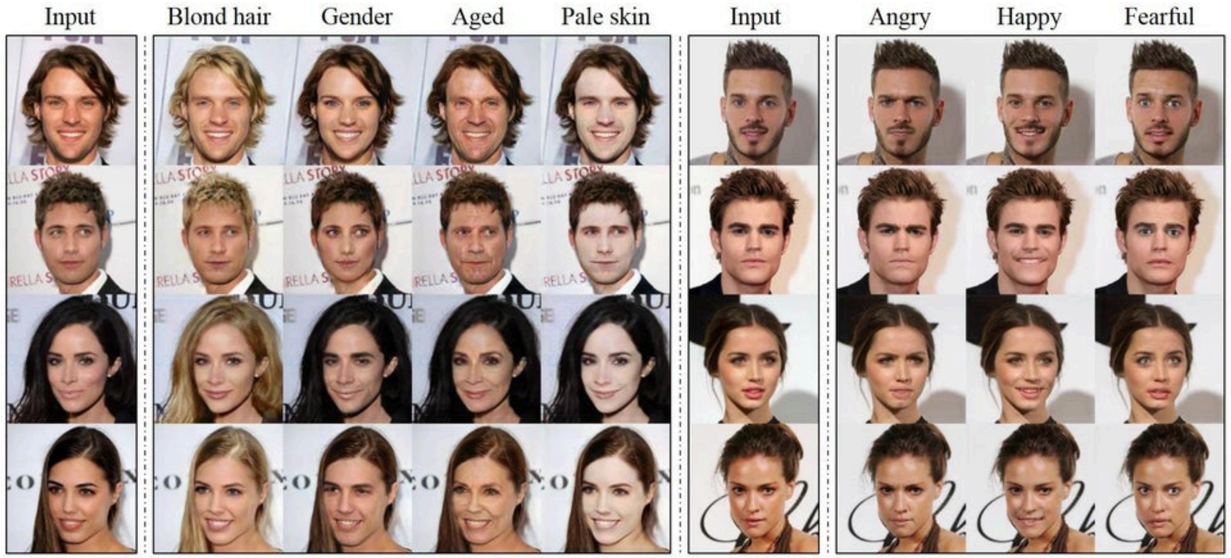


Autoencoder

Unsupervised Learning

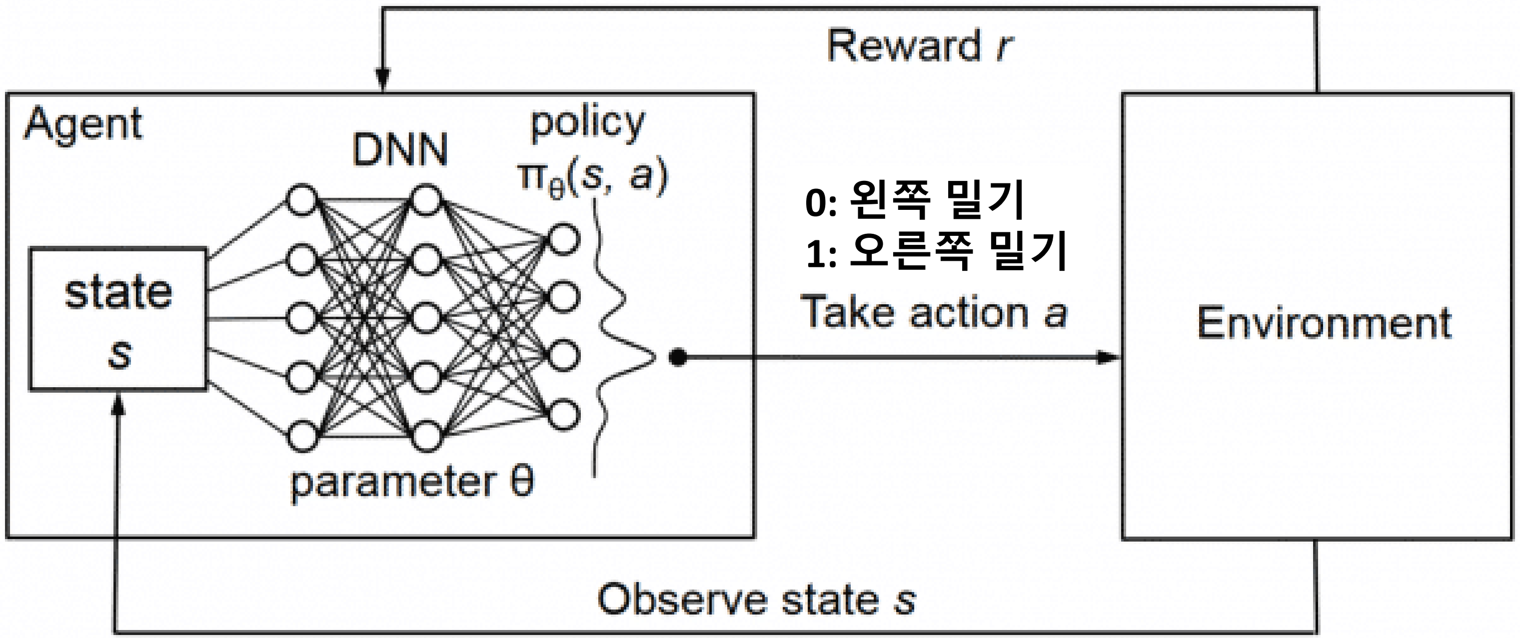
- Data: x
  - Just data, no labels!
- Goal: Learn some underlying hidden structure of the data
- Examples:
  - Clustering
  - Dimensionality reduction
  - Feature learning
  - Generative model

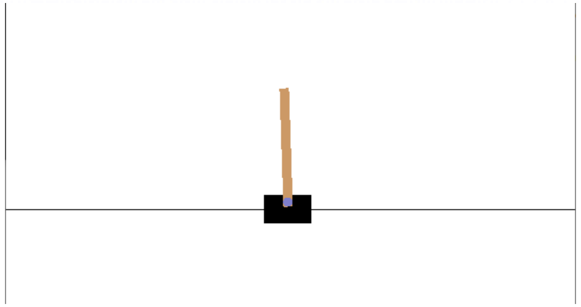




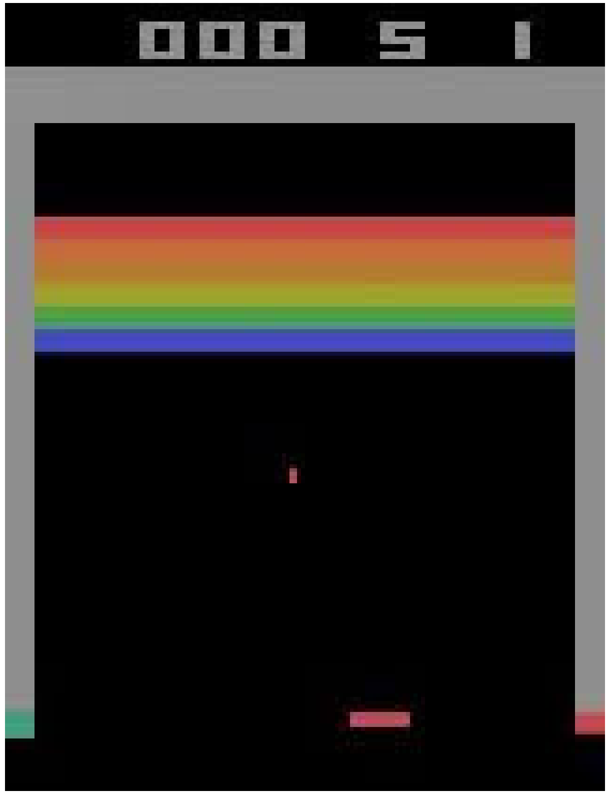
Generative Adversarial Network (GAN)

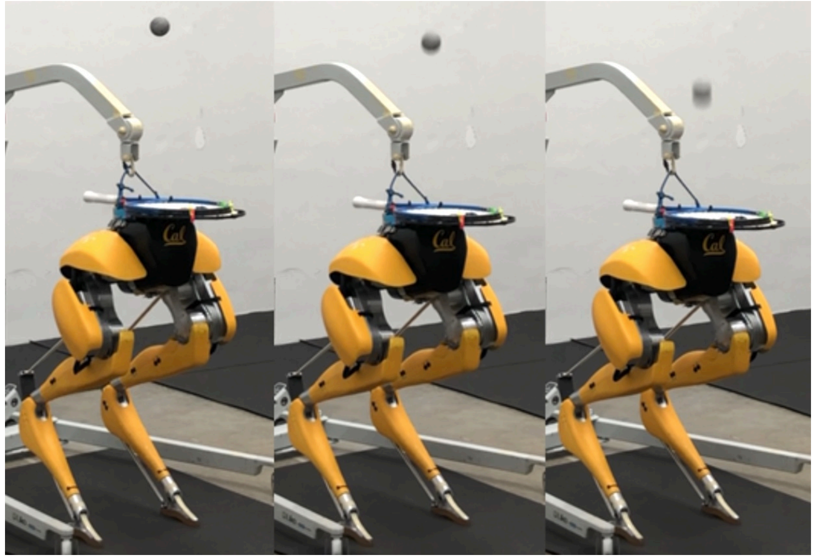
Reinforcement Learning





Reinforcement Learning

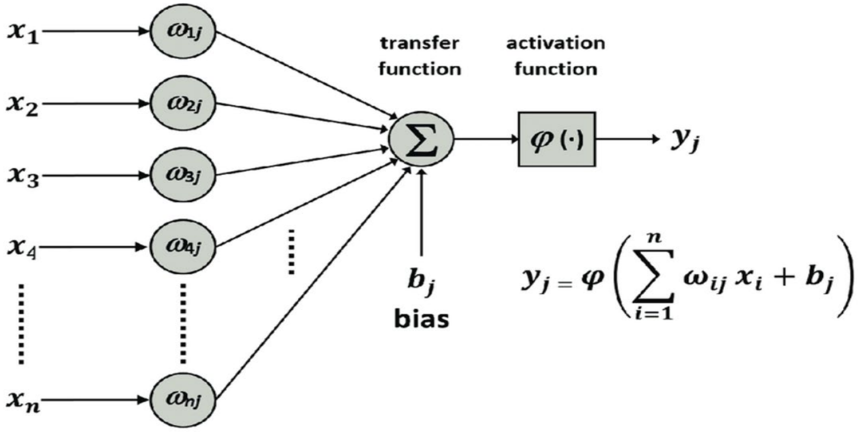


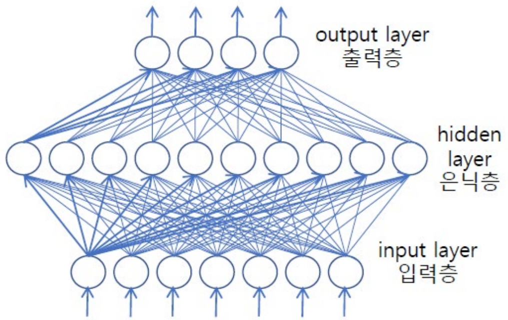


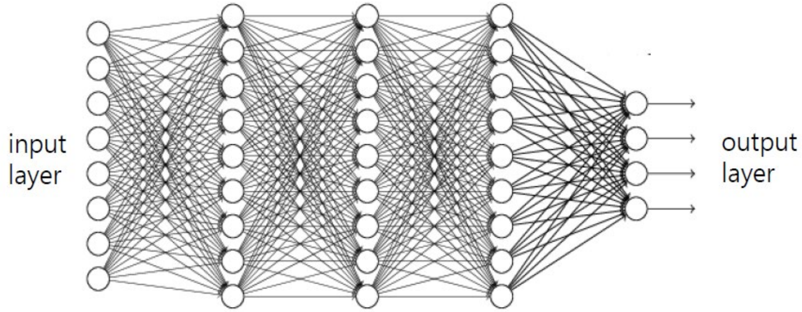
AlphaGo vs. 이세돌

History of deep learning?

- First generation (1958~): perceptrons (F. Rosenblatt, 1958)
  - Criticized by Marvin Minsky about XOR problem
- Second generation (1986~) : multilayer perceptrons
  - Trained by back-propagating error signal
  - Mostly used shallow network with 1 hidden layer
  - MLP and back-propagation algorithm have been experimentally proven and the XOR problem is solved through this (Hinton, 1982)
- Third generation (2006~): deep learning
  - Deep neural network (DNN), convolutional neural network (CNN)

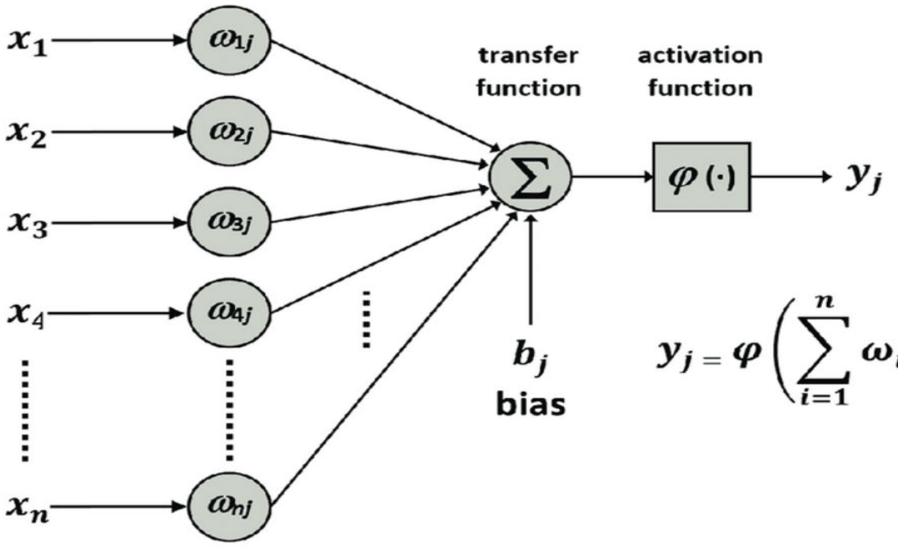




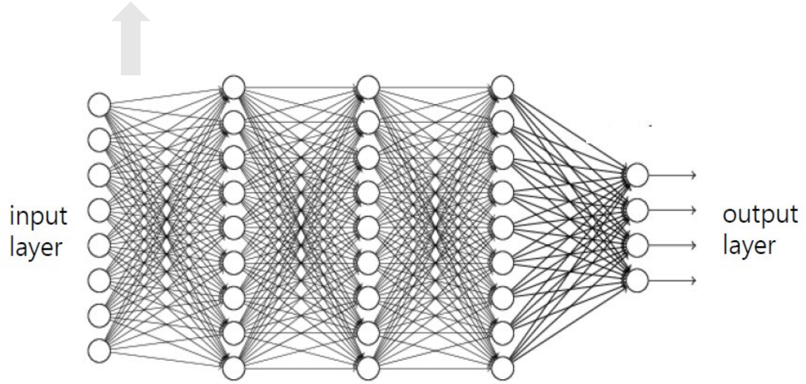


How to train?

✓ **Forward propagation**



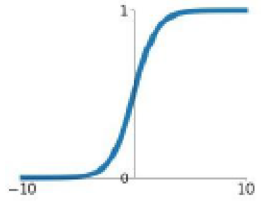
$$y_j = \varphi\left(\sum_{i=1}^{n} \omega_{ij}\, x_i + b_j\right)$$



- Activation function?

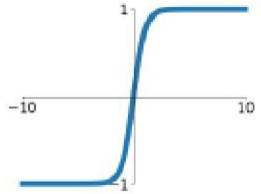
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



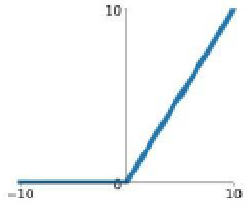
tanh

$\tanh(x)$



ReLU

$\max(0, x)$



How to train?

✓ **Loss function, $E(x, \theta)$**

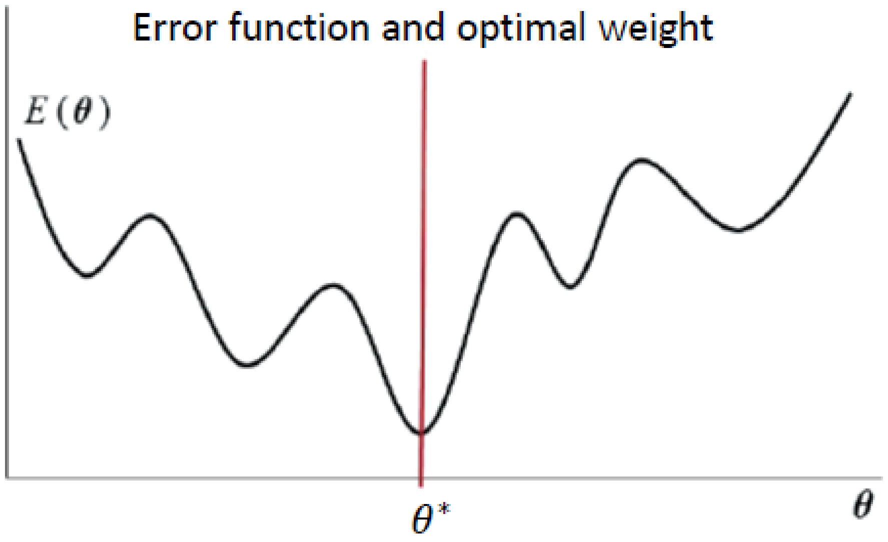
$$MSE = \frac{1}{n}\sum_{i}^{n}\frac{1}{2}(y_i - \tilde{y}_i)^2$$

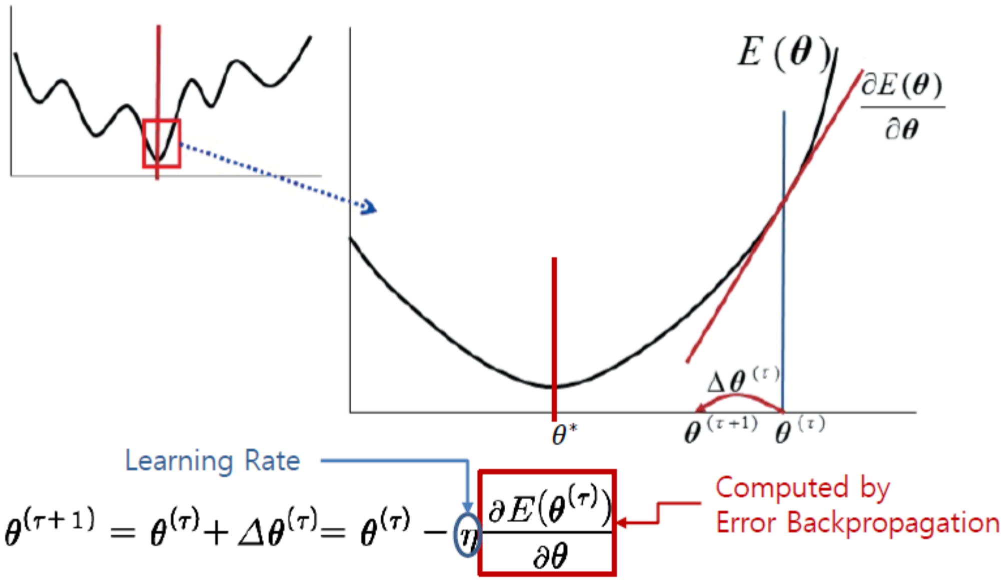
$$CEE = -\sum_{i} y_i log(\tilde{y}_i)$$

- $y = target,\ \tilde{y} = prediction$
- Minimize difference between output $\tilde{y}$ computed from input x and target y
- Mean squared error (MSE) is used for regression problems where the target value is a continuous real number
- Cross entropy error (CEE) is used in classification problems where the target value is 1 or 0
- Optimal weight: θ to minimize error function $E(x, \theta)$

How to train?

- **Goal of learning?**
  - Find an optimal weight $\theta* = argmin_{\theta} E(x, \theta)$
- **Gradient Descent Method**
  - Method for finding $\theta*$ of highly nonlinear function $E(\theta)$
  - Move in direction to minimize error function (loss function) $E(\theta)$

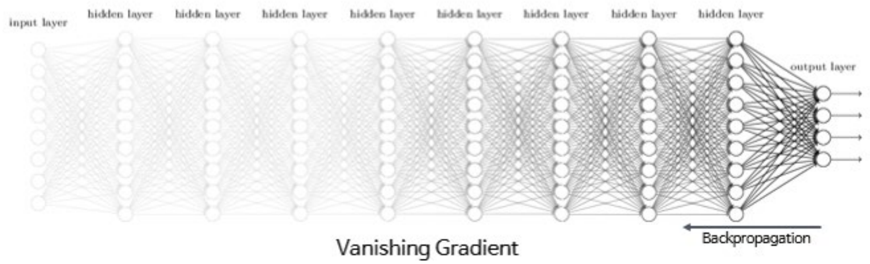




Why DNN had slumped?

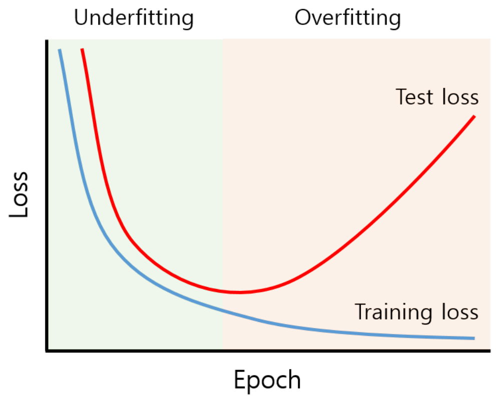
✓ **Vanishing gradient problem by non-linear activation**

- Sigmoid, Hyper-tangent

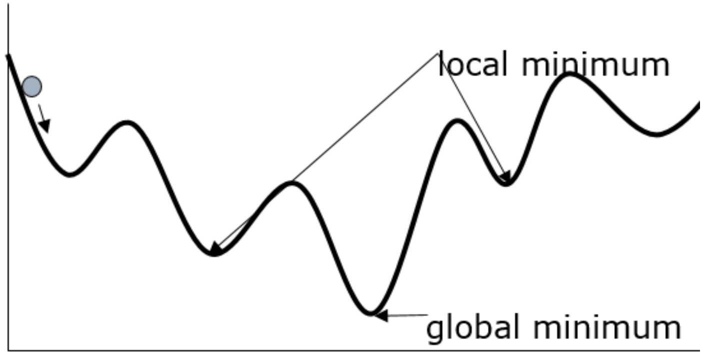


✓ **Overfitting problem**

- Given limited amounts of labeled data, learning does not work well
- Required a lot of labeled data

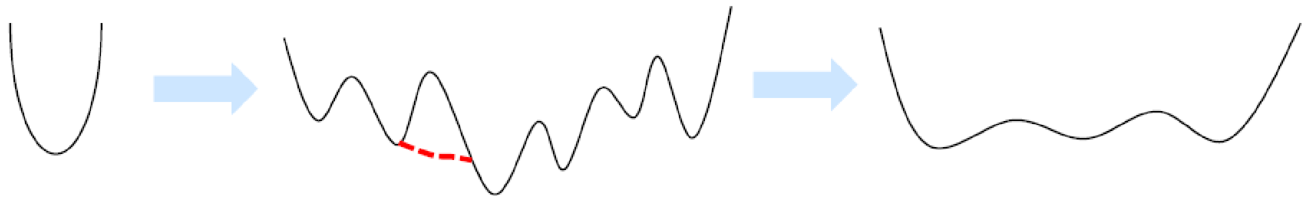


✓ **Get stuck in local minima**



Why DNN had slumped?

- ✓ **Vanishing gradient problem by non-linear activation**
  - ReLU, LSTM, …
- ✓ **Overfitting problem**
  - Dropout
  - Pre-training 기법
- ✓ **Get stuck in local minima**
  - Non-convex optimization in high-dimensional space

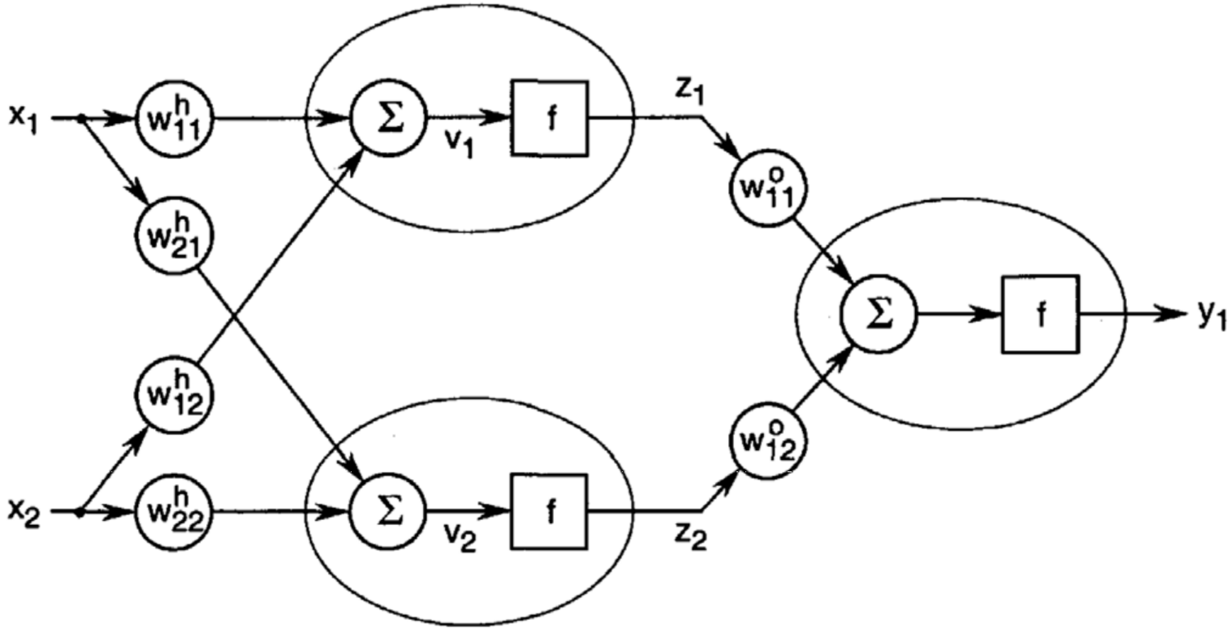


실습 자료

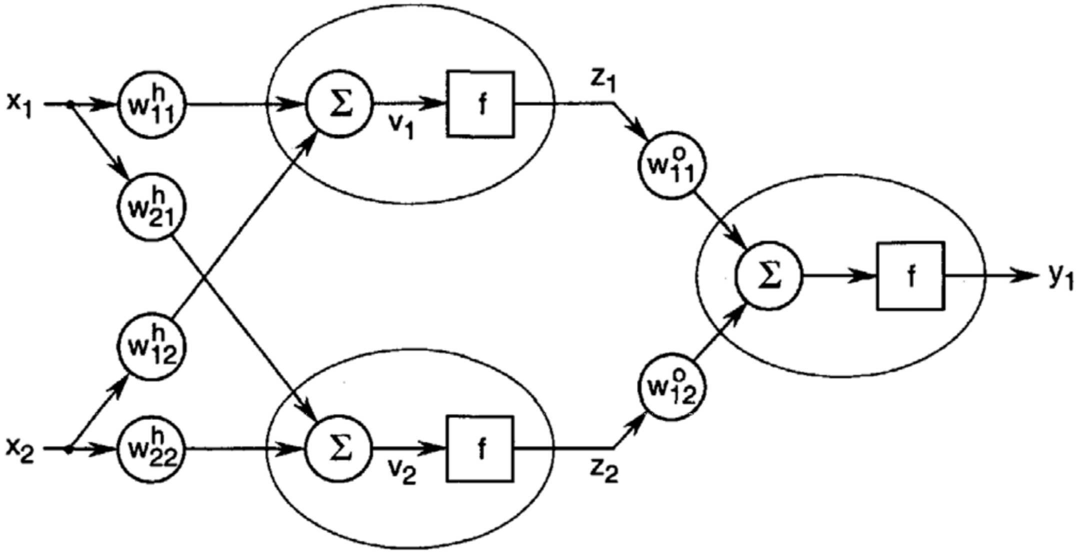
https://colab.research.google.com/drive/1pAFfnvcmk3L_F4DB6z9bZVlHdScHYyUw?usp=share_link

Homework1

- Let us consider a three-layer feedforward network shown below, where the activation functions for all the neurons are given by $f(v) = \frac{1}{(1+e^{-}v)}$.
- There are 2 input nodes, 2 hidden nodes, and 1 output node.



Homework1



- Suppose that the initial weights are $w_{11}^{h(0)} = 0.1,\ w_{12}^{h(0)} = 0.3,\ w_{21}^{h(0)} = 0.3,\ w_{22}^{h(0)} = 0.4,\ w_{11}^{o(0)} = 0.4,\ w_{12}^{o(0)} = 0.6$
- Also, let $x_d = [0.2, 0.6]^T$ and $y_d = 0.7$ be the training example and the target, respectively
- (1) Perform one iteration of the backpropagation algorithm with the step size of $\eta = 10$ and write down the updated weights
    - We use this simple error: $output - target$

Homework2

```
============= [Epochs] 10/10 =============
100%
Train Loss: 2.203221
100%
Test Loss: 2.200249
[Test Result] 21.270%
```

```
============= [Epochs] 10/10 =============
100%
Train Loss: 1.497197
100%
Test Loss: 1.496087
[Test Result] 96.790%
```

- Improve performance by changing **only the model structure**
- I'm not asking you to make the performance the same at 96.79%! Improve performance higher than 21.27%
- Make a PPT presentation of the results of your experiment

Thank you!

Presentation finished