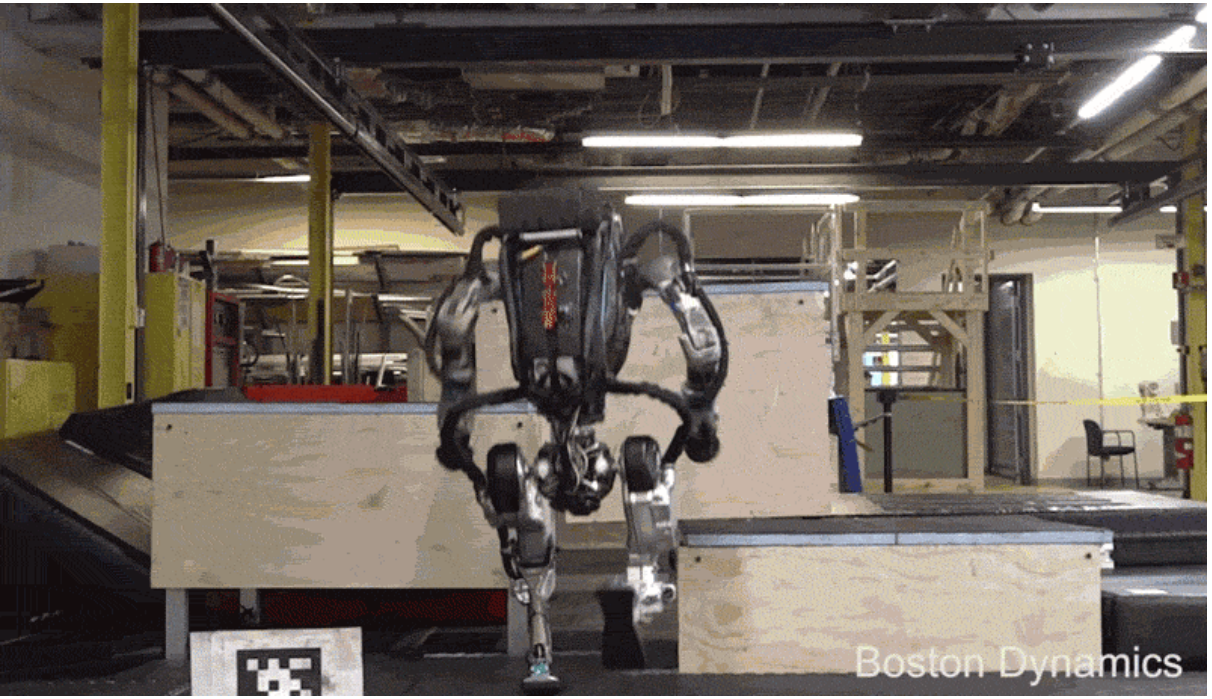
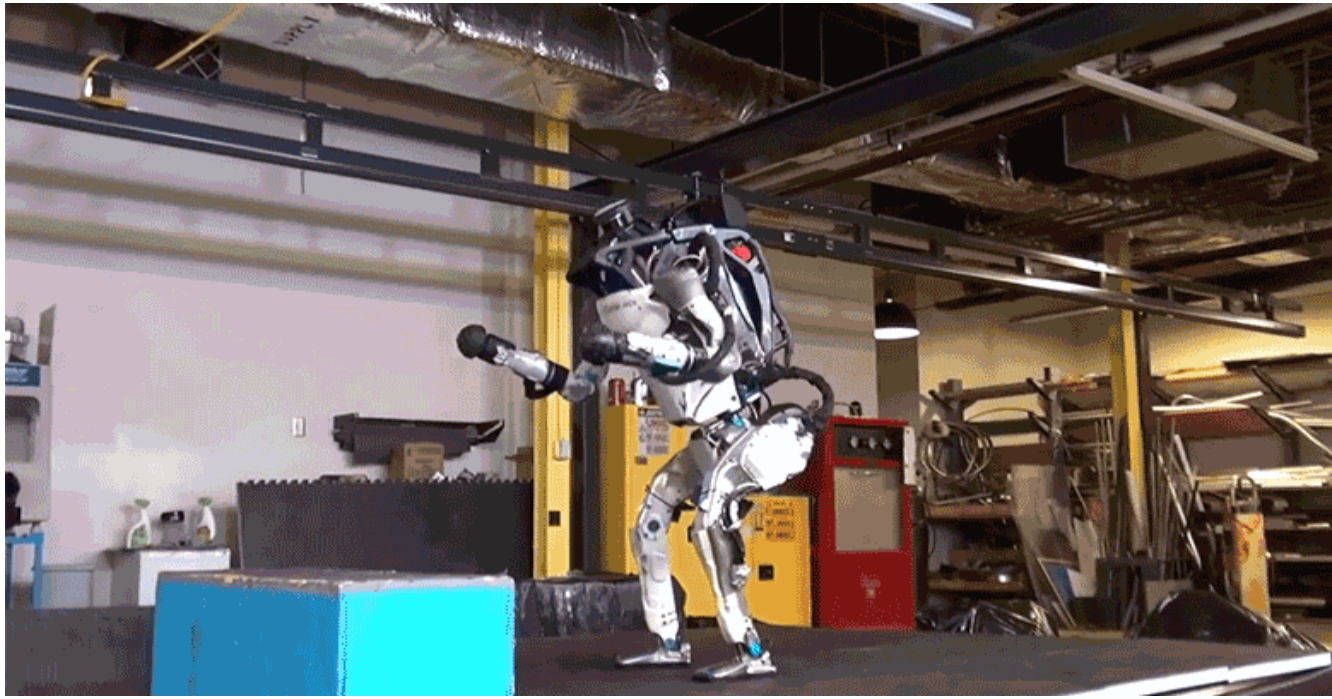
Introduction to Deep Learning

Fast Campus
Start Deep Learning with TensorFlow

2001: A Space Odyssey

Boston Dynamics' Robots





2018 Google Assistant

AI Speaker

AI vs Human

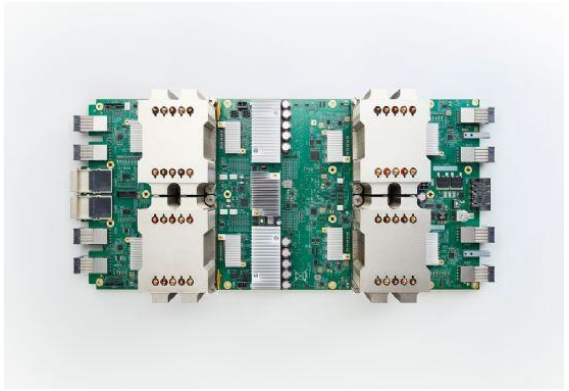
Deep Blue – 2승 3무 1패

Watson

AlphaGo

TPU Server used against Lee Sedol

AlphaGo Lee – 4승 1패



TPU Board used against Ke Jie

AlphaGo Master – 3승 0패

Dota 2 & StarCraft 2

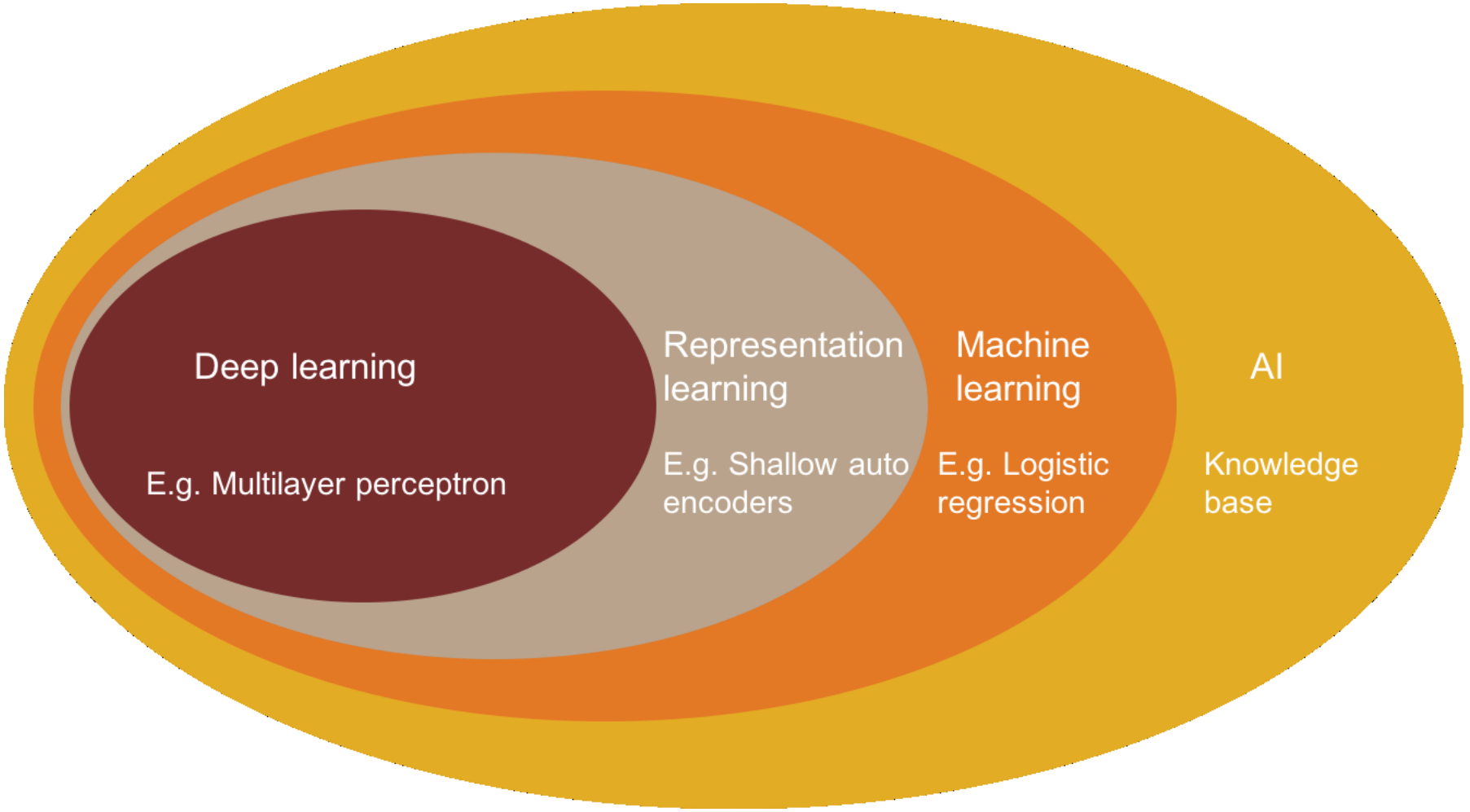
Elon Musk's AI beats the world's best Dota 2 players

Staff Writer 14 August 2017 15 Comments

DeepMind's AlphaStar AI wins 10-1 against professional StarCraft II players

Abner Li - Jan. 24th 2019 12:50 pm PT @technacity

AI, ML, DL


Deep learning
E.g. Multilayer perceptron
Representation learning
E.g. Shallow auto encoders
Machine learning
E.g. Logistic regression
AI
Knowledge base

인공지능(Artificial Intelligence)

- 1956년 다트머스 회의에서 처음 사용
- From Wikipedia:

Artificial intelligence (**AI**) is intelligence exhibited by machines.
In computer science, an ideal "intelligent" machine is a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal

→ AI Translation:

인공 지능 (AI)은 기계가 나타내는 지능입니다. 컴퓨터 과학에서 이상적인 " 지능형 " 기계는 환경을 인식하고 목표 달성의 기회를 극대화하는 유연하고 합리적인 에이전트입니다.

Artificial Intelligence

- 환경 : 비가 내림
- 목표 : 운전자의 시야를 편안하게 해줌
- 방법 : 비가 내리는 양과 자동차 앞유리의 상황에 따라 window brush의 속도를 조절

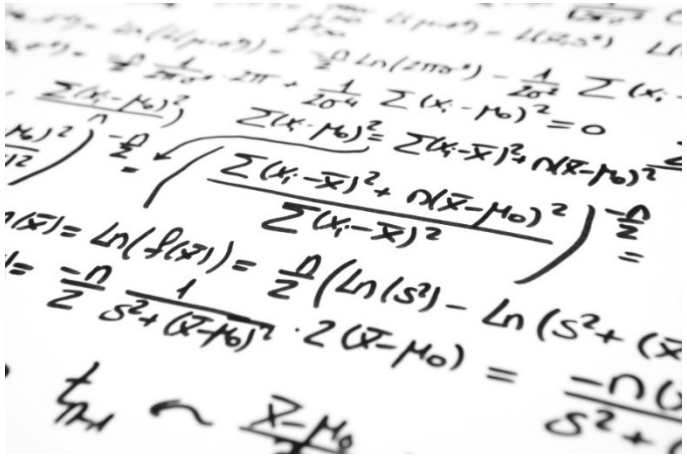
쉬운 것과 어려운 것

- 여우와 두루미

- 사람과 컴퓨터?

컴퓨터에게 쉬운 것과 어려운 것

Easy



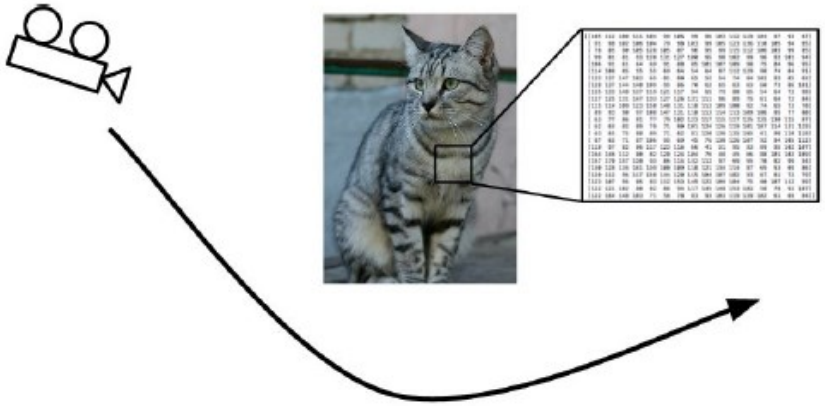
VS

Hard

- 컴퓨터가 잘하는 것은 명확하게 정의된 일, 즉 알고리즘에 대한 수행이다.
- 사람이 진화과정에서 자연스럽게 터득한 것들이 컴퓨터에게는 어렵다.
- 언어의 해상도가 인식의 해상도보다 낮기 때문이다

What is a CAT?

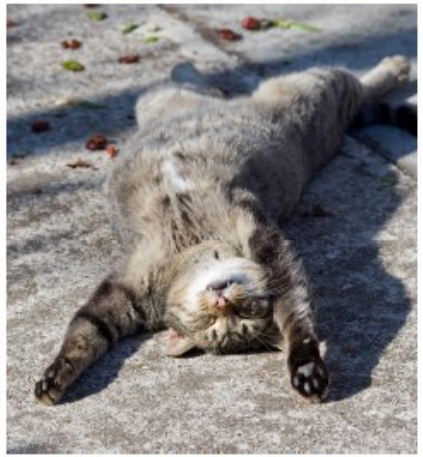
Viewpoint



Illumination

This image is CC0 1.0 public domain

Deformation



This image by Umberto Salvagnin is licensed under CC-BY 2.0

Occlusion

This image by jonsson is licensed under CC-BY 2.0

Clutter

This image is CC0 1.0 public domain

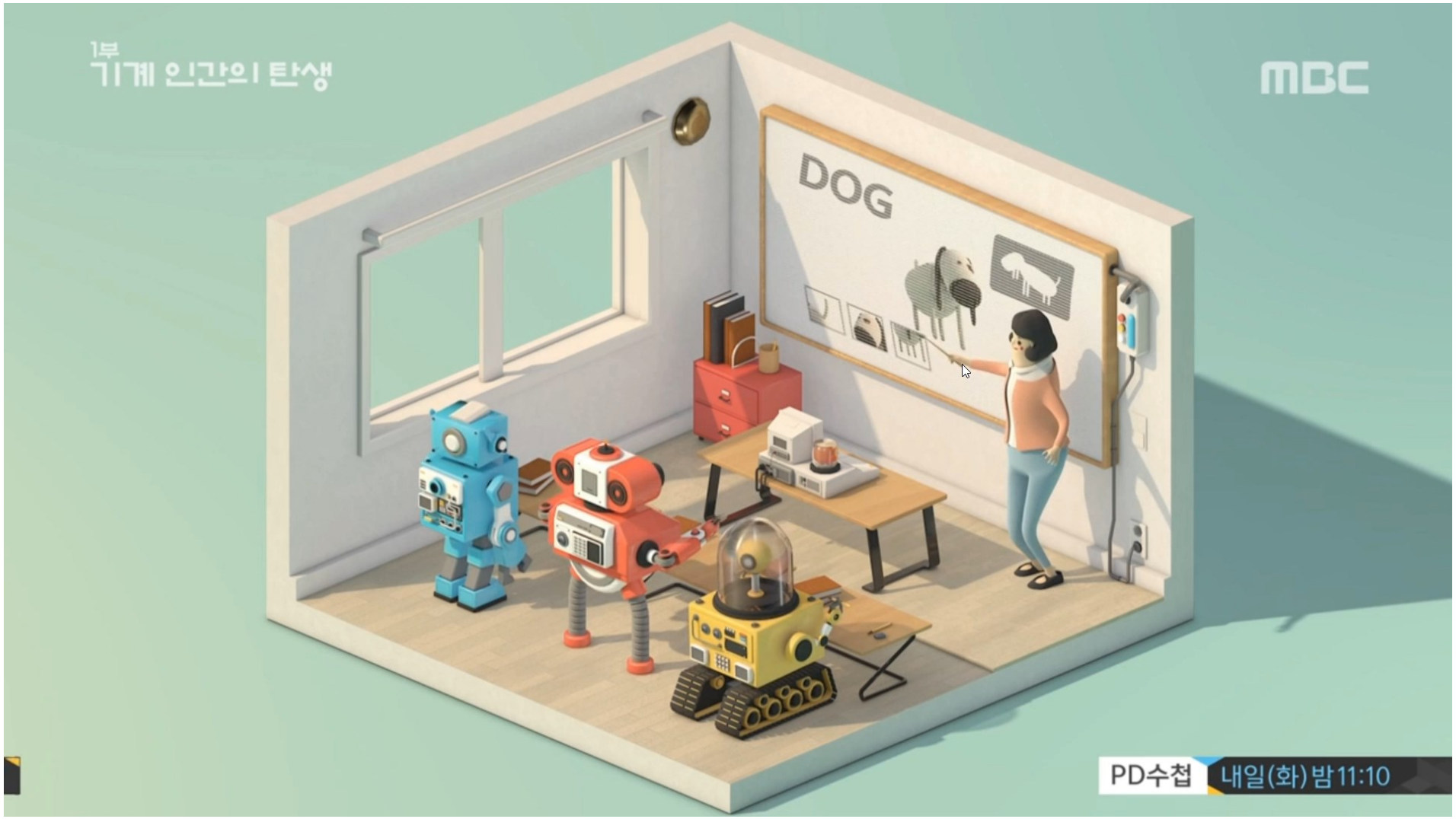
Intraclass Variation

This image is CC0 1.0 public domain

규칙 기반 학습의 부작용(?)

https://sites.google.com/view/totally-looks-like-dataset

어떻게 학습할 것인가?



https://youtu.be/f_uwKZIAeM0

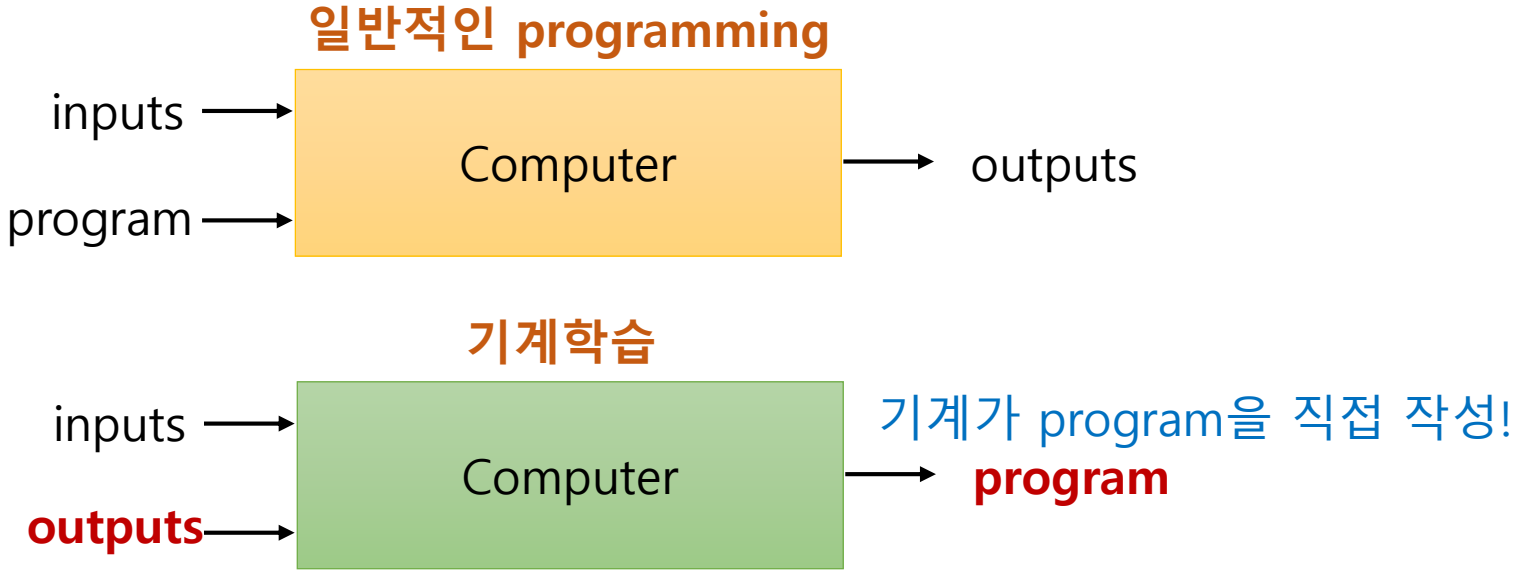
Credit : MBC 기계 인간의 탄생

기계학습(Machine Learning) – Data Driven Approach

- 기계학습(Machine Learning)

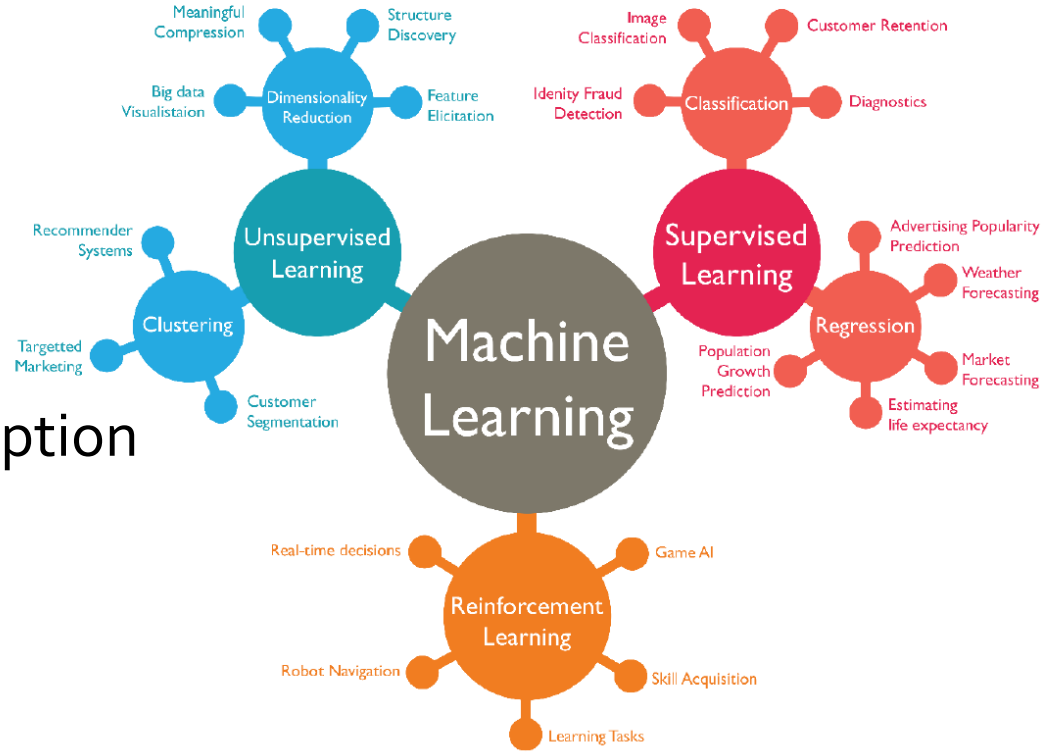
Machine learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed"

기계 학습은 "컴퓨터에 명시적으로 프로그래밍하지 않고 학습 할 수 있는 능력을 부여하는" 컴퓨터 과학의 하위 분야입니다.

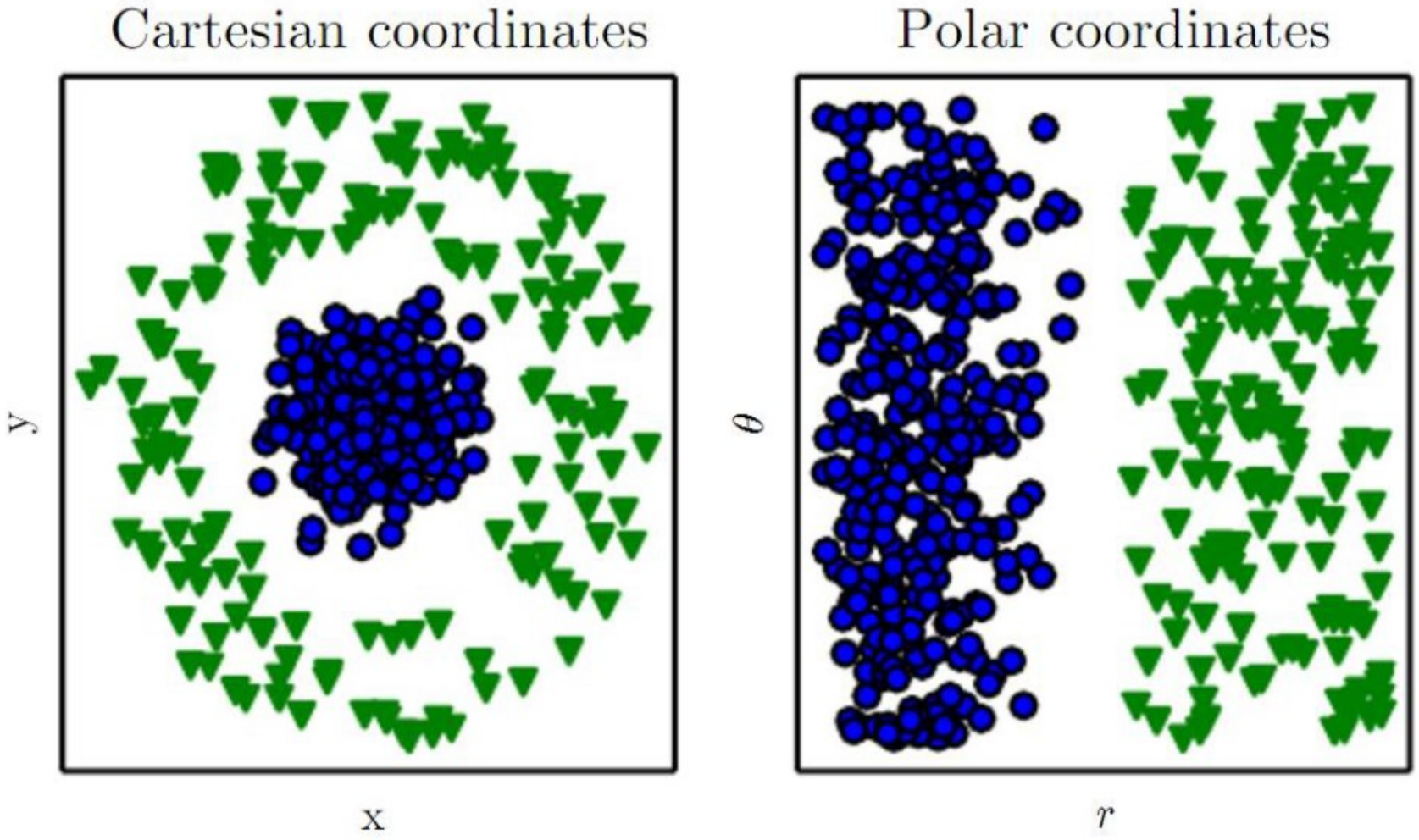


Machine Learning의 종류

- Supervised Learning(지도학습)
  - Input 과 labels을 이용한 학습 → function approximator
  - 분류(classification), 회귀(regression)
- Unsupervised Learning(비지도학습)
  - Input만을 이용한 학습 → (shorter) description
  - 군집화(clustering), 압축(compression)
- Reinforcement Learning(강화학습)
  - Trial and error를 통한 학습 → sequential decision making
  - Action selection, policy learning

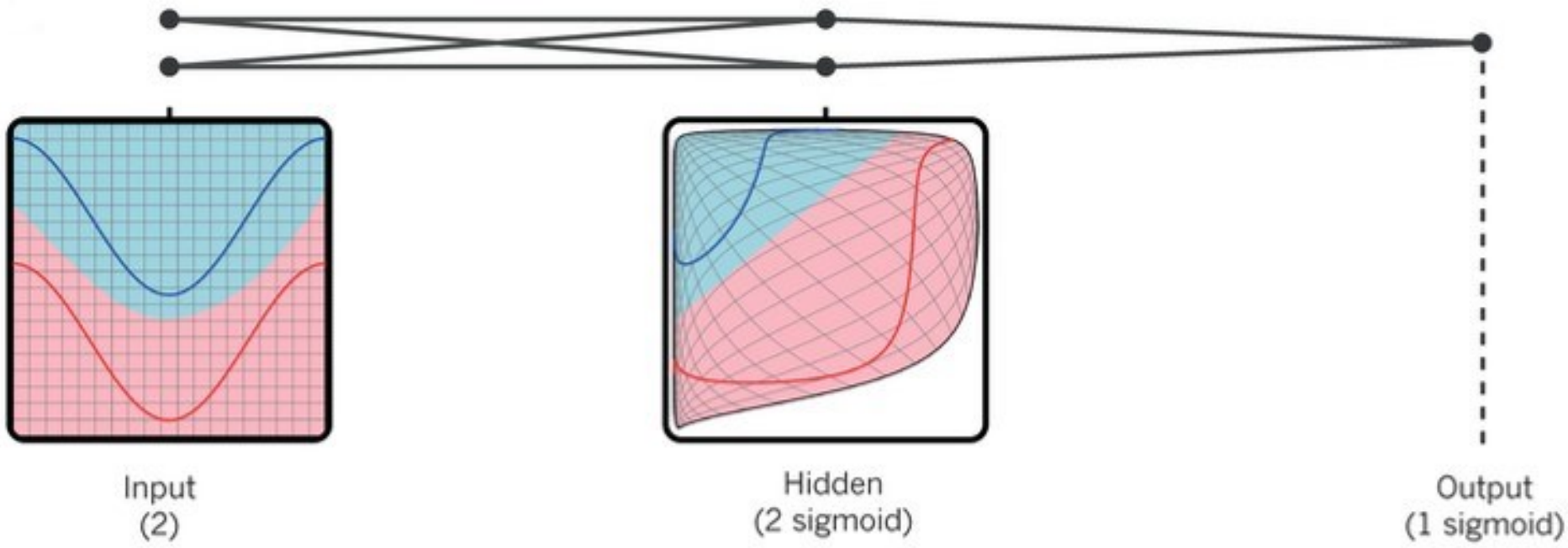


Representation Matters



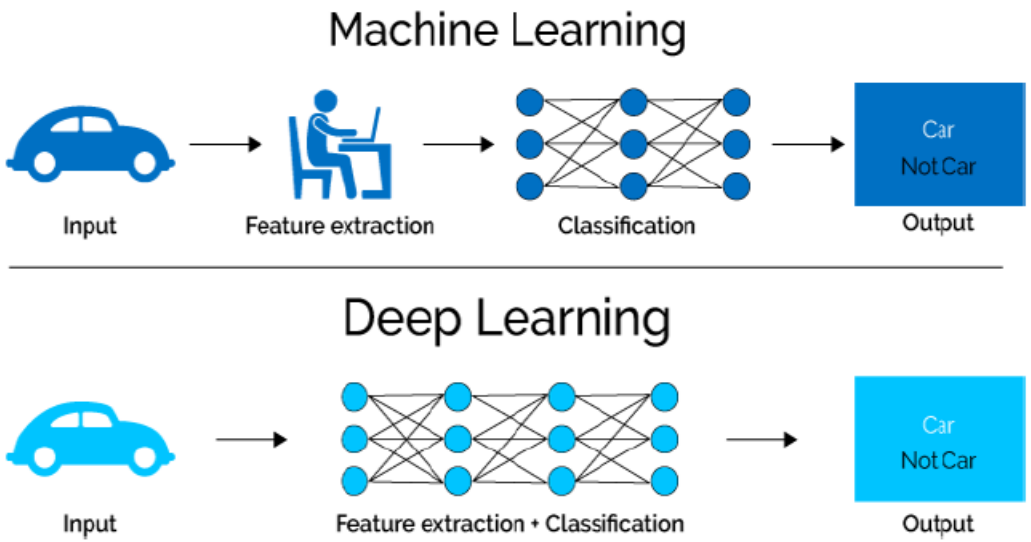
Task: Draw a line to separate the **green triangles** and **blue circles**.

Representation Learning



Task: Draw a line to separate the **blue curve** and **red curve**

Machine Learning vs Deep Learning



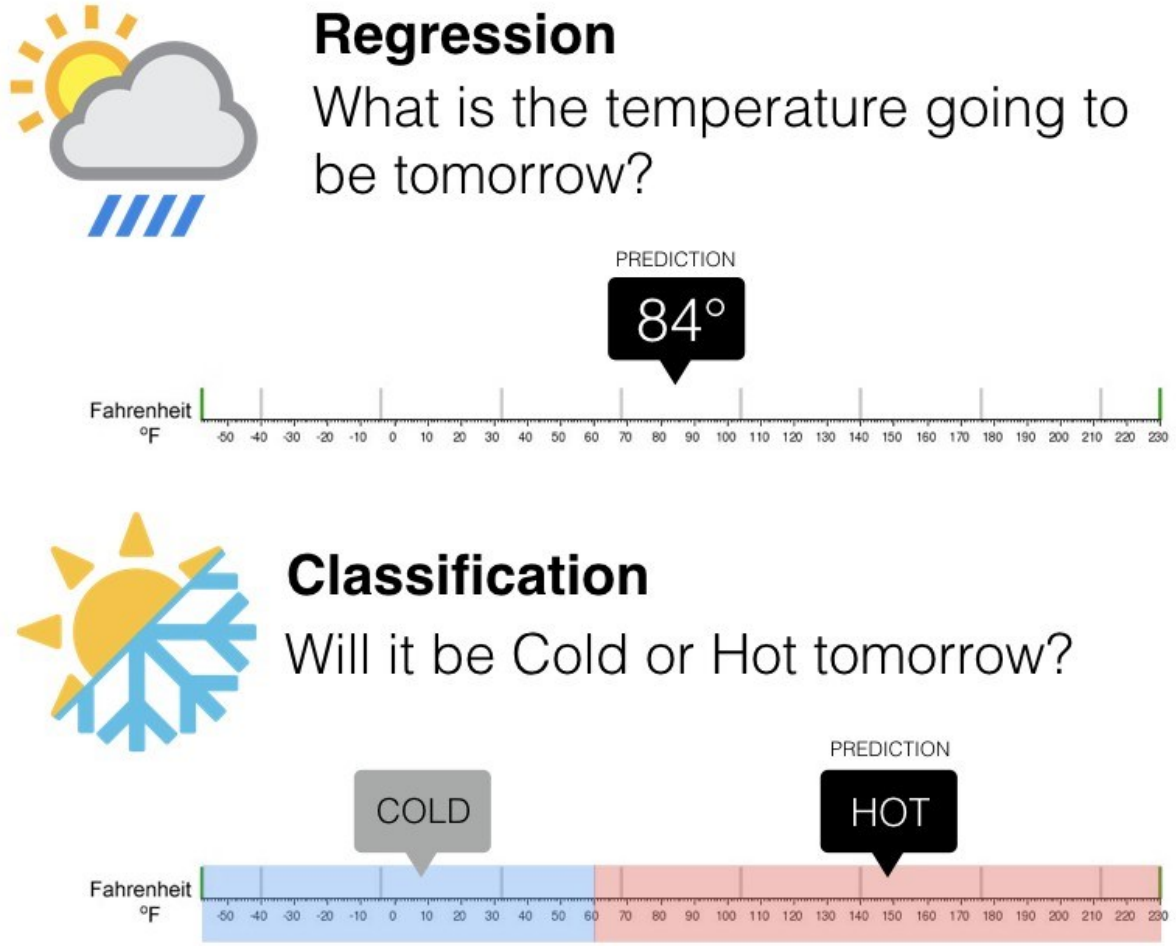
ML : Audio → Acoustic Model → Phonetic Model → Language Model → Text

DL : Audio → Deep Networks → Text (End to End Learning)

ML : Pixels → Key Points → SIFT Features → Deformable Part Model → Label

DL : Pixels → Deep Networks → Label (End to End Learning)

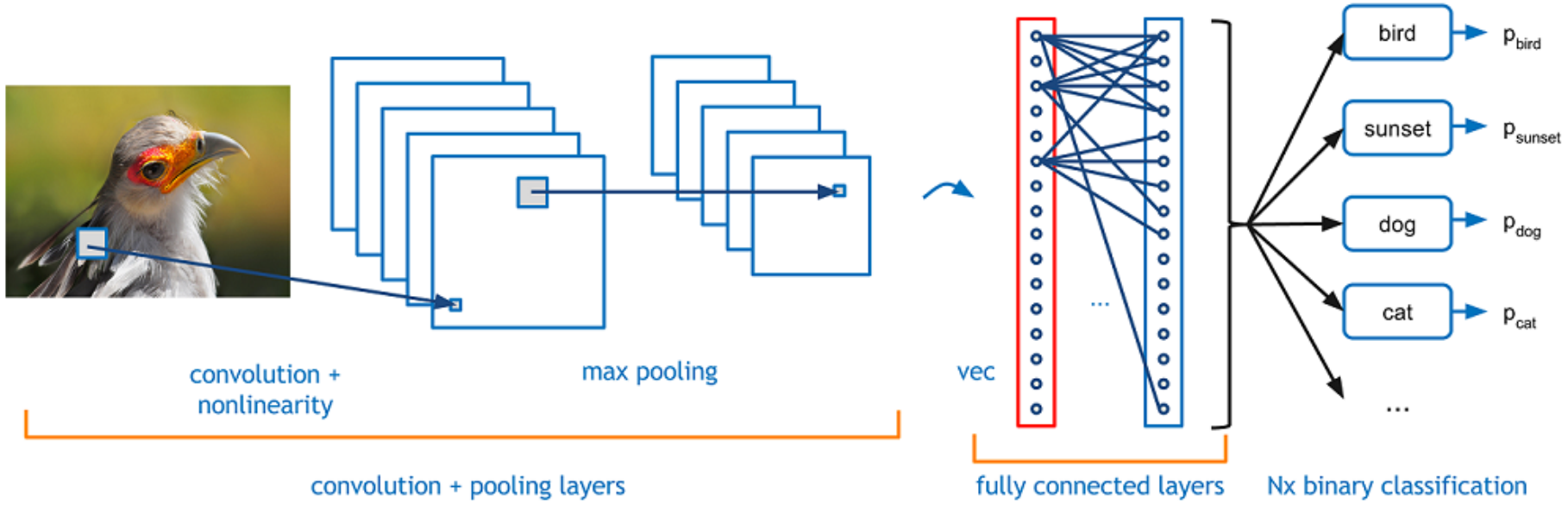
Supervised Learning – Regression vs Classification



Convolutional Neural Network

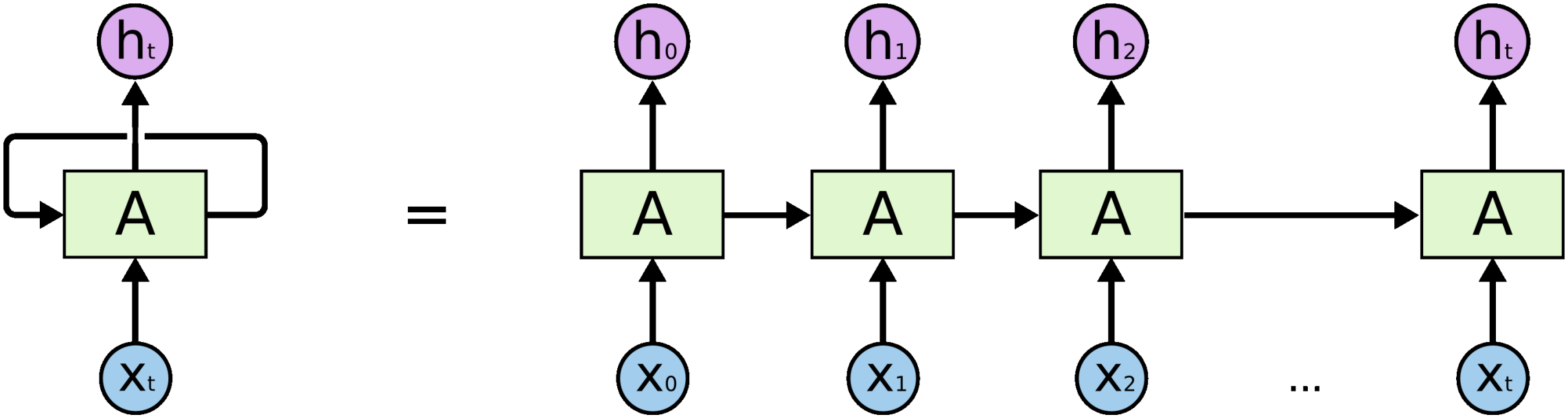
What We See

What Computers See



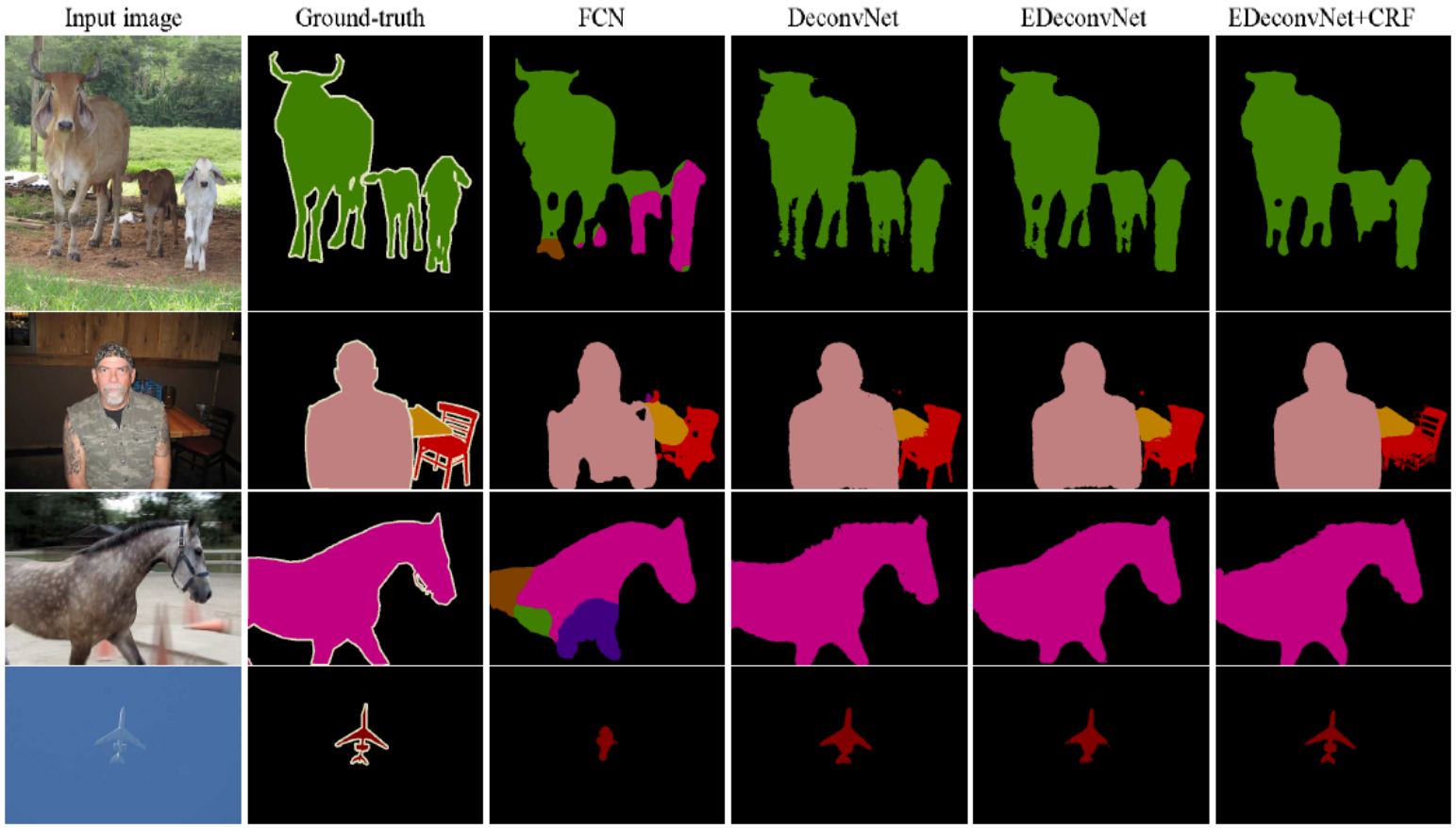
Recurrent Neural Network

- 알파벳 순서를 거꾸로 말하기 어려운 이유는?



Object Detection

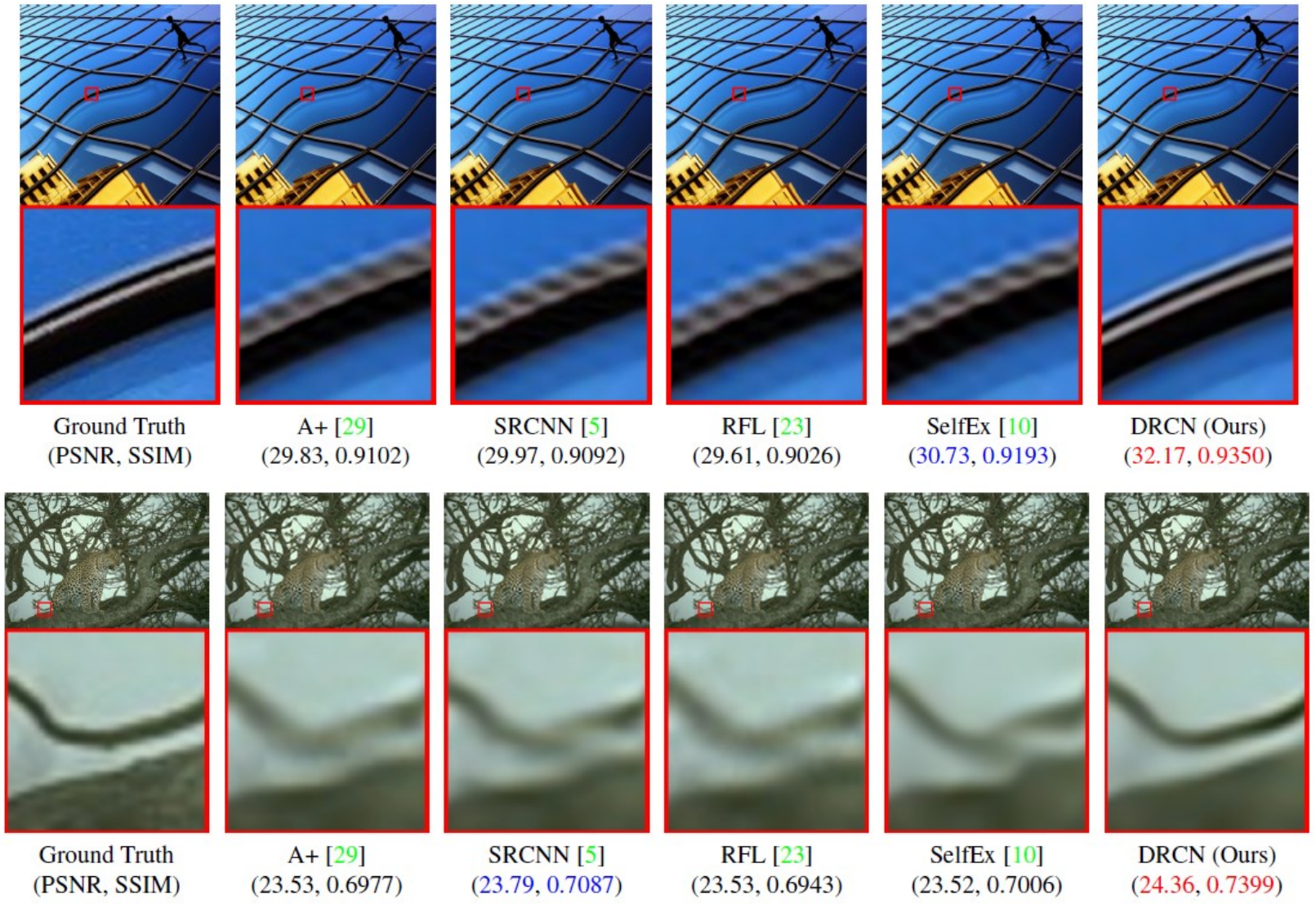
Segmentation



(a) Examples that our method produces better results than FCN [19].

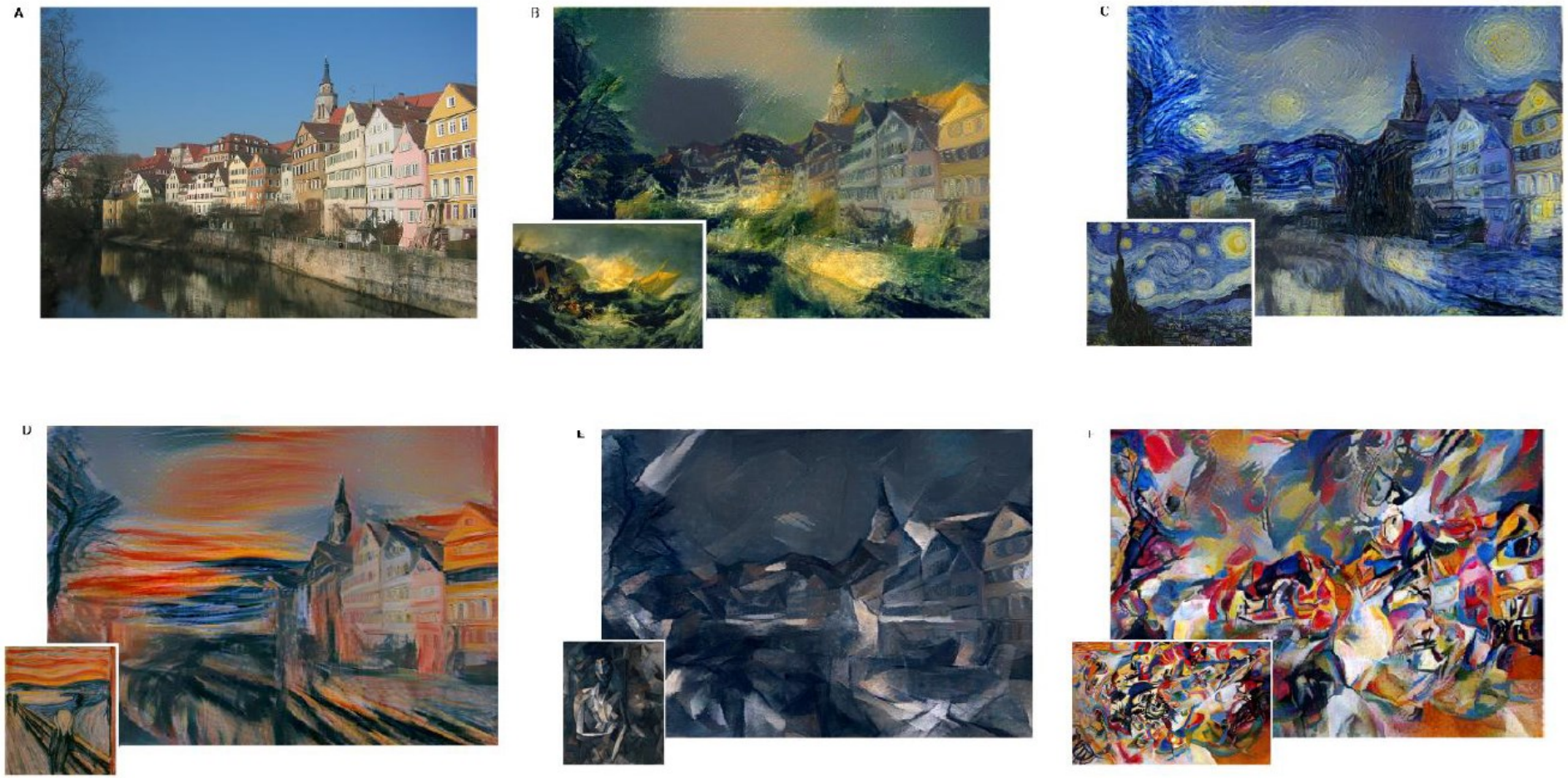
"Learning Deconvolution Network for Semantic Segmentation

Super Resolution



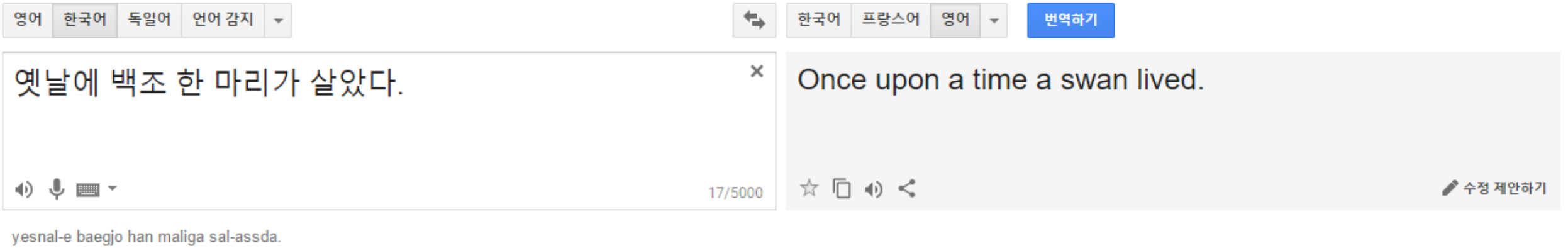
"Deeply-Recursive Convolutional Network for Image Super-Resolution"

Artistic Style Transfer



"Image Style Transfer Using Convolutional Neural Networks"

Machine Translation



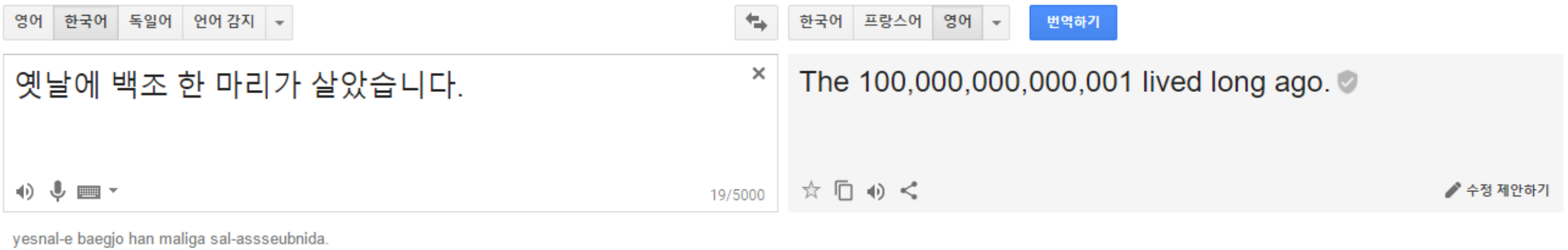
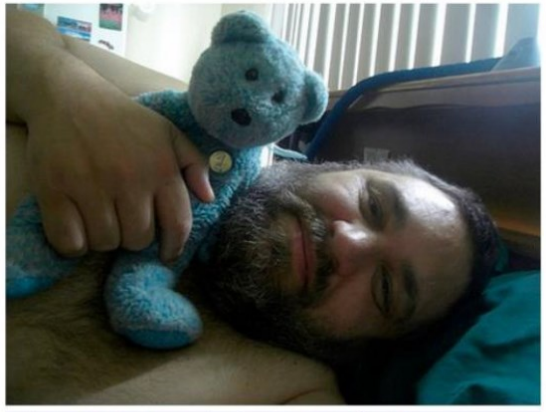


Image Captioning

a group of people standing around a room with remotes
logprob: -9.17

a young boy is holding a baseball bat
logprob: -7.61

a cow is standing in the middle of a street
logprob: -8.84



a baby laying on a bed with a stuffed bear
logprob: -8.66

a young boy is holding a baseball bat
logprob: -7.65

a woman holding a teddy bear in front of a mirror
logprob: -9.65

Visual QnA

Q: What is the boy holding?

DPPnet: surfboard

DPPnet: bat

Q: What is the animal doing?

DPPnet: resting (relaxing)

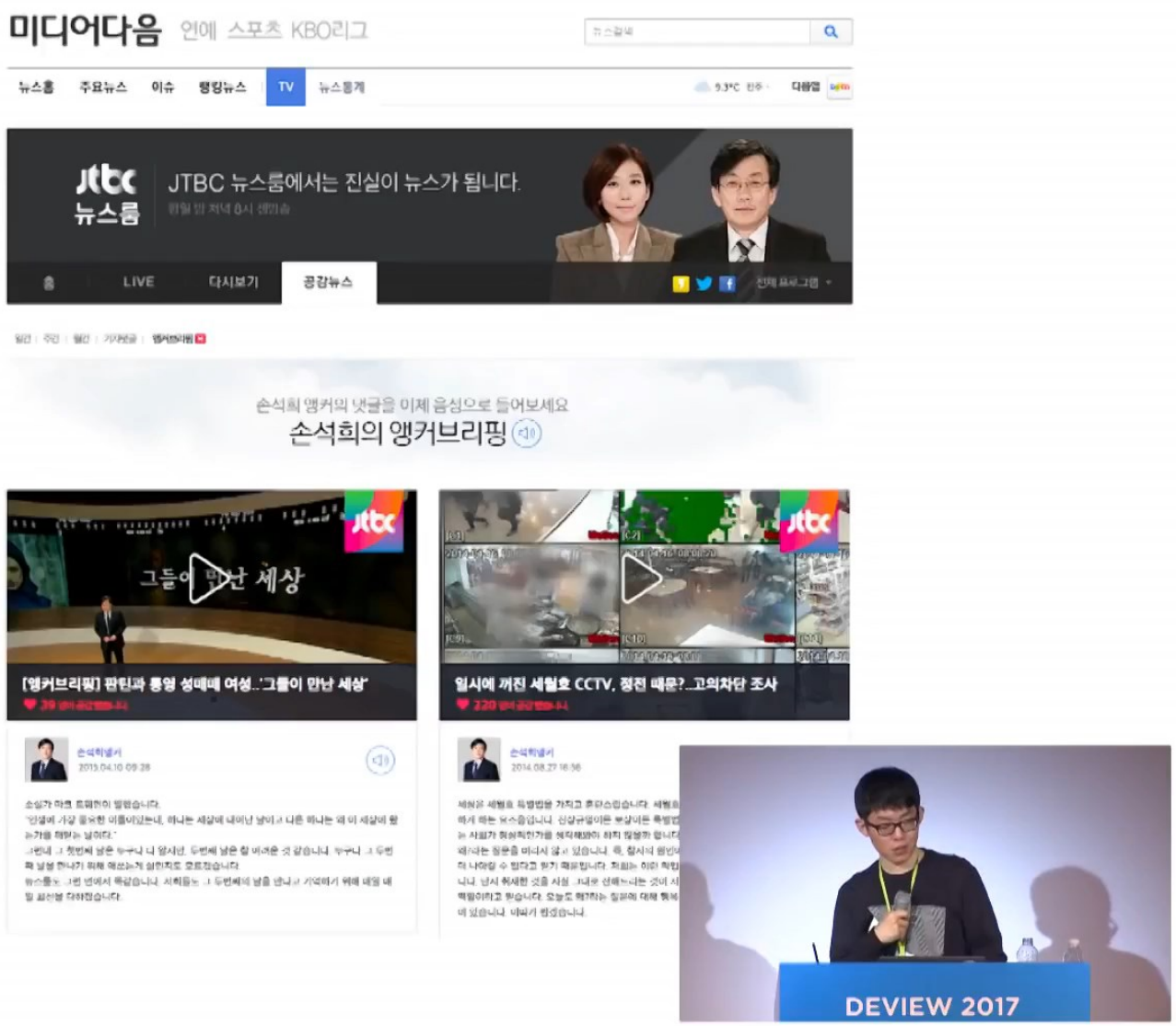
DPPnet: swimming (fishing)

"Image Question Answering using Convolutional Neural Network with Dynamic Parameter Prediction"

Auto Speech Recognition

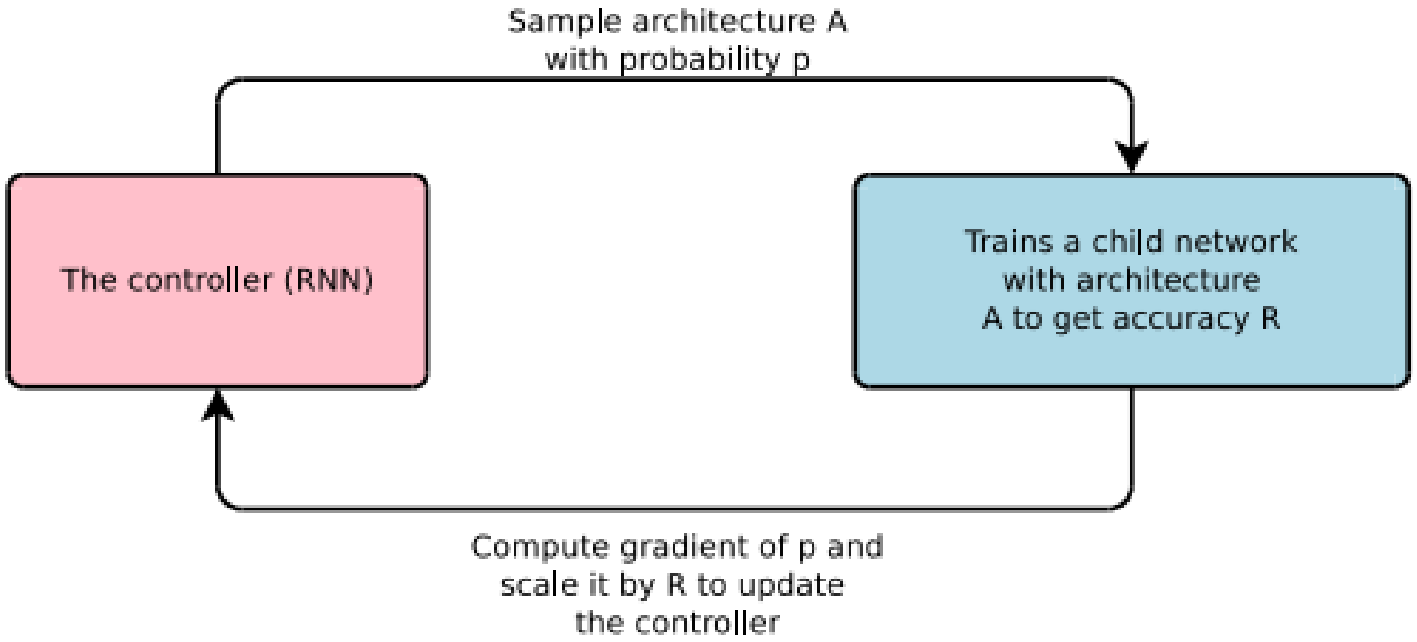
Text to Speech

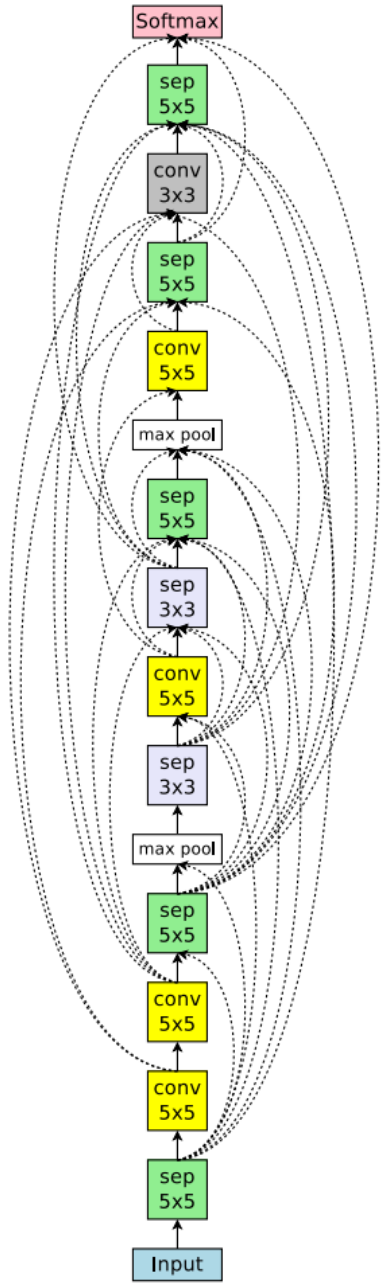
kakao



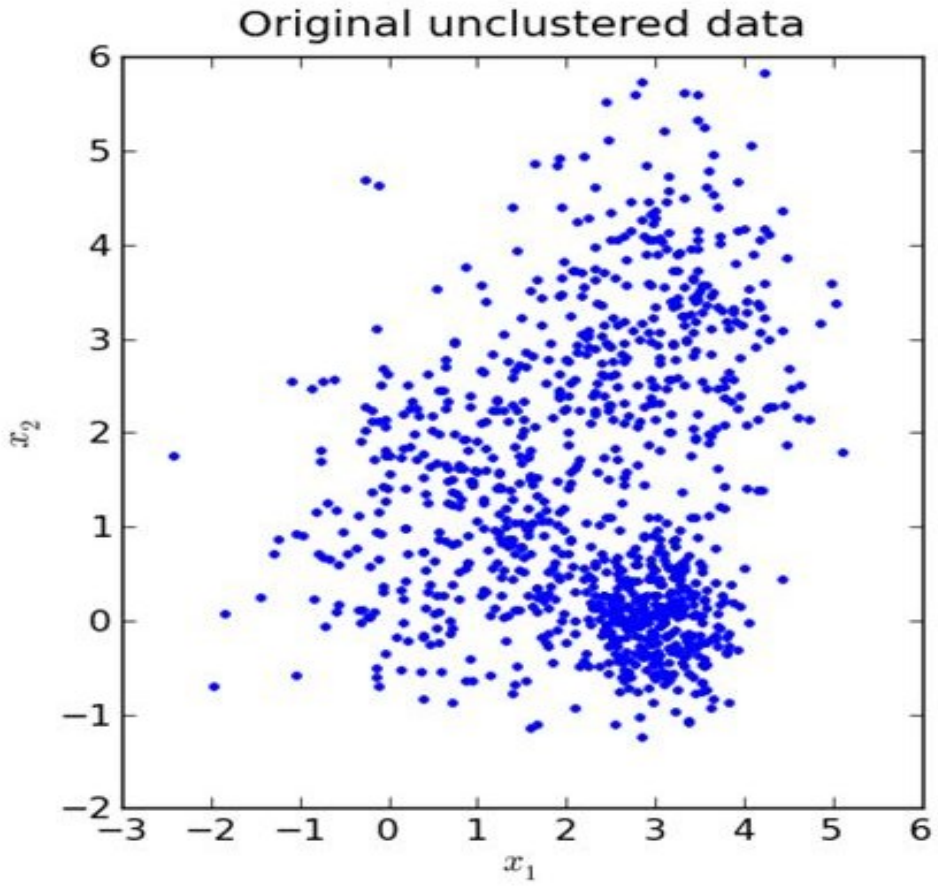
Auto ML – Neural Architecture Search

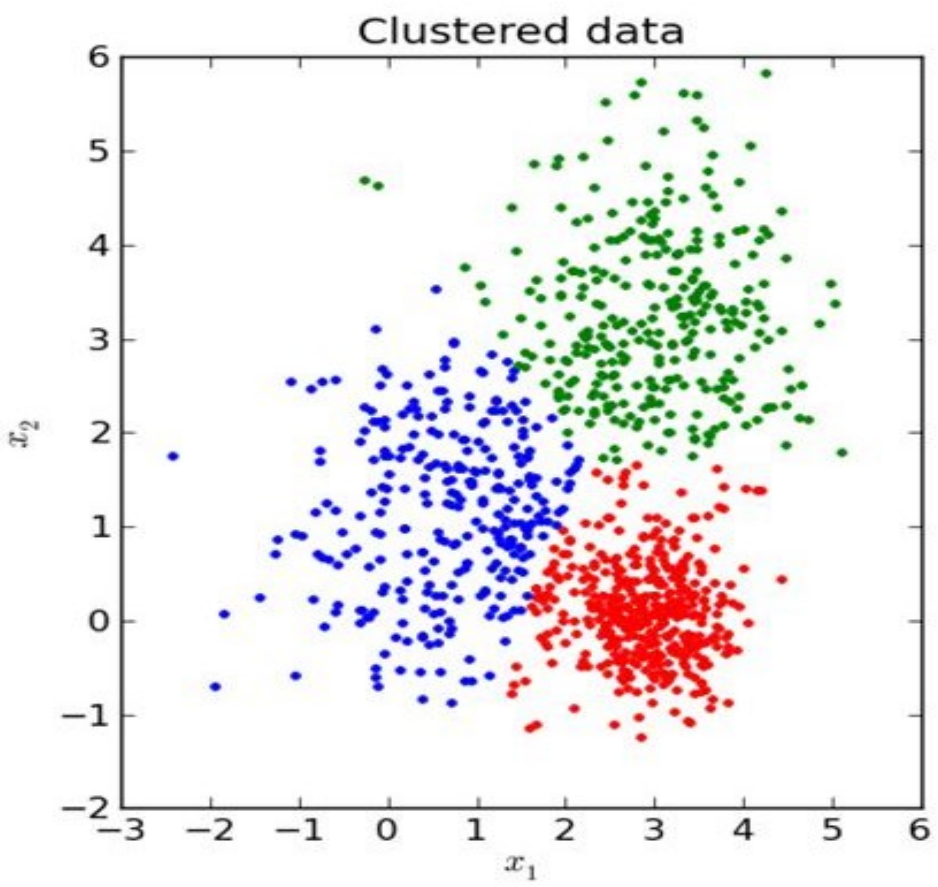
- Leaning to Learn





Unsupervised Learning





Unsupervised Learning

- 아래 음식을 둘로 분류하면?


(1)


(2)


(3)


(4)


(5)

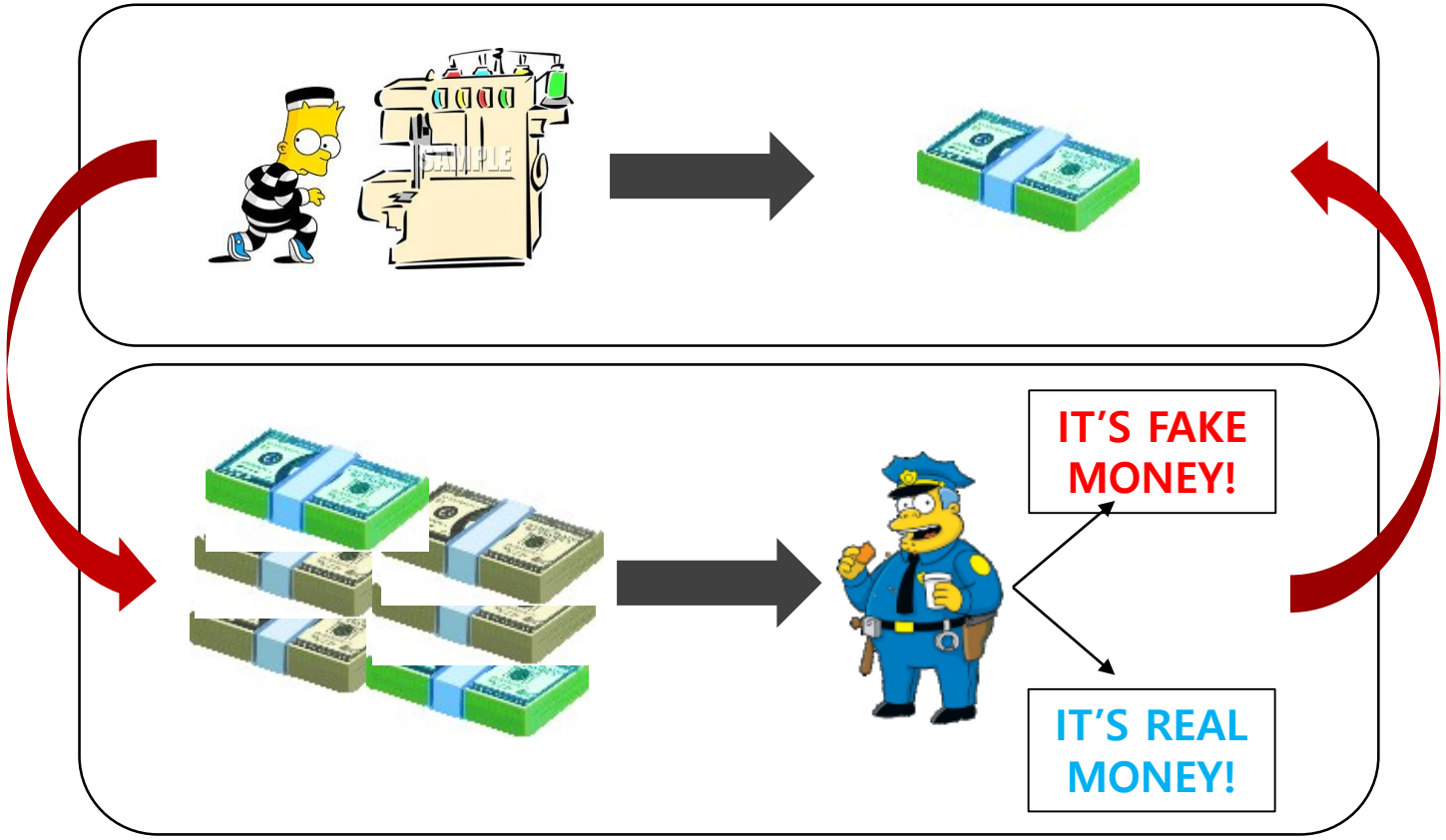

(6)


(7)

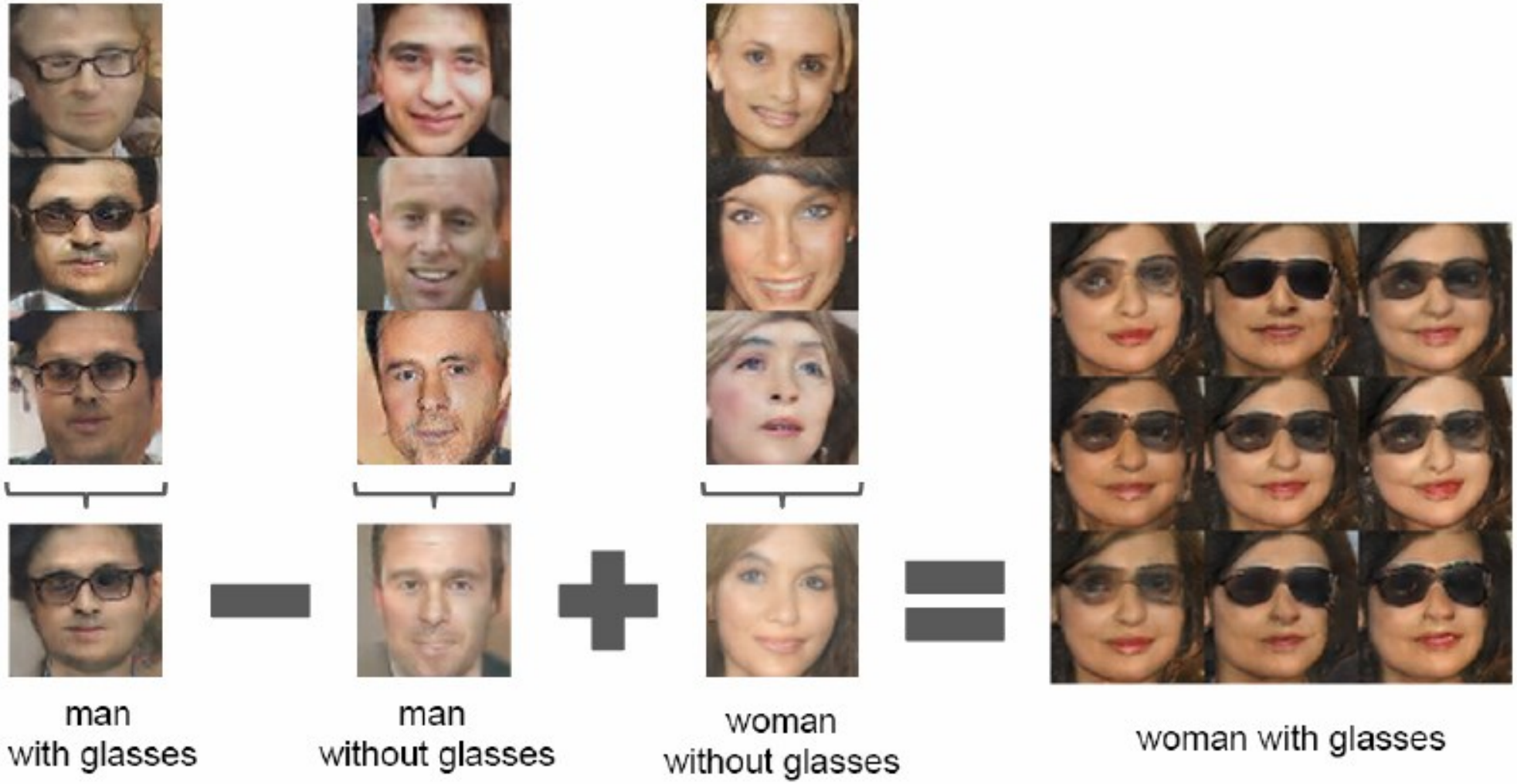

(8)

Generative Adversarial Network

- Counterfeiters vs Police Game



DCGAN


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

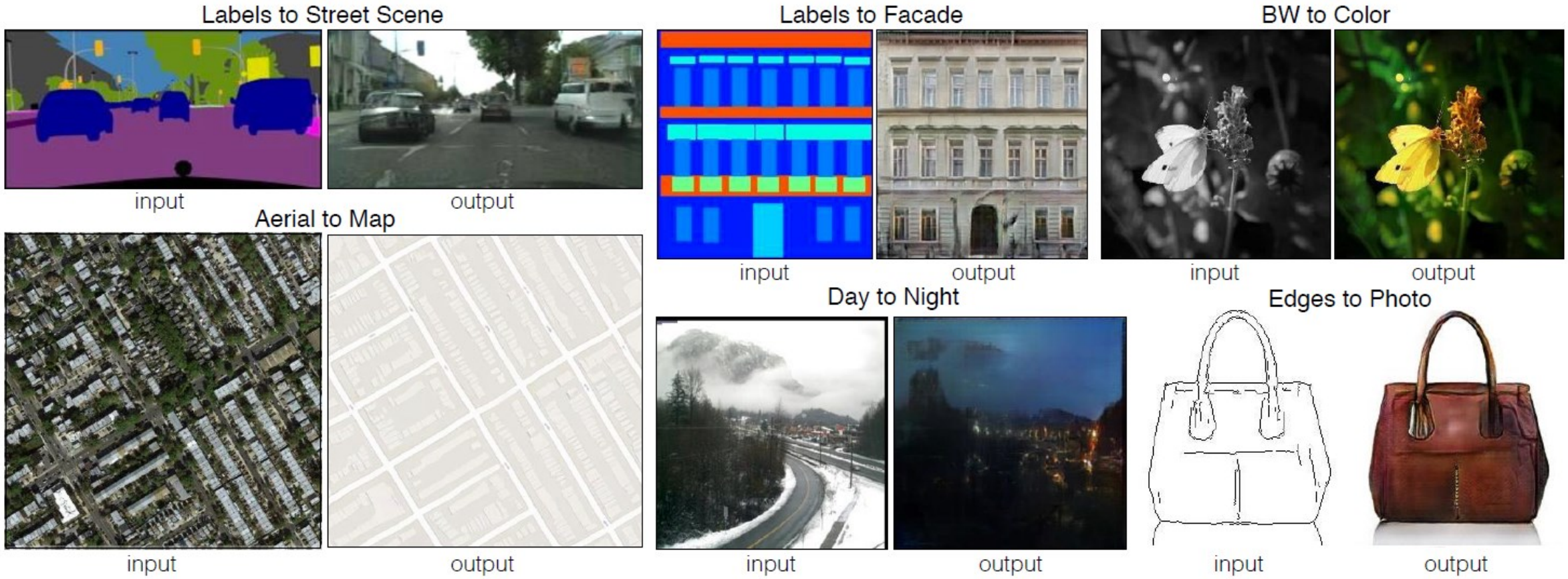
Text to Image Generation

"StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks "

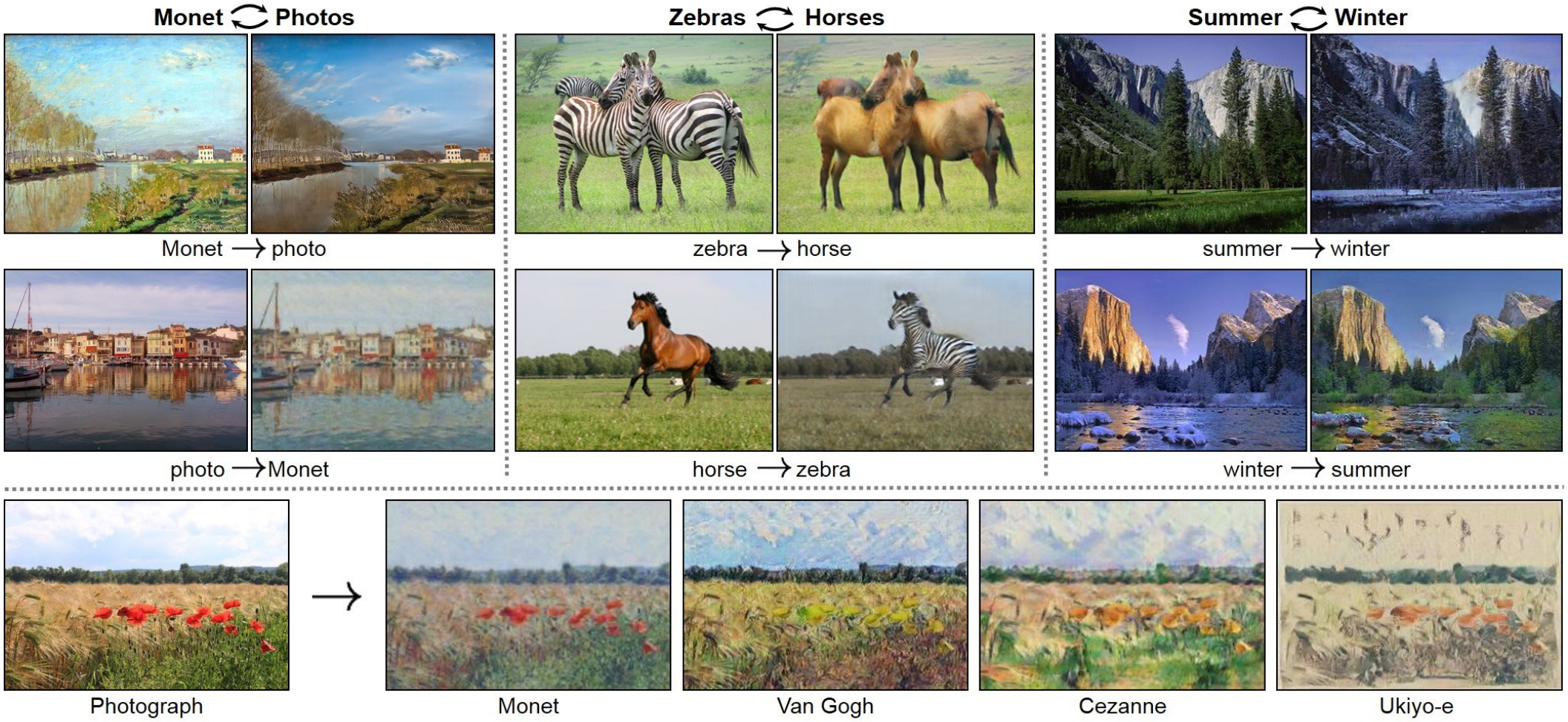
Domain Transfer

Pix2Pix

- https://affinelayer.com/pixsrv/



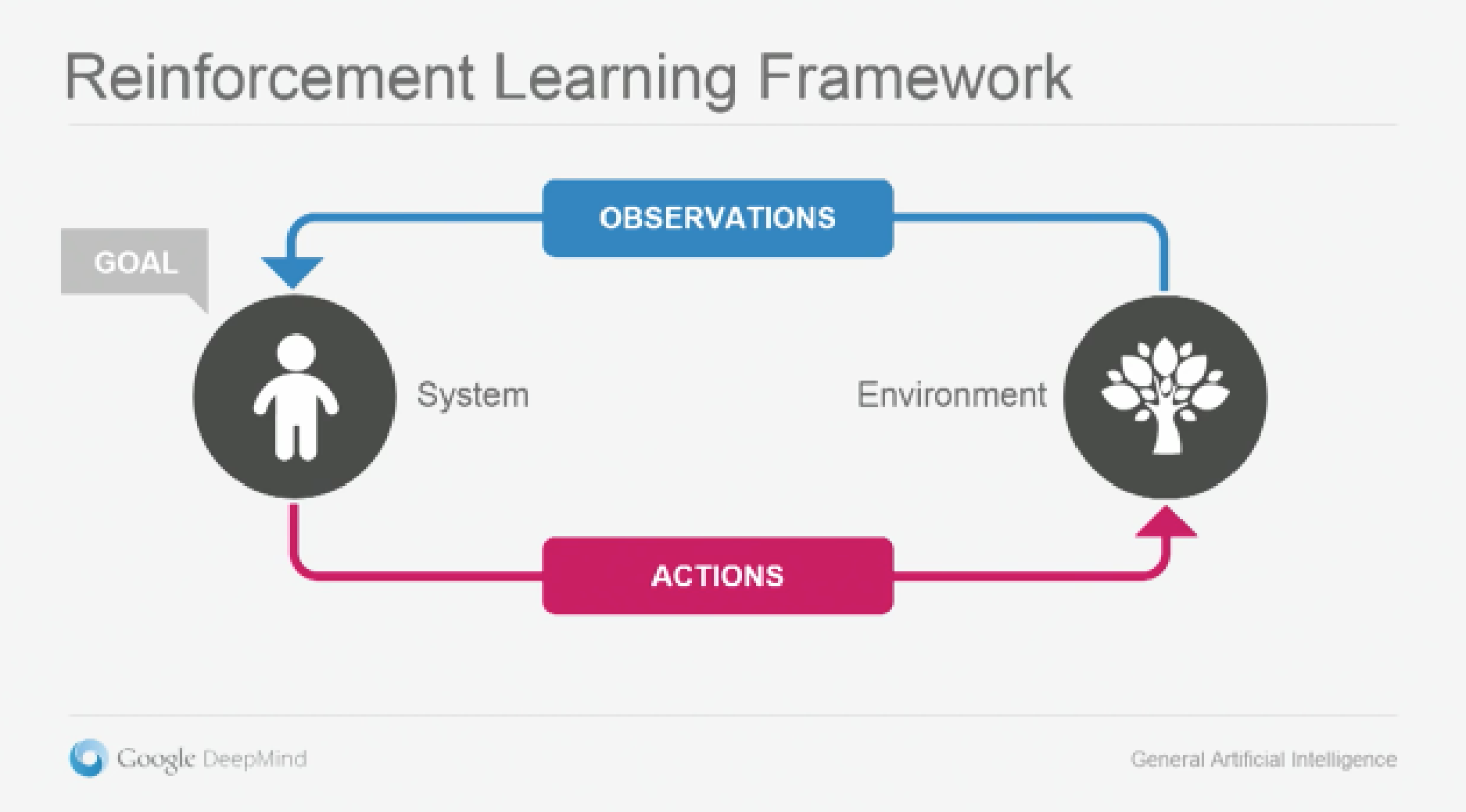
CycleGAN


Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

Video-to-Video Synthesis

Video Generation

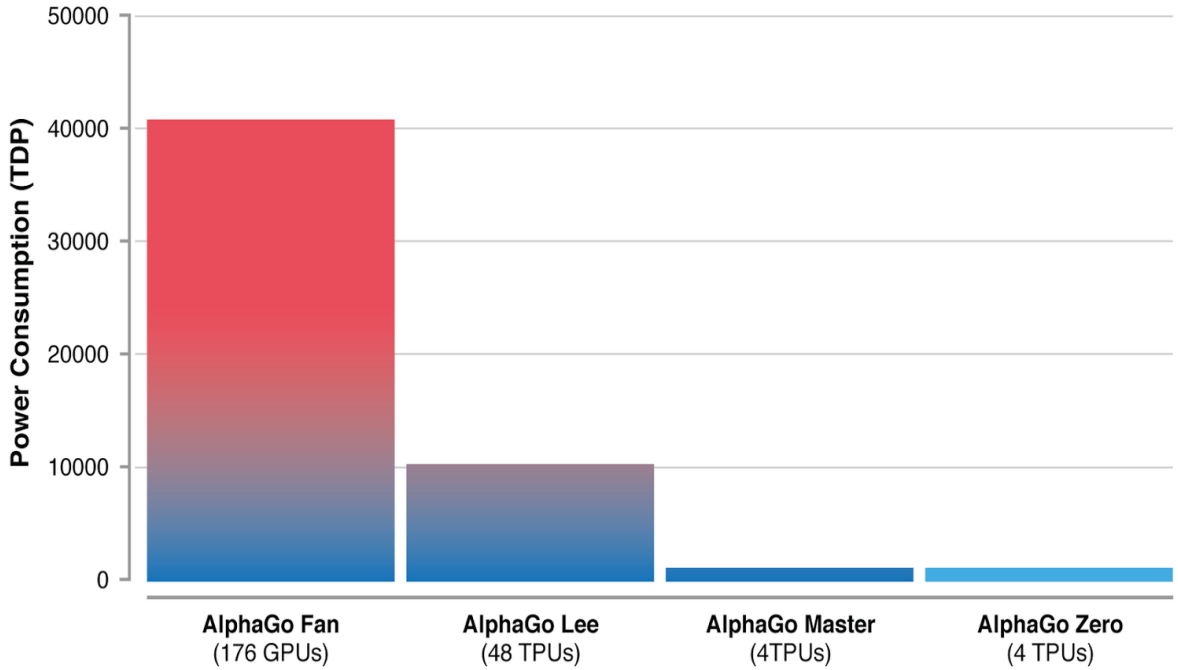
Reinforcement Learning

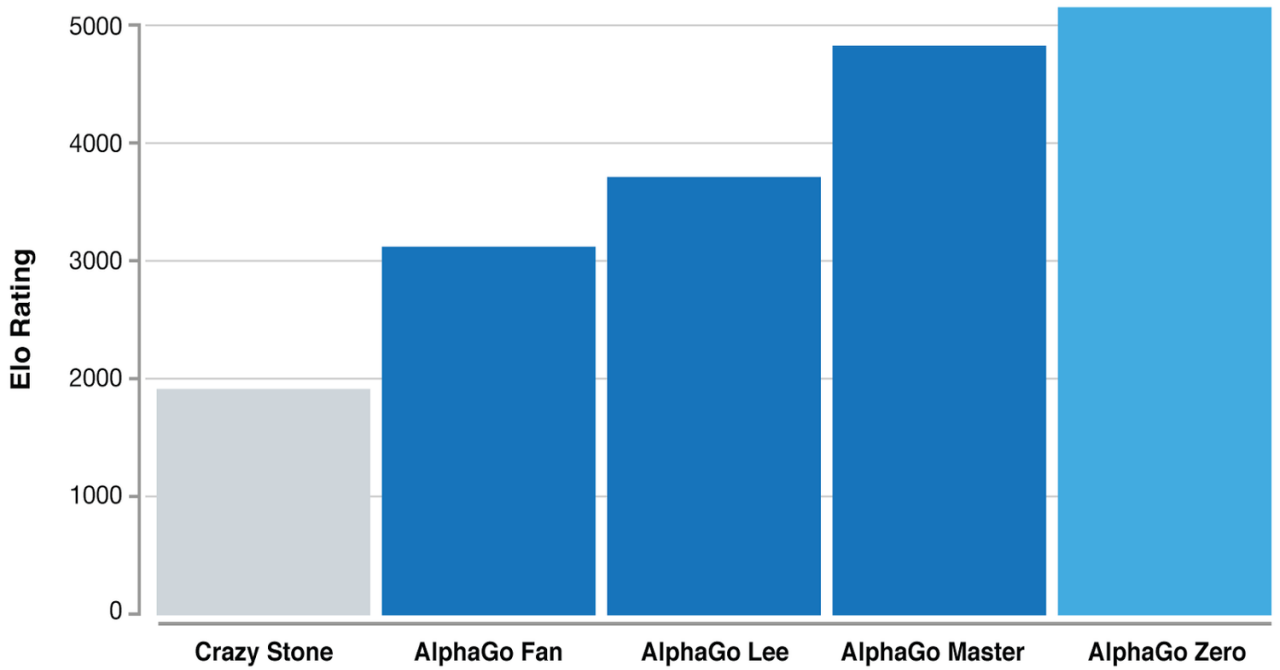


Reinforcement Learning

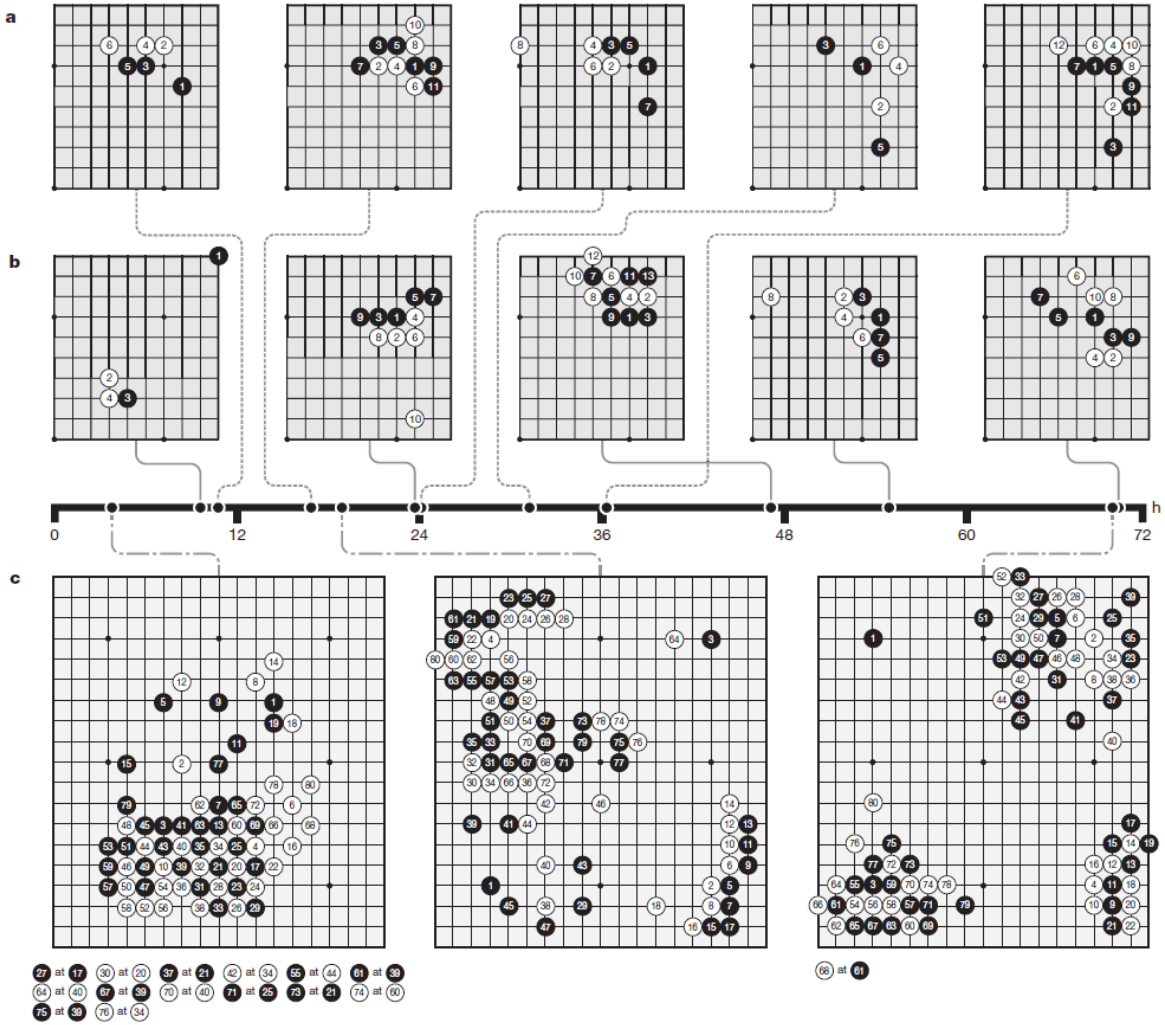
AlphaGo Zero

- Tabula rasa
- Without handcrafted input feature
- Rollout is removed





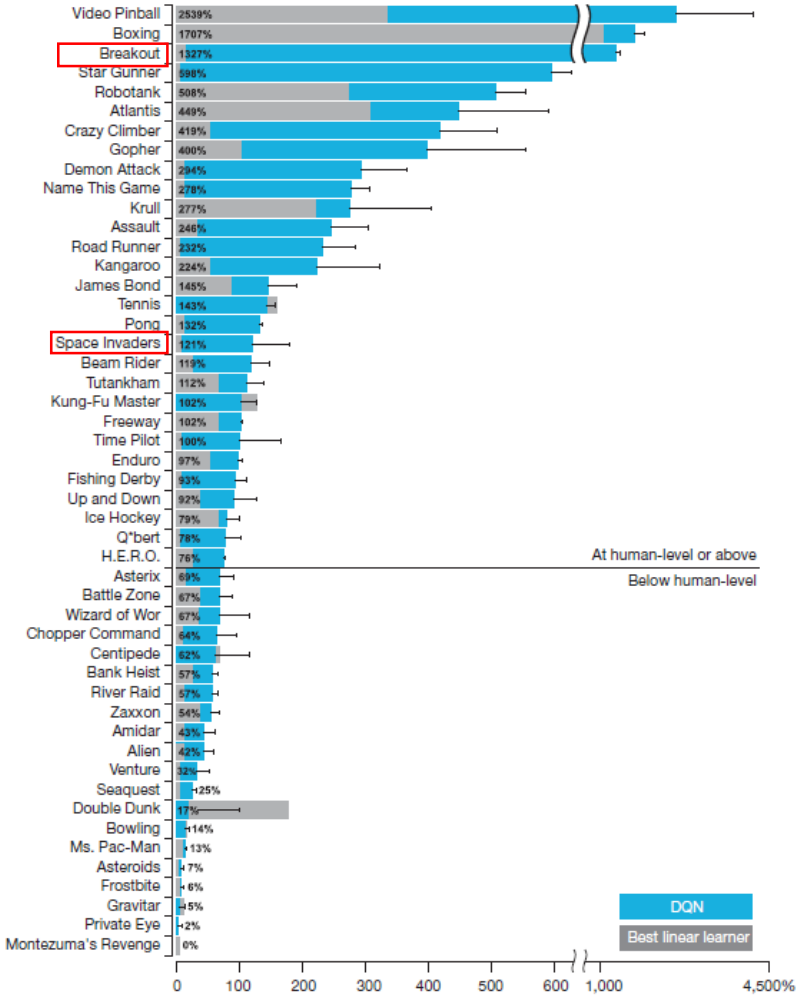
AlphaGo Zero



Atari Games

Space Invaders

https://youtu.be/SuoouUnjDo



Other Games

- Super Mario World
  - https://youtu.be/L4KBBAwF_bE
  - https://youtu.be/qv6UVOQ0F44
- Cookie Run
  - https://youtu.be/exXD6wJLJ6s
- Starcraft I
  - https://youtu.be/GgkmJDjeJtw
- GTA
  - https://youtu.be/X4u2DCOLoIg

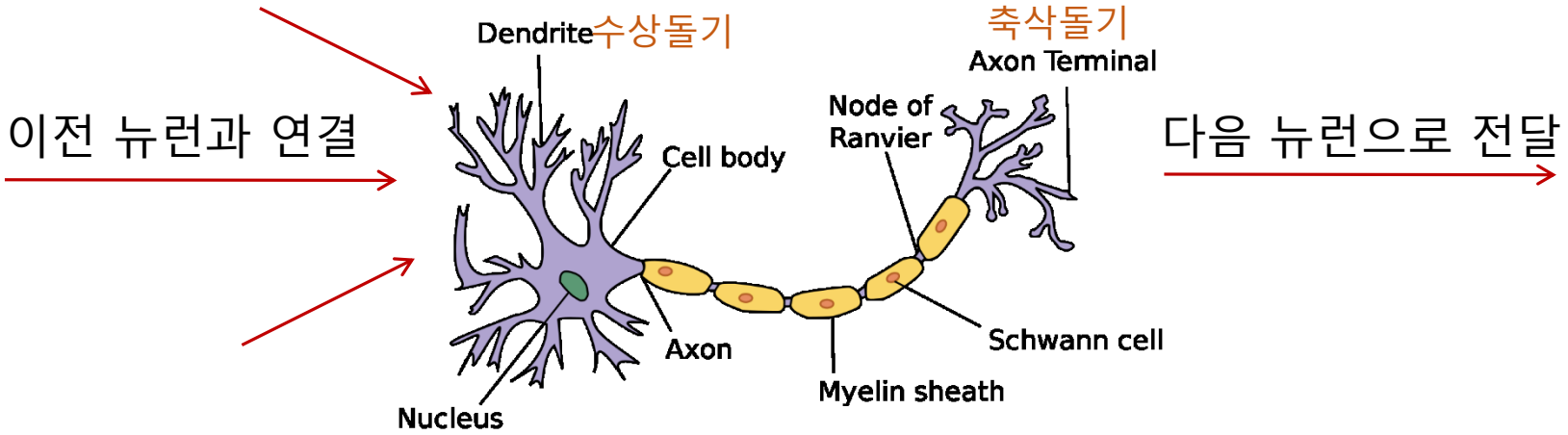
Autonomous Driving

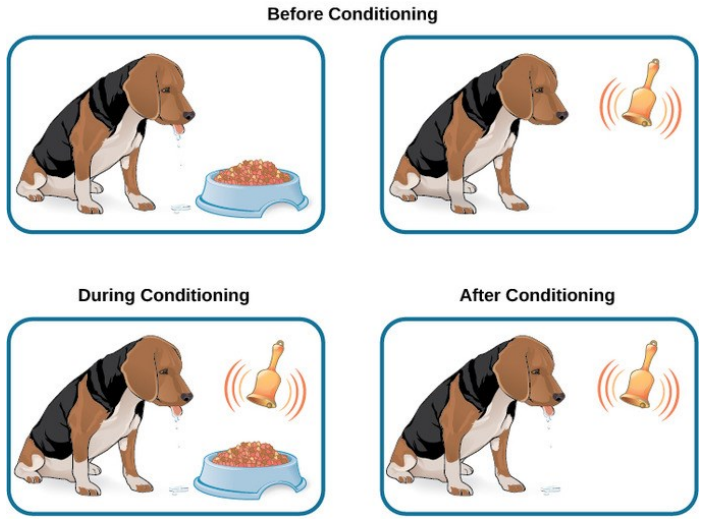
Quiz

- □와 △에 들어갈 정수는?
  - 3 x □ + 2 x △ = 1
  - 1 x □ + 4 x △ = -3
  - 5 x □ + 5 x △ = 0
  - 8 x □ + 3 x △ = 5
- **□ = 1, △ = -1**

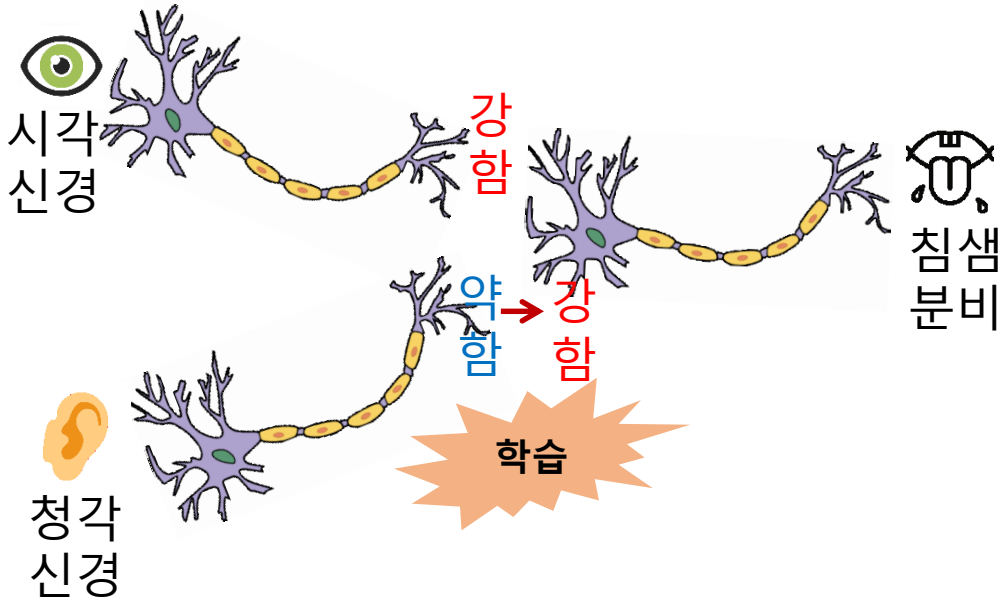
- (3, 2), (1, 4), (5, 5), (8, 3) 은 input data, 1, -3, 0, 5 는 label이다
- □와 △를 weight라고 하며 이 weight 값을 기계가 스스로 학습을 통해 찾아내도록 하는 것이 neural network를 이용한 기계학습이 하는 일

뉴런과 사람의 학습

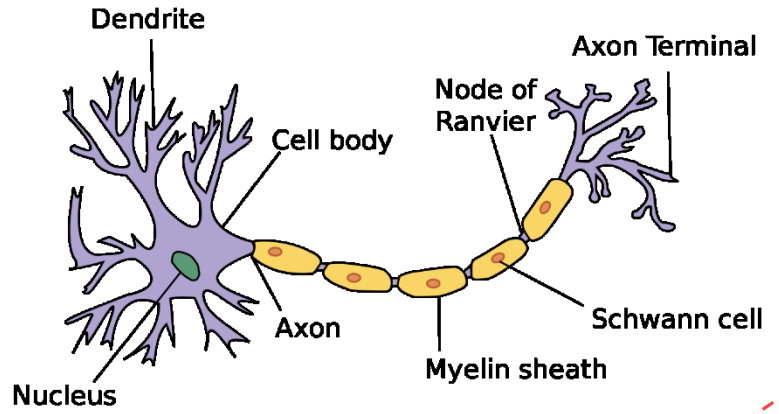


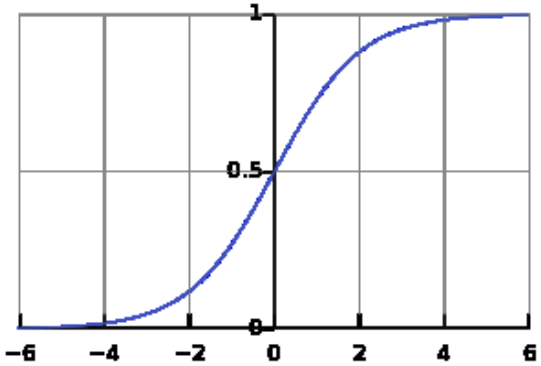


파블로프의 개 실험



Perceptron(Artificial Neural Network)





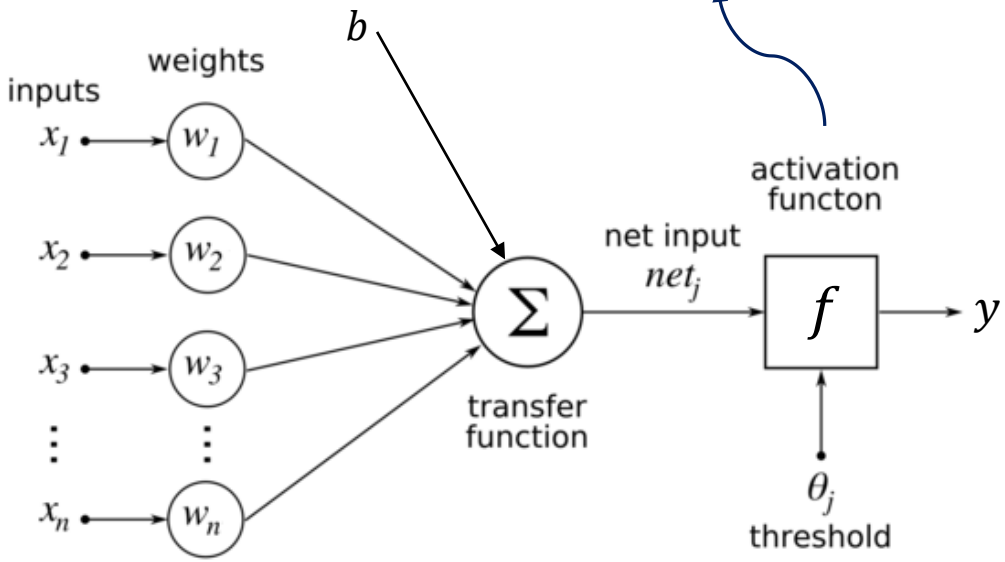
sigmoid activation function

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

$$y = f(\mathbf{wx} + \mathrm{b})$$

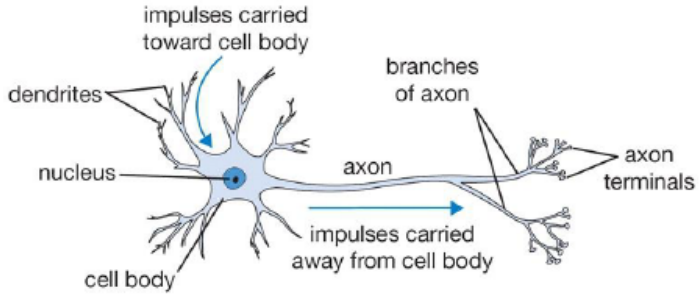
$$\boldsymbol{w} = [w_1\ w_2\ w_3\ \ldots\ w_n]$$

$$\boldsymbol{x} = [x_1\ x_2\ x_3\ \ldots\ x_n]^T$$

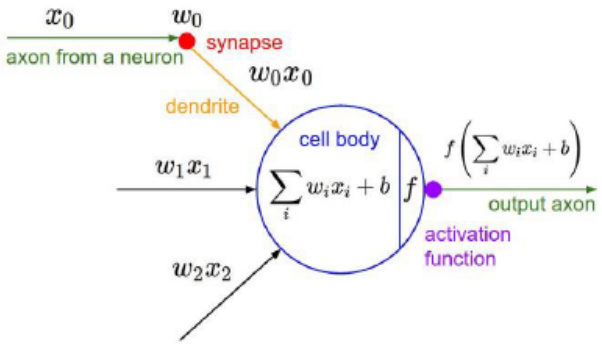


<Perceptron>

Neuron vs Perceptron



- **Neuron:** computational building block for the brain

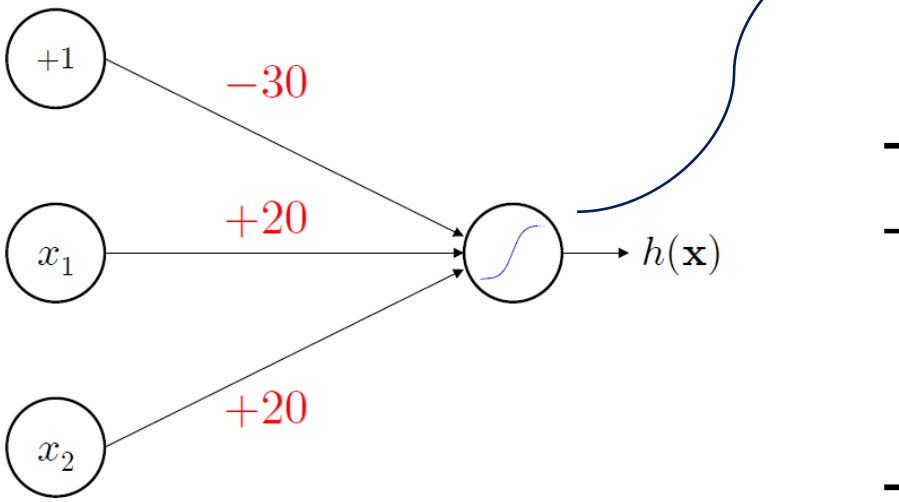


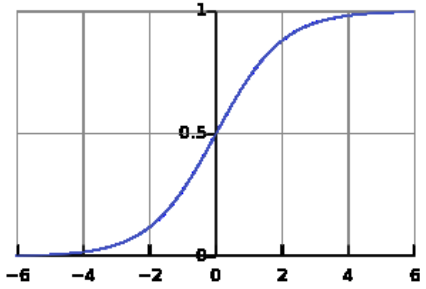
- **(Artificial) Neuron:** computational building block for the "neural network"

Key Difference:

- **Parameters:** Human brains have ~10,000,000 times synapses than artificial neural networks.
- **Topology:** Human brains have no "layers". **Async:** The human brain works asynchronously, ANNs work synchronously.
- **Learning algorithm**: ANNs use gradient descent for learning. We don't know what human brains use
- **Power consumption:** Biological neural networks use very little power compared to artificial networks
- **Stages:** Biological networks usually never stop learning. ANNs first train then test.

Example of ANN(logical AND)

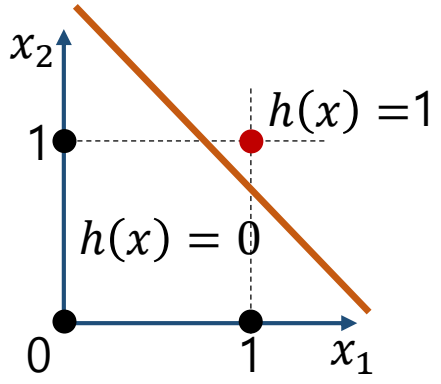




x_1	x_2	$h(\mathbf{x})$
0	0	$\sigma(-30) \approx 0$
0	1	$\sigma(-10) \approx 0$
1	0	$\sigma(-10) \approx 0$
1	1	$\sigma(10) \approx 1$

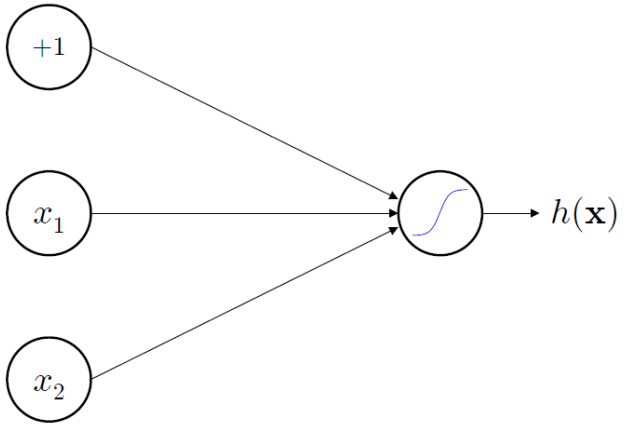
$$h(\mathbf{x}) = \sigma(-30 + 20x_1 + 20x_2)$$

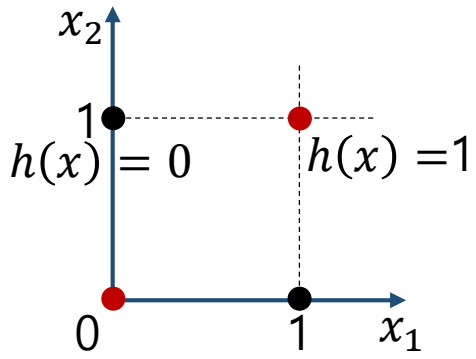
학습이란 이러한 weight 값(-30, 20, 20)을
기계 스스로 찾을 수 있도록 해주는 과정!



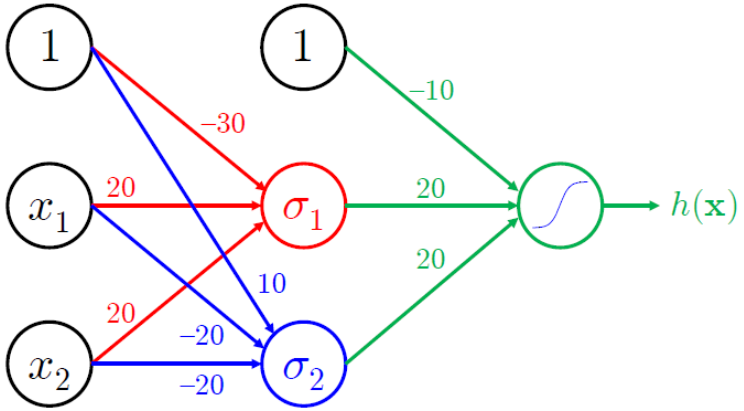
Logical XNOR with Perceptron

- Is it possible(Linearly separable)?



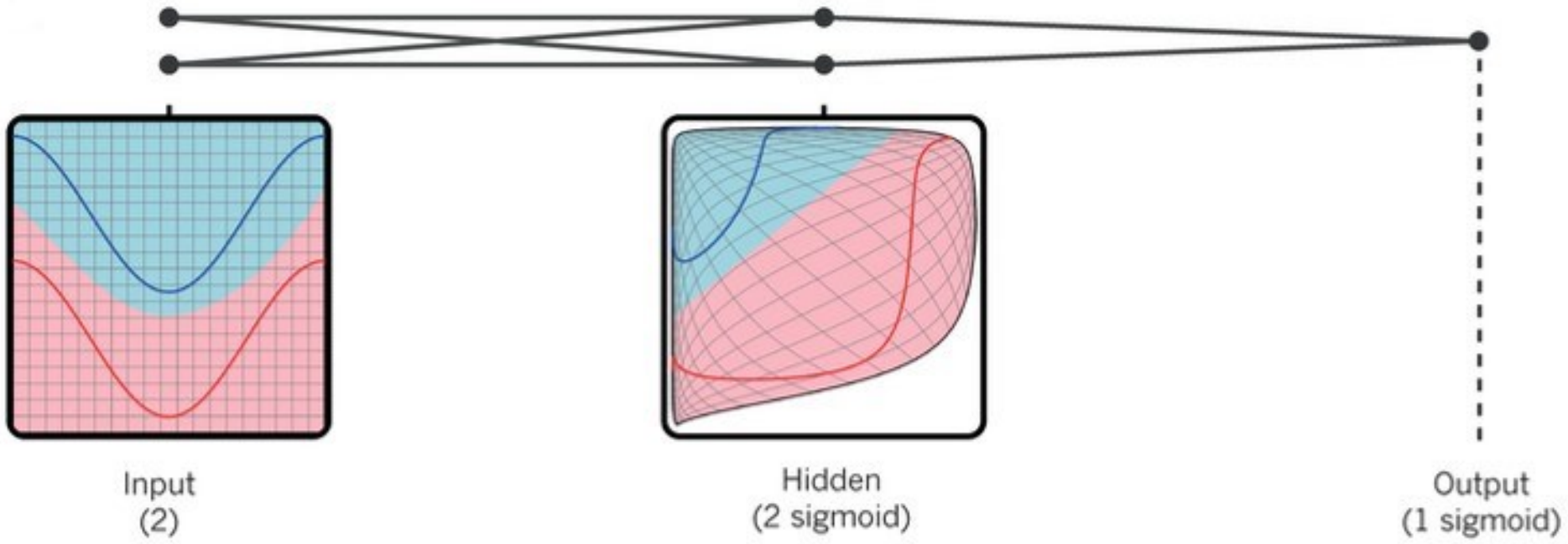


- We need more perceptrons and more layers



x_1	x_2	σ_1	σ_2	$h(\mathbf{x})$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

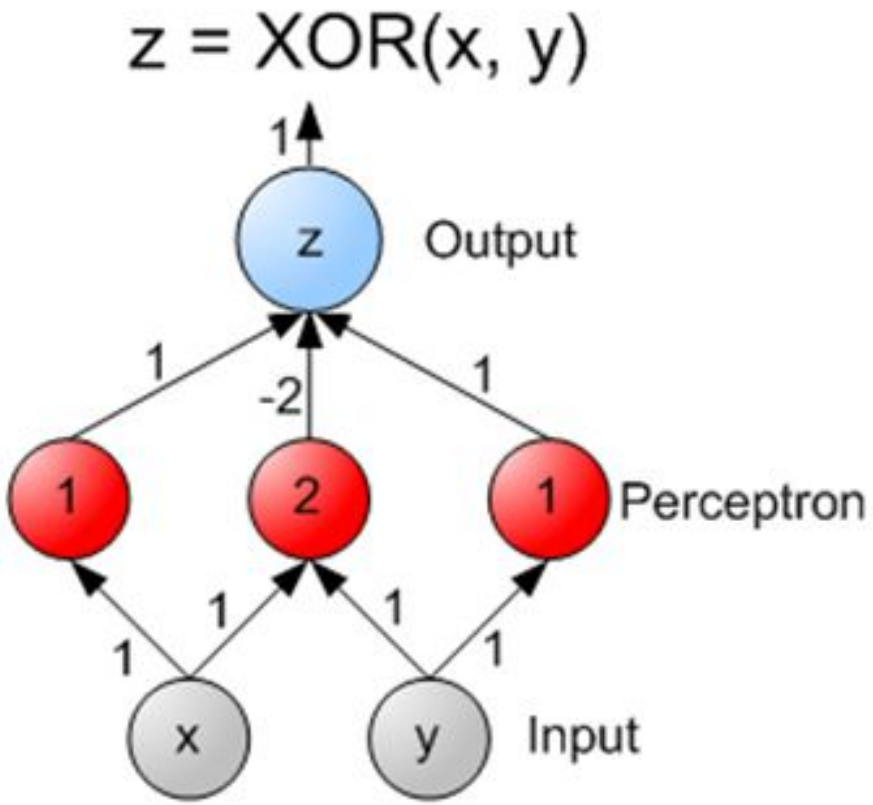
Multi Layer Perceptron(MLP)



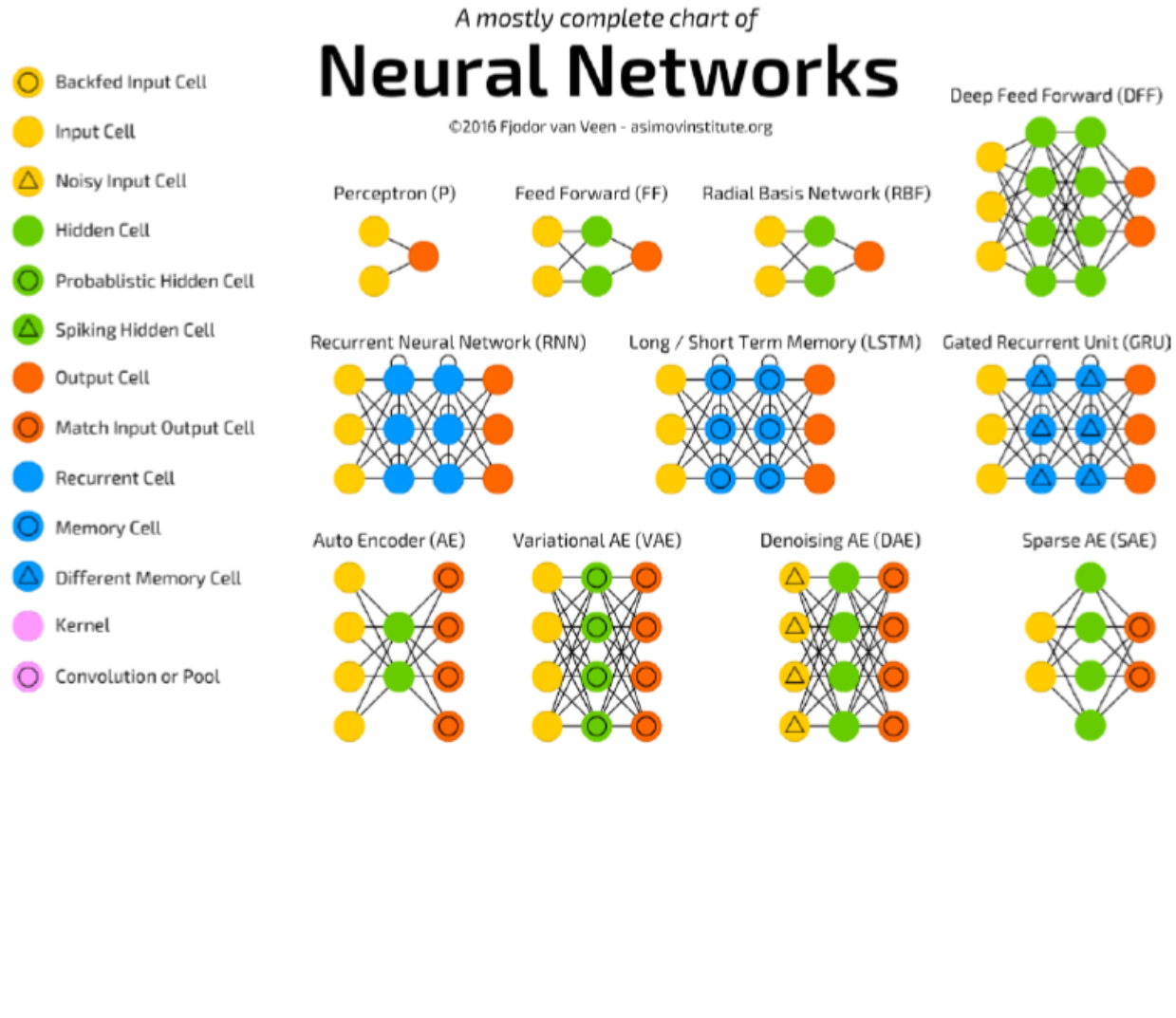
- 선을 잘 긋고 input 공간을 잘 왜곡하고 합하는 과정을 반복해서 데이터들을 잘 구분해보자(classification)
- 이렇게 perceptron을 여러층으로 쌓으면 더 복잡한 문제를 풀 수 있다
- Linear fitting과 Non-linear transform의 반복

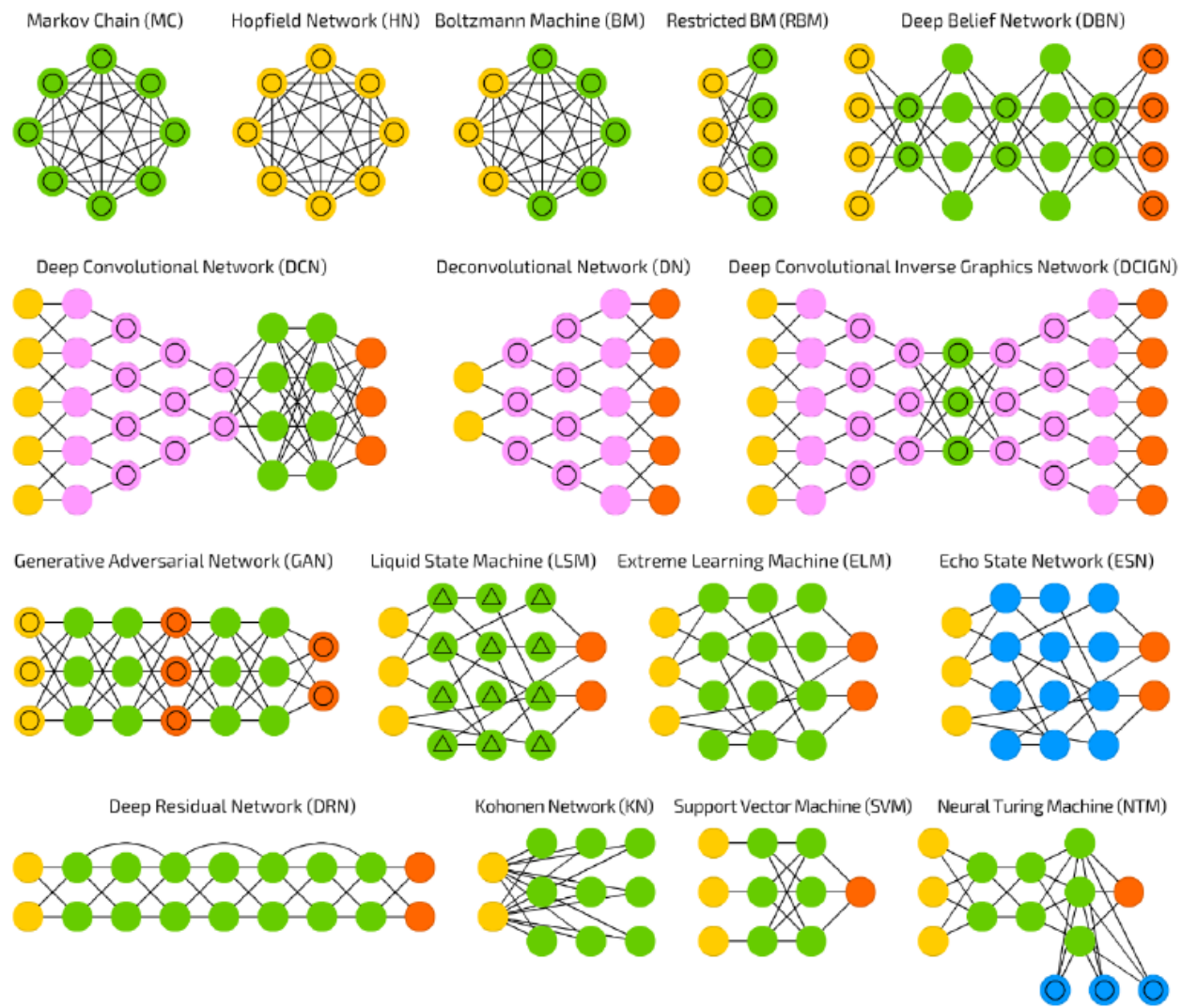
Universal Function Approximation

The *universal approximation theorem* for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer. This result holds only for restricted classes of activation functions, e.g. for the sigmoidal functions. (Wikipedia.org)



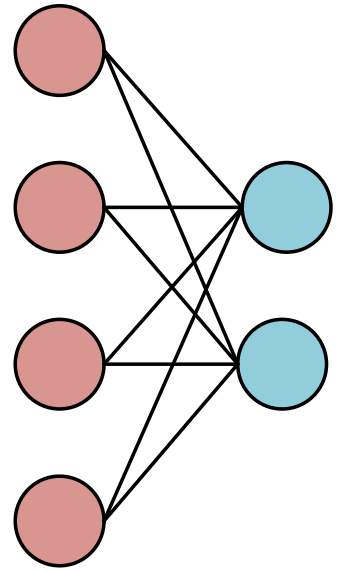
Making Neural Networks





Single-Layer Perceptron & Multi-Layer Perceptron(MLP)

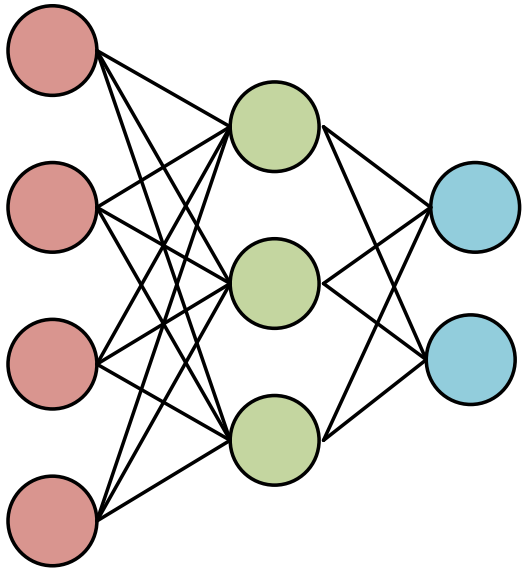
Single-Layer Perceptron



input layer

output layer

Multi-Layer Perceptron



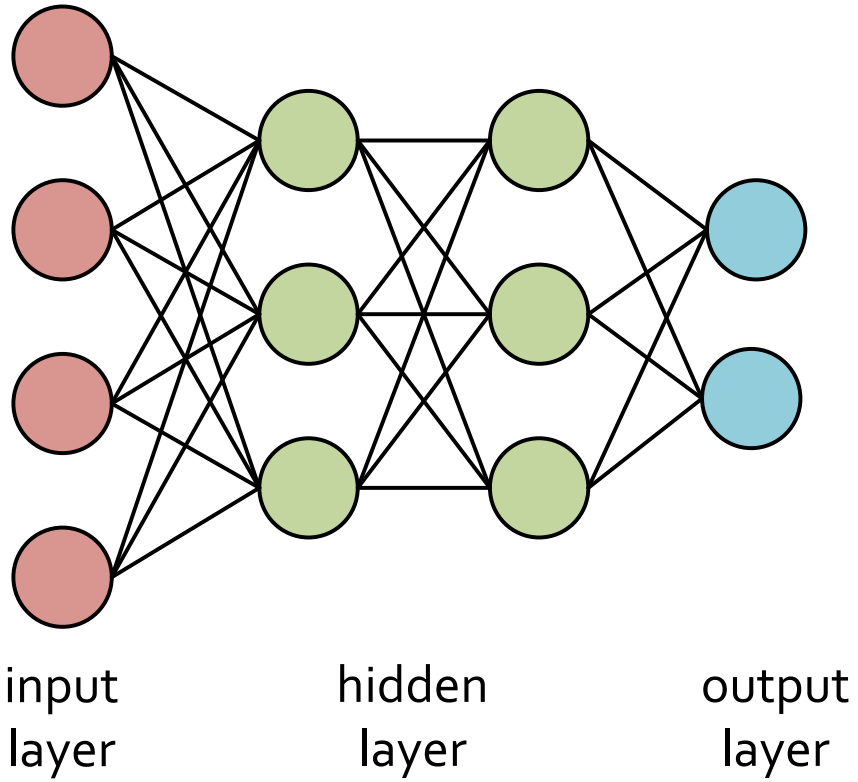
input layer

hidden layer

output layer

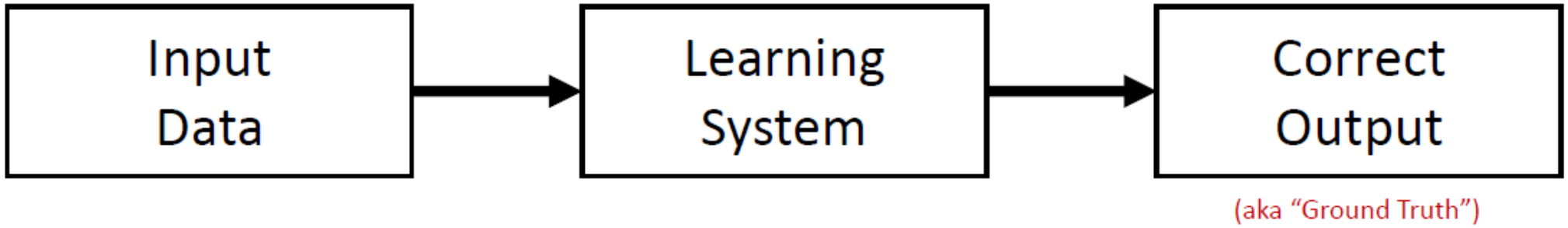
Deep Learning

- Deep Neural Network을 이용한 Machine Learning 방법
  - Deep Neural Network : Hidden layer 수가 2개 이상인 network

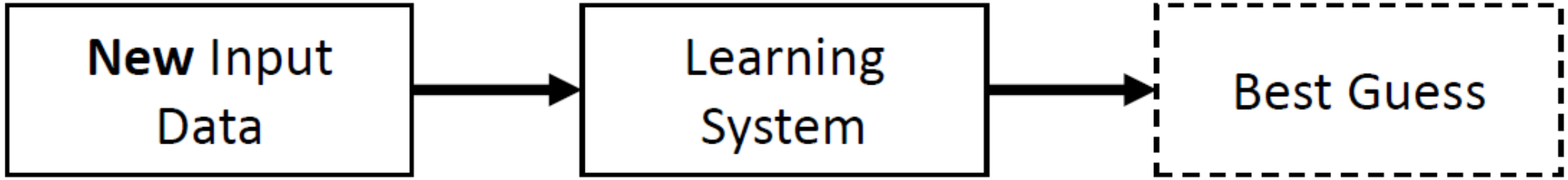


Training and Testing

Training Stage:



Testing Stage:

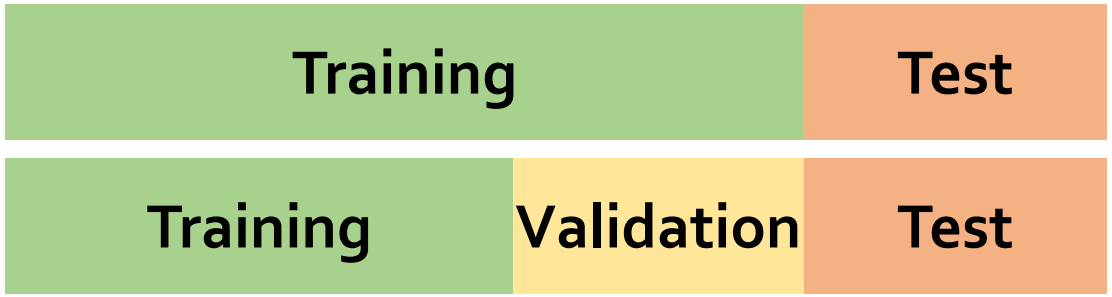


Dataset

- Training set, Test set

  or

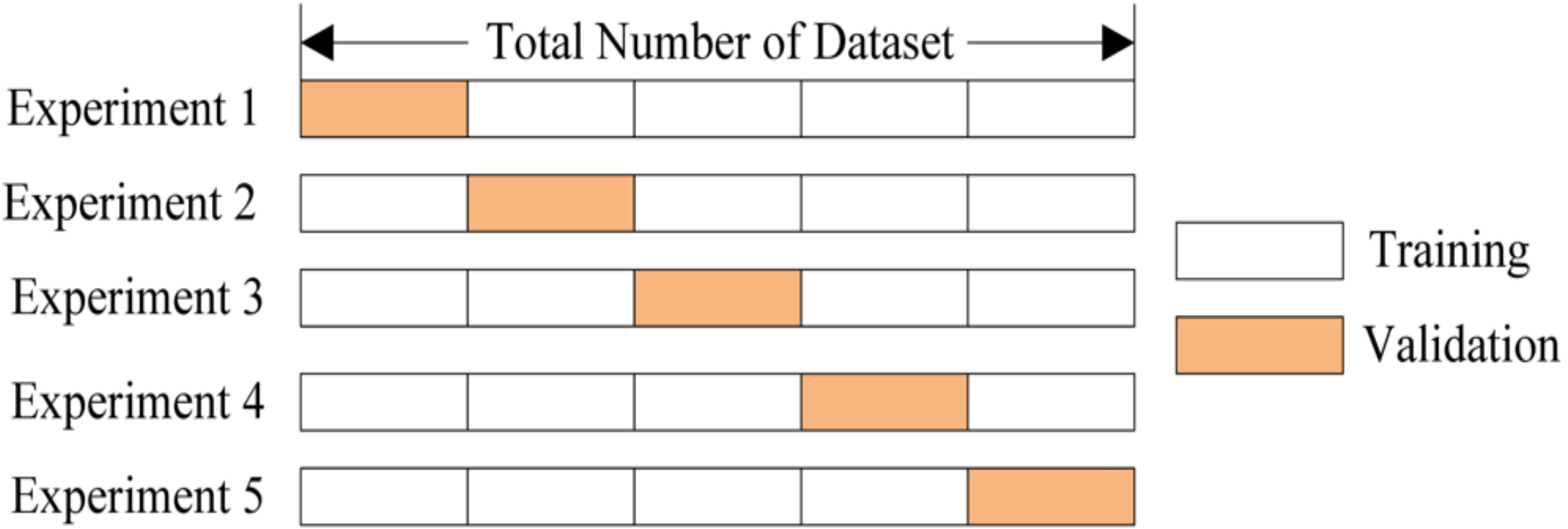
- Training set, Validation set, Test set



- Training → Tuning(validation) → Test

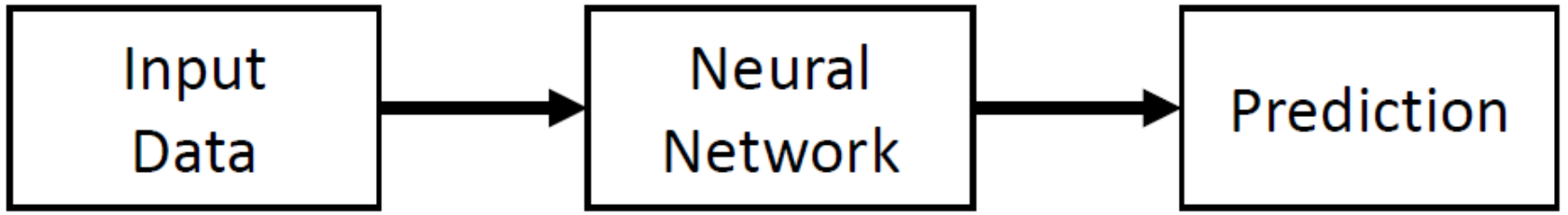
- In any case, DO NOT USE the test set before the training is over!

Cross-Validation

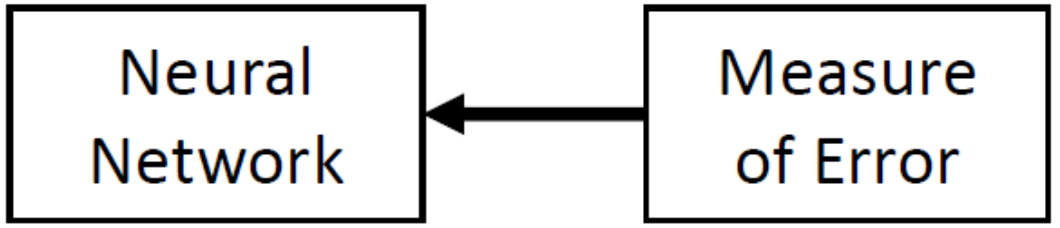


How Neural Networks Learn

Forward Pass:

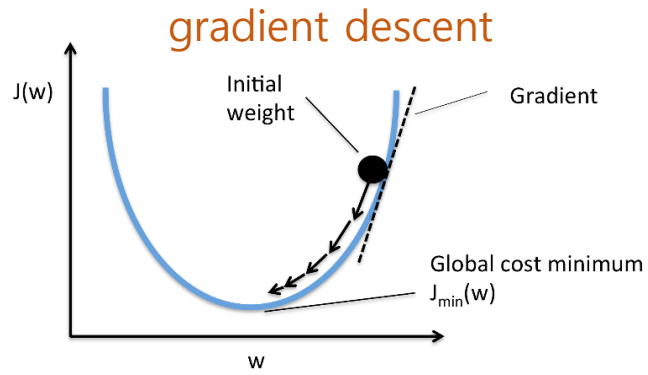


Backward Pass (aka Backpropagation):

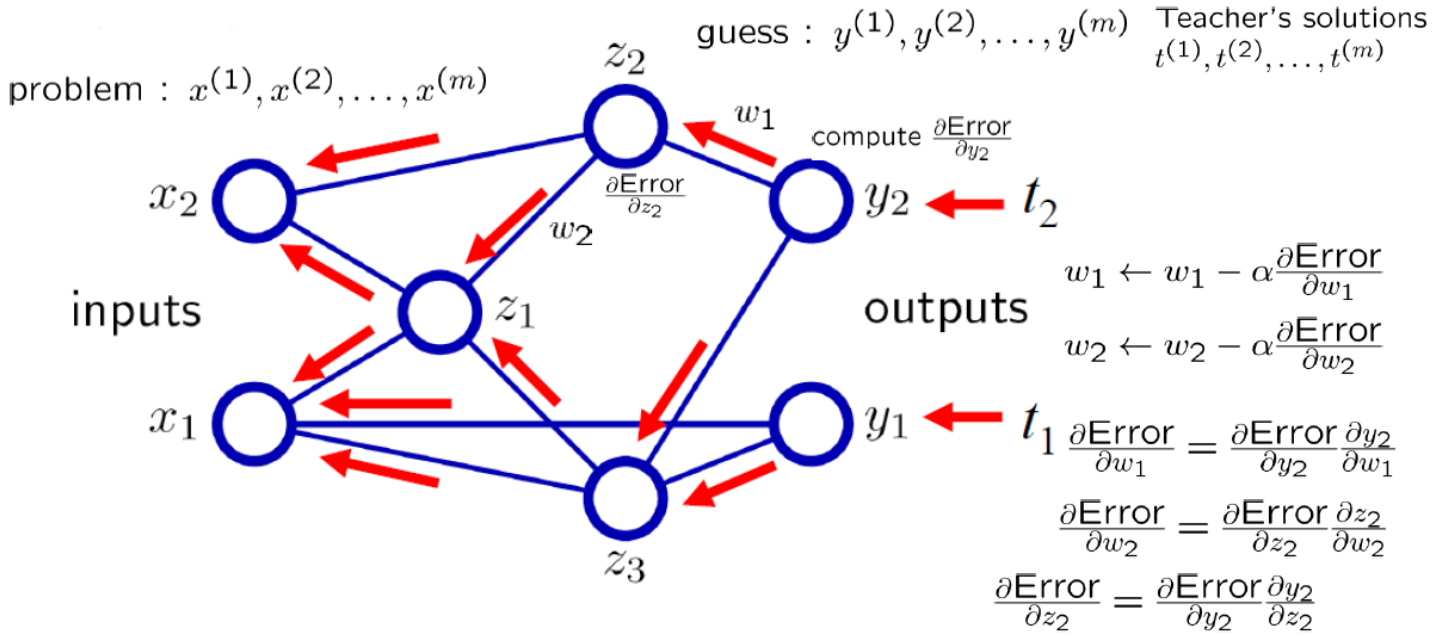


Back Propagation

- 학습과정 : back propagation of error
  - Output layer에서 error(cost)를 계산
  - Error의 미분값을 back propagation
  - 미분값에 α를 곱한 만큼 w를 보정(학습!)
  - α는 learning rate를 의미함



$$\text{minimize Error} = \sum_{l=1}^{m} (y^{(l)} - t^{(l)})^2$$



$$w_1 \leftarrow w_1 - \alpha \frac{\partial \text{Error}}{\partial w_1}$$

$$w_2 \leftarrow w_2 - \alpha \frac{\partial \text{Error}}{\partial w_2}$$

$$\frac{\partial \text{Error}}{\partial w_1} = \frac{\partial \text{Error}}{\partial y_2} \frac{\partial y_2}{\partial w_1}$$

$$\frac{\partial \text{Error}}{\partial w_2} = \frac{\partial \text{Error}}{\partial z_2} \frac{\partial z_2}{\partial w_2}$$

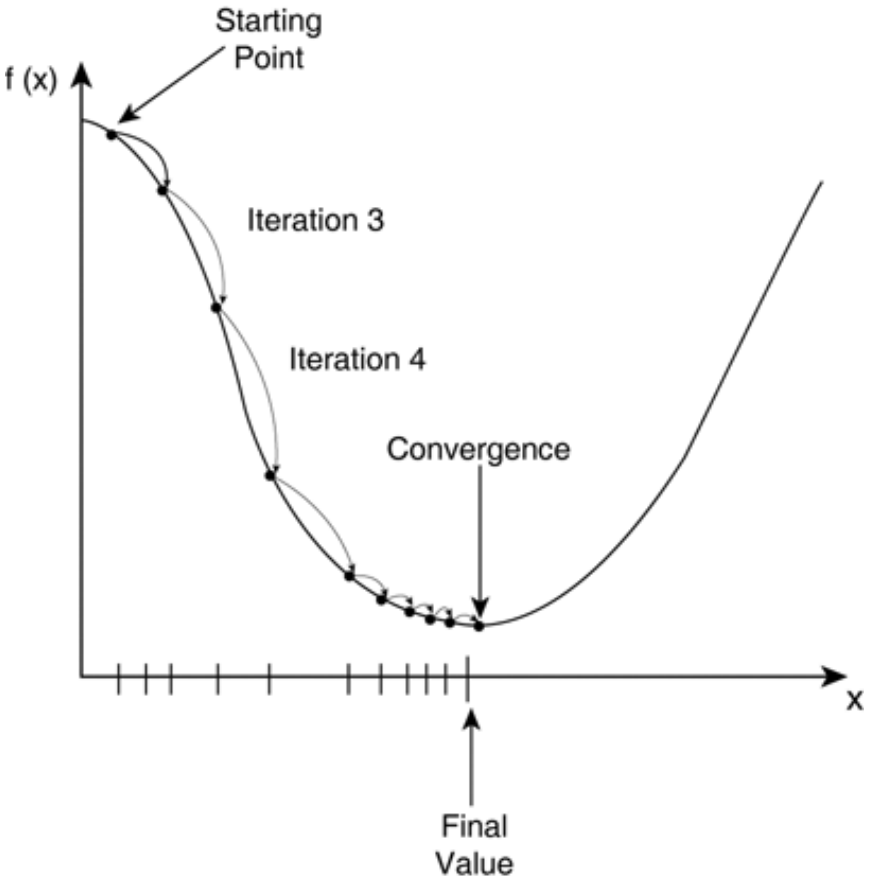
$$\frac{\partial \text{Error}}{\partial z_2} = \frac{\partial \text{Error}}{\partial y_2} \frac{\partial y_2}{\partial z_2}$$

Gradient Descent

- Loss Function 을 W(parameter)로 편미분해서 W에 대한 Gradient를 구한다

- Gradient를 이용해서 W를 업데이트 한다

$$W^* = W - \alpha \text{Loss}'(W)$$

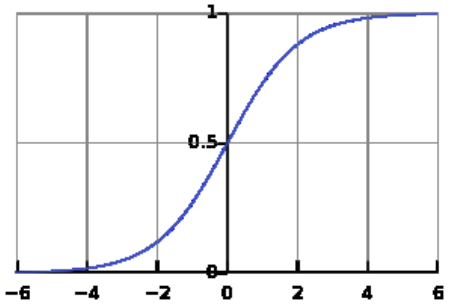


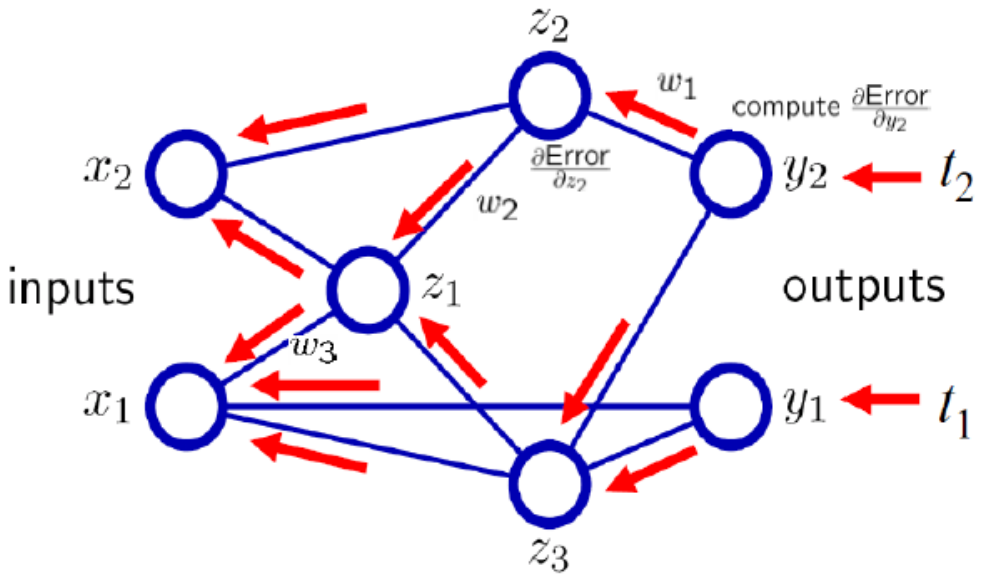
딥러닝을 어렵게 하는 것들

- Vanishing gradient problem
- Overfitting problem
- Get stuck in local minima

Vanishing Gradient Problem

- Gradient 값이 뒤로 전달될 수록 점점 작아짐
- Sigmoid 사용으로 인하여(미분값의 최대 : ¼) 아래쪽 layer 는 학습이 이루어지지 않음





$$\frac{\partial \text{Error}}{\partial w_1} = \frac{\partial \text{Error}}{\partial y_2}\frac{\partial y_2}{\partial w_1} = \frac{\partial \text{Error}}{\partial y_2}\sigma'(\cdot)z_2$$

$$\frac{\partial y_2}{\partial w_1} = \sigma'(\cdot)z_2$$

$$\frac{\partial \text{Error}}{\partial w_2} = \frac{\partial \text{Error}}{\partial z_2}\frac{\partial z_2}{\partial w_2} = \frac{\partial \text{Error}}{\partial y_2}\sigma'(\cdot)w_1\sigma'(\cdot)z_1$$

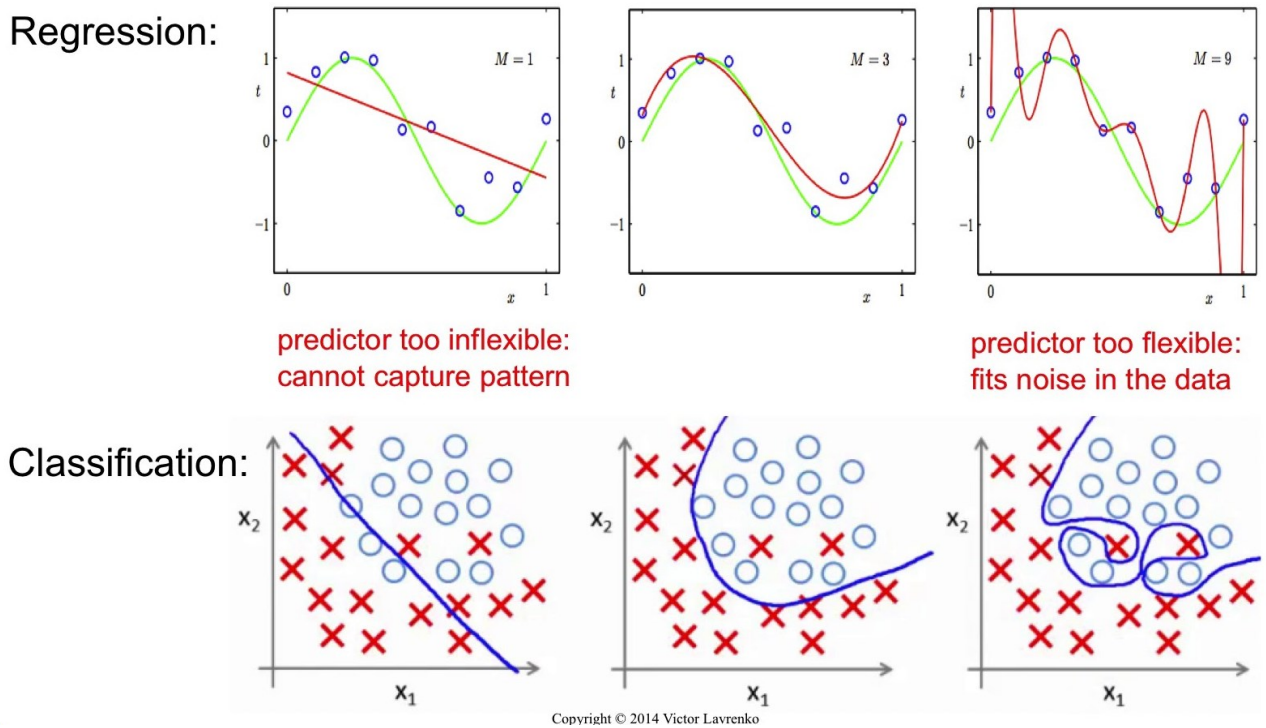
$$\frac{\partial z_2}{\partial w_2} = \sigma'(\cdot)z_1$$

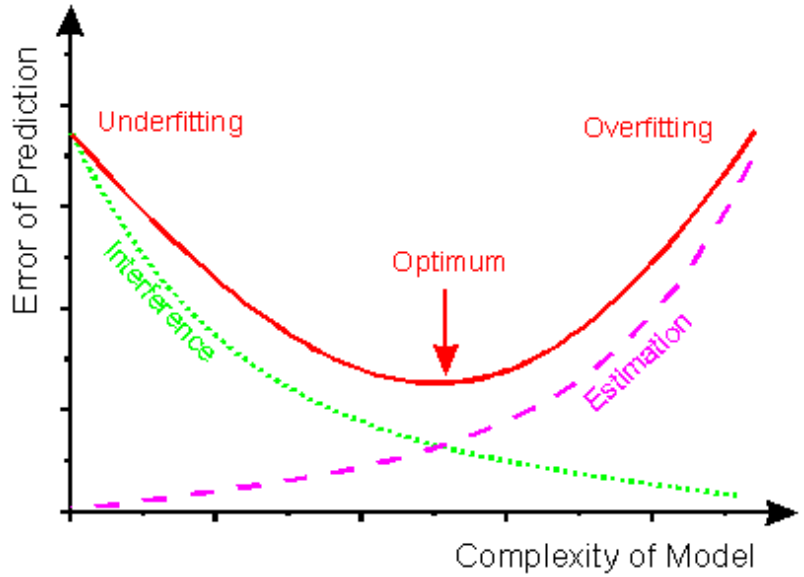
$$\frac{\partial \text{Error}}{\partial w_3} = \sigma'(\cdot)\sigma'(\cdot)\sigma'(\cdot)C$$

$$\sigma'(\cdot) \approx 0$$

Overfitting Problem

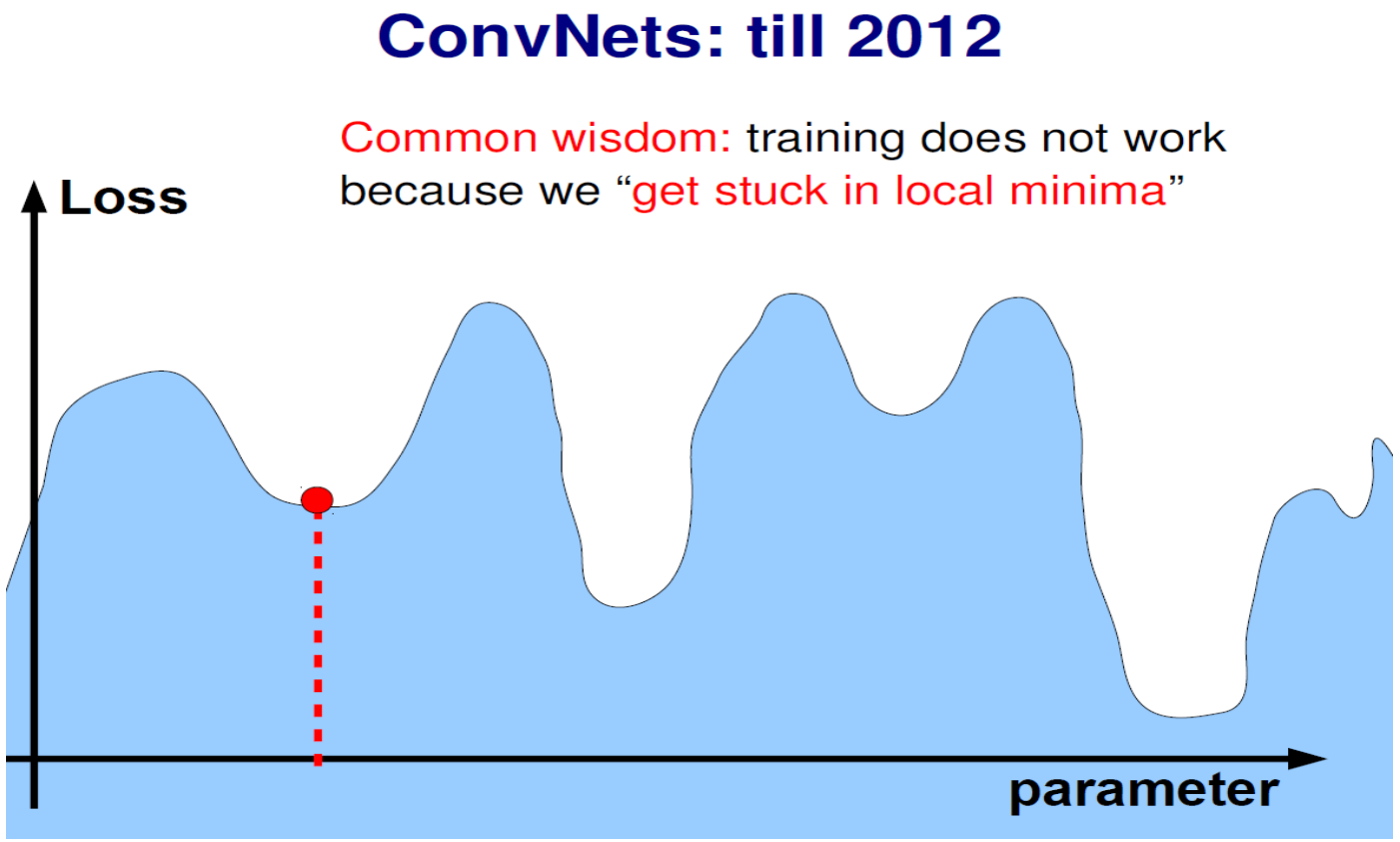
- Data가 많지 않은 경우에 발생할 수 있음
- 학습한 data에만 최적화되어서, 학습하지 않은 data(test data)에 대한 추론 성능이 악화되는 현상





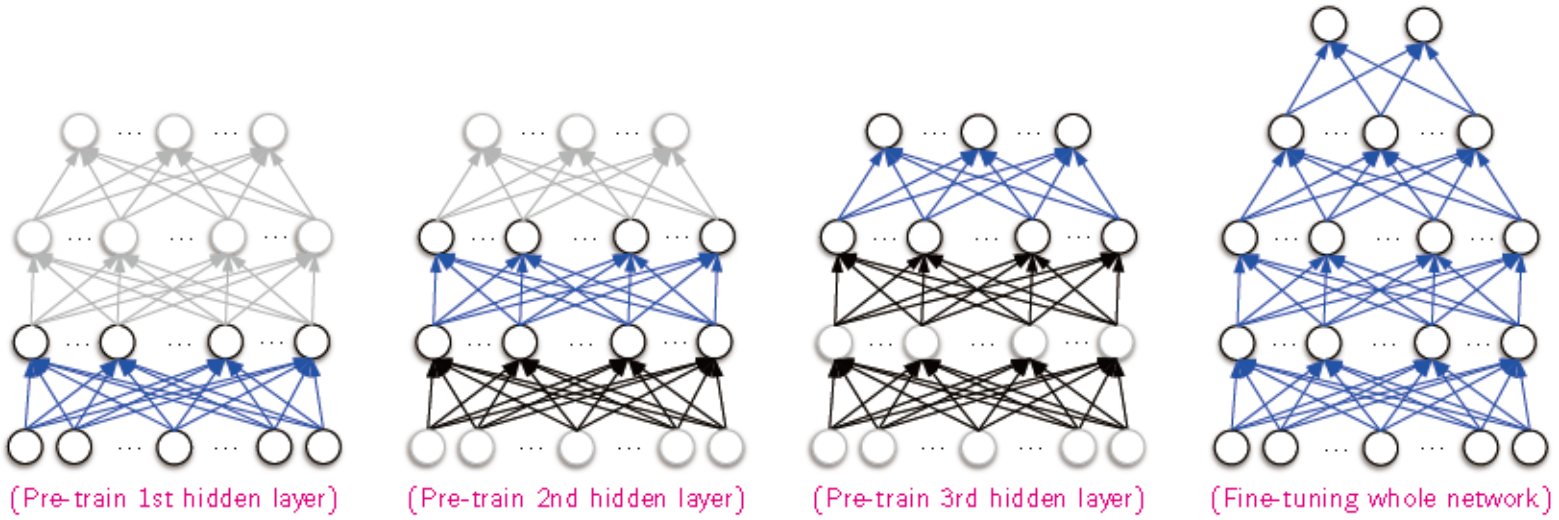
Local Minima

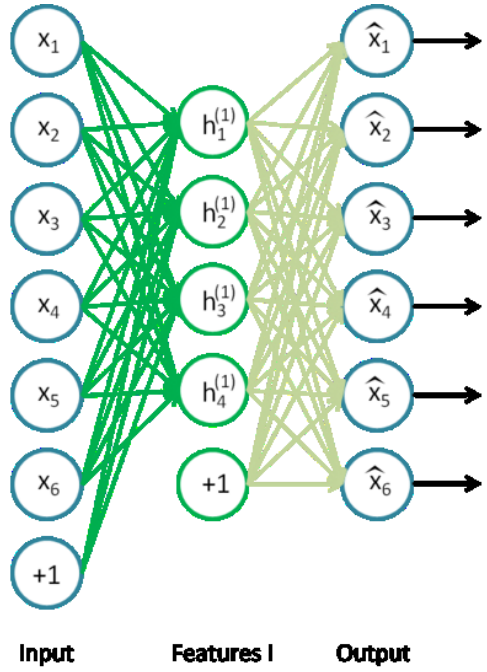
- 어디서 시작하느냐에 따라서 잘못하면 local minima에 빠질 위험이 존재



Deep Belief Network

- Probabilistic generative model
- Deep architecture – multiple layers (stacks of RBMs)
- Greedy layer-wise training algorithm
- Supervised fine-tuning can be applied





Difficulties of Training DNN

- Vanishing gradient problem

  → Solved by bottom-up layerwise unsupervised pre-training

- Typically requires lots of labeled data
- Overfitting problem

  → Solved by using lots of unlabeled data

- Get stuck in local minima(?)

  → Unsupervised pre-training may help the network initialize with good parameters(?)

새로운 방법들

- Vanishing gradient problem

→ Sigmoid 말고 ReLU를 쓰자

- Overfitting problem

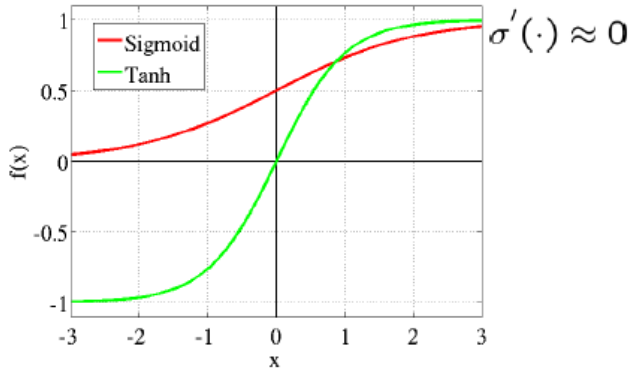
→ Regularization method를 쓰자(예 : dropout)

- Get stuck in local minima

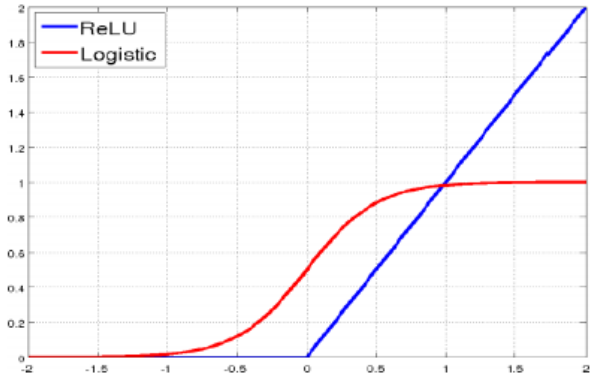
→ Local minima에 빠져도 괜찮다

ReLU : Rectified Linear Unit

- ReLU를 activation function으로 사용 → sparse activation
- ReLU는 미분값이 0 아니면 1 → vanishing gradient 해결

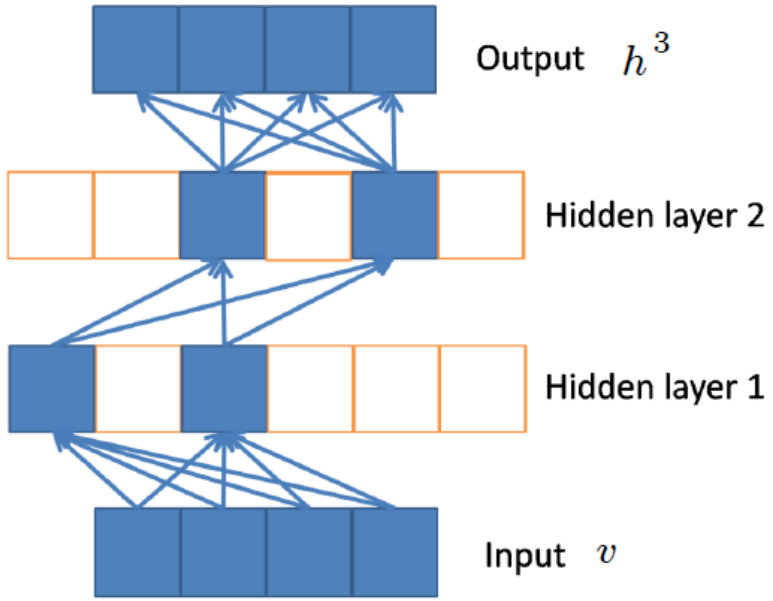


$\sigma'(\cdot) \approx 0$



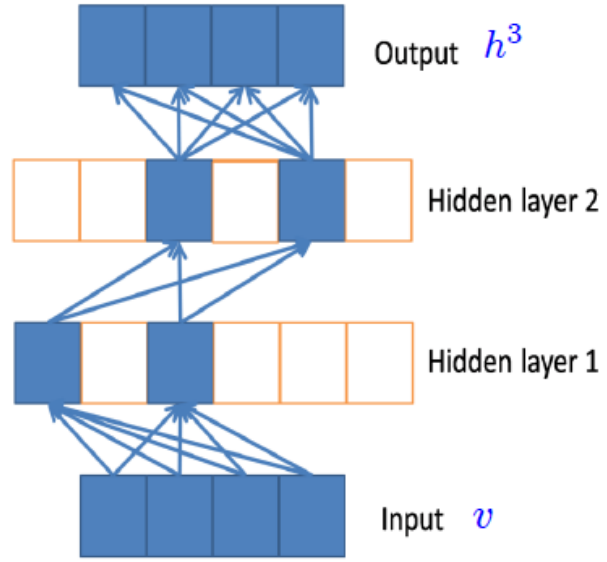
$$f(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

slope: 1



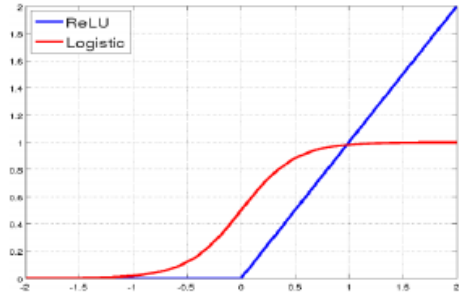
ReLU의 의미

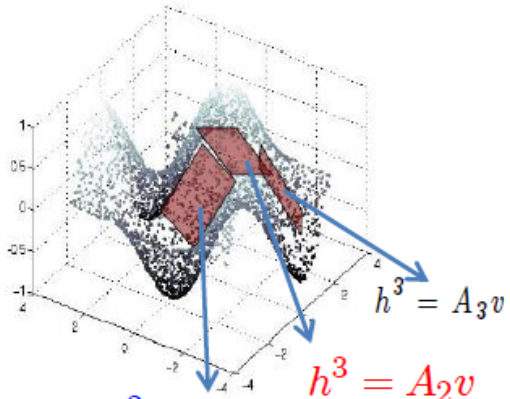
- Piece-wise linear tiling : locally linear mapping

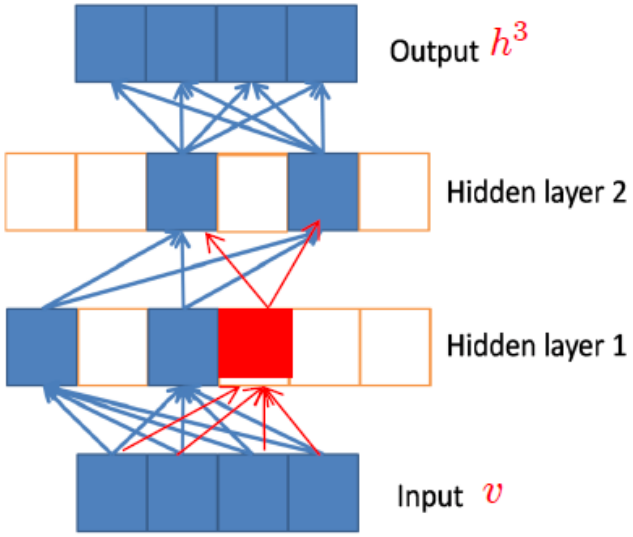


$$h^3 = \begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix} \underbrace{\begin{bmatrix} \ddots & \\ & \ddots \end{bmatrix}}_{2\times 2} \underbrace{\begin{bmatrix} \cdots & \cdots \end{bmatrix}}_{2\times 4} v = A_1 v$$

4×2



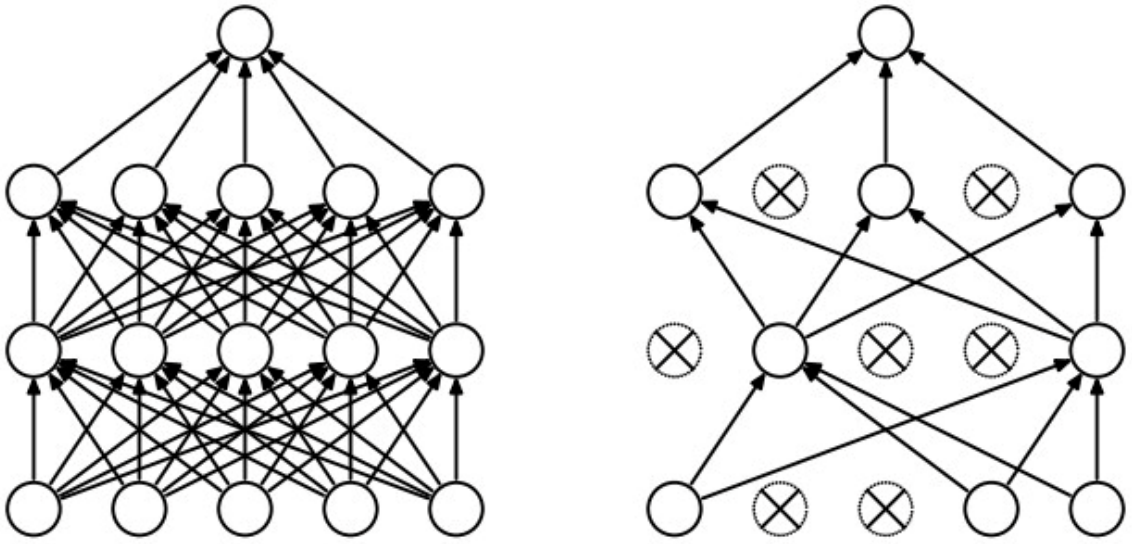




$$h^3 = \underbrace{\begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix}}_{4\times 2} \underbrace{\begin{bmatrix} \ddots & \\ & \ddots \end{bmatrix}}_{2\times 3} \underbrace{\begin{bmatrix} \cdots & \cdots \end{bmatrix}}_{3\times 4} v = A_2 v$$

Dropout(Regularization Method)

- 각 학습 단계마다, 특정 확률로(예 : 50%) random하게 hidden layer에 있는 unit들을 없애고 학습하는 방법
- Ensemble 개념을 적용
  - 여러 개의 model을 사용하여 평균값을 쓰면 하나의 model을 쓰는 경우보다 좋음
  - 하나의 model로 비슷한 효과를 낼 수 있는 방법



(a) Standard Neural Net

(b) After applying dropout.

Other Regularization Methods

- Weight Decay(L2 Regularization)

$$E(w) = E_0(w) + \frac{1}{2}\lambda \sum_i w_i^2$$

- Batch Normalization
  - Benefits of BN
    - Increase learning rate
    - Remove dropout
    - Reduce L2 weight decay
    - Remove LRN

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

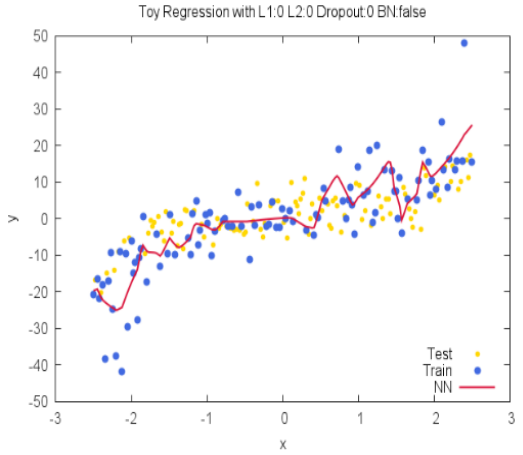
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

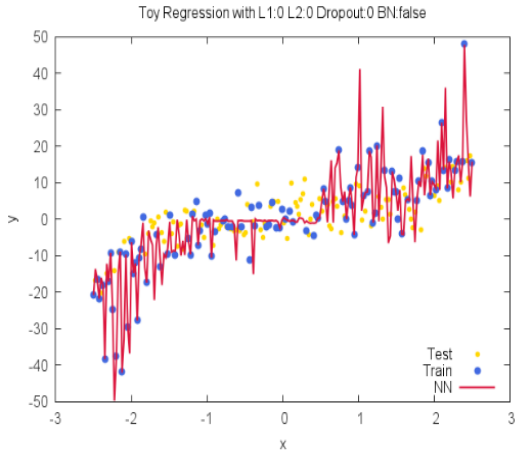
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

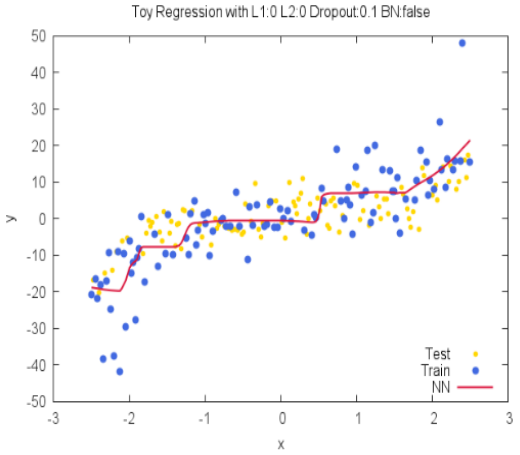
Regularization Methods

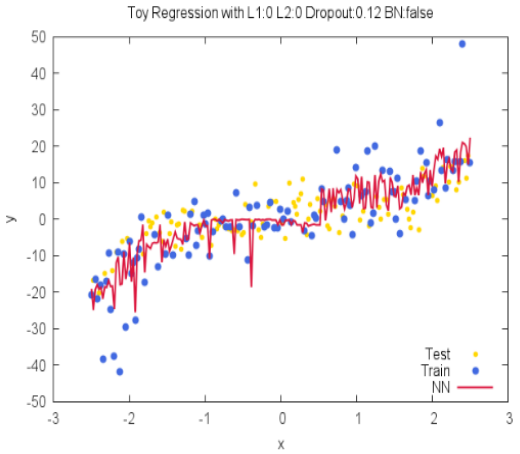
No Regularization



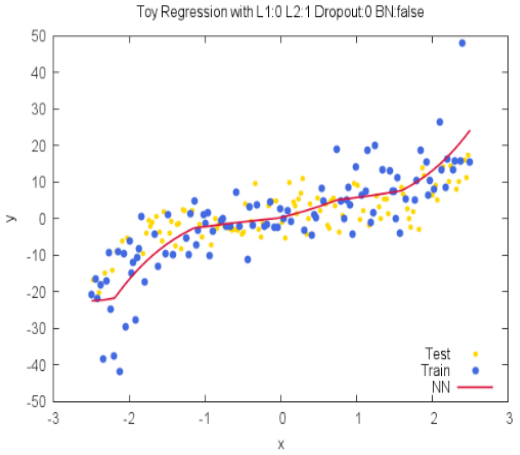


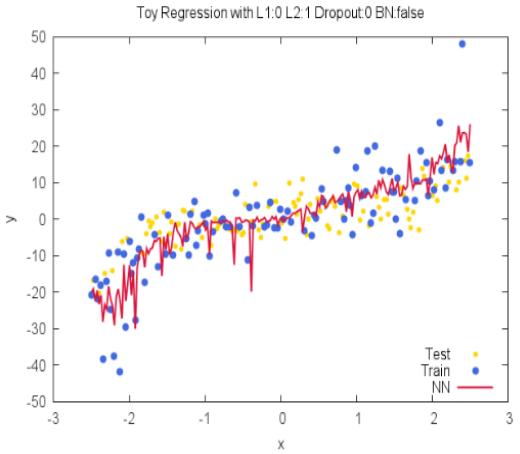
Dropout



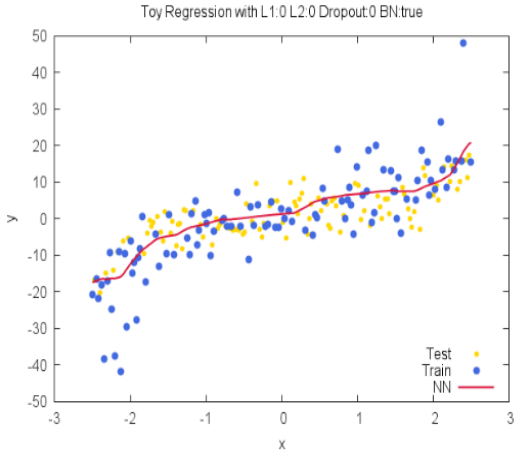


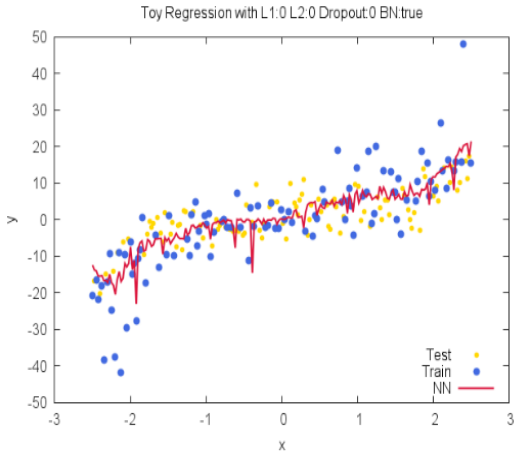
L2 Regularization





Batch Normalization

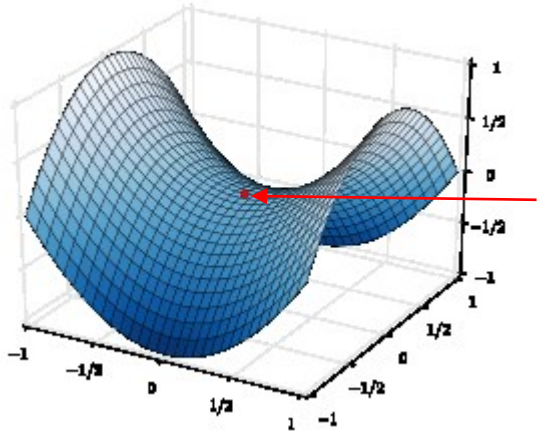


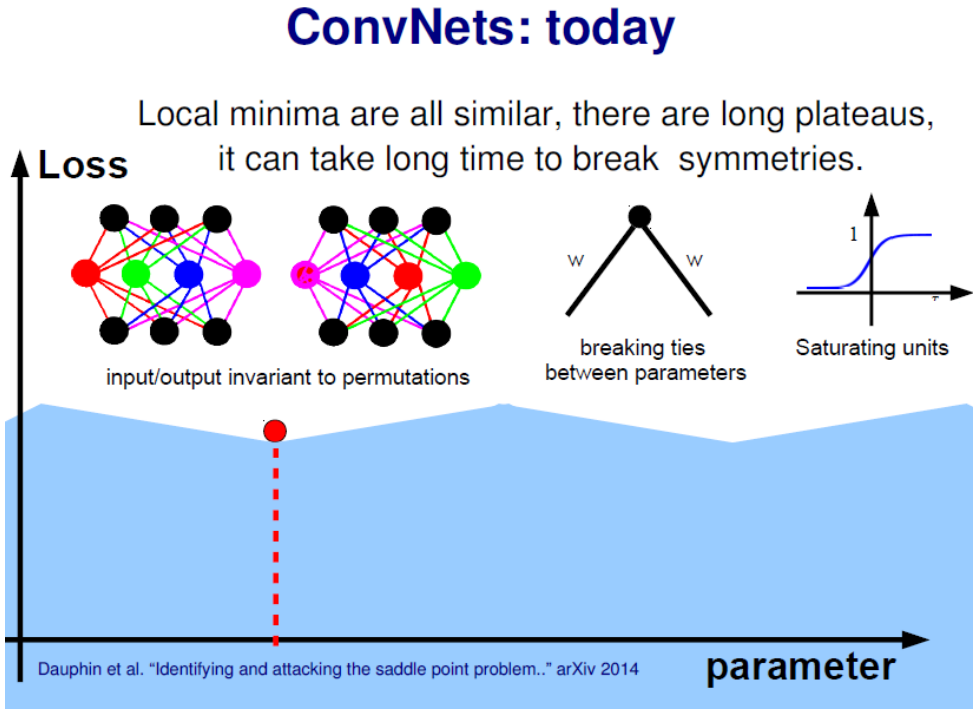


Local Minima

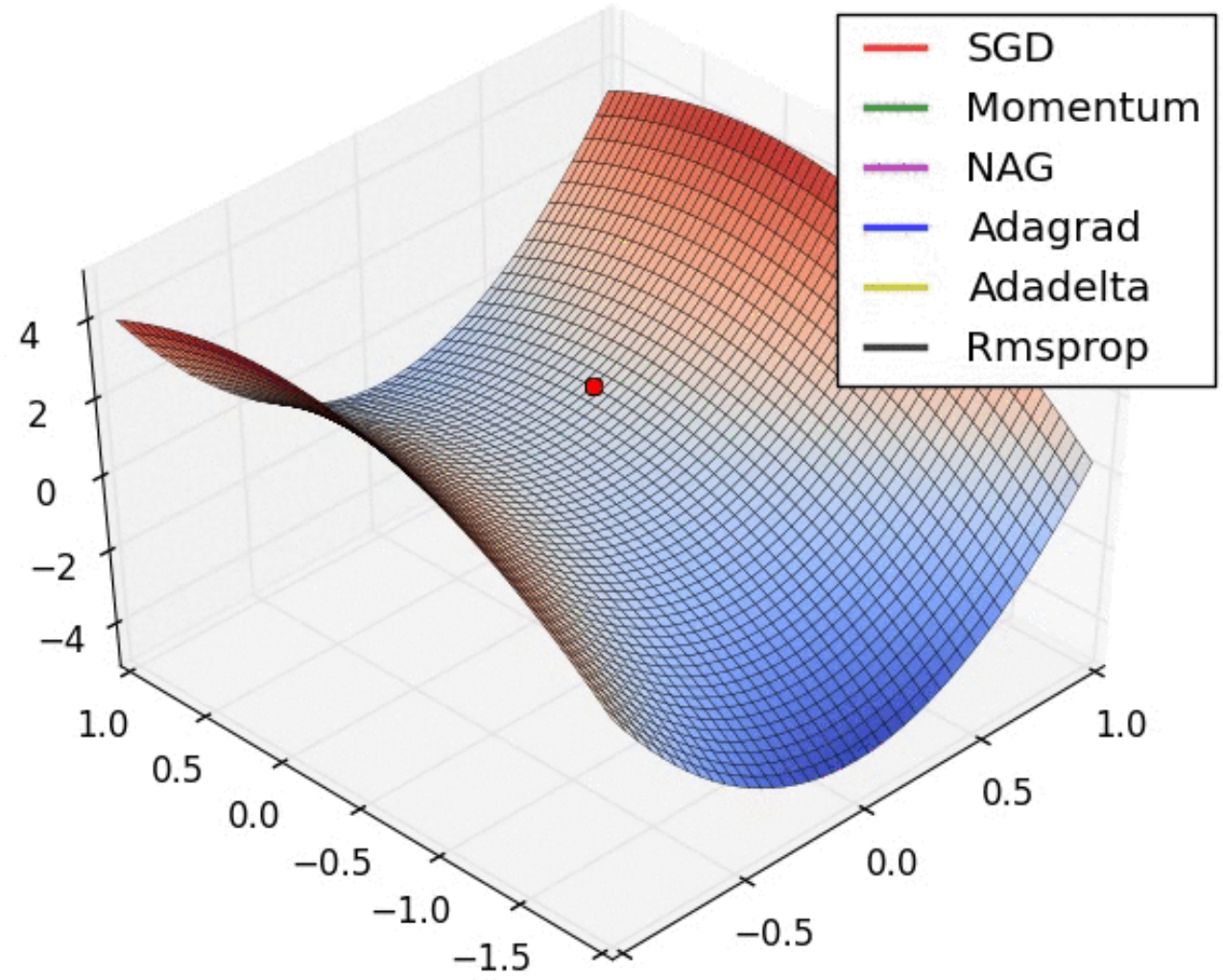
Local Minima에 대하여

- minimum이라고 하는 것은 현재 차원에서 이동할 수 있는 모든 방향으로의 gradient 값이 증가하는 방향이어야 하는데 이런 경우는 확률적으로 희박함
- DNN과 같은 고차원 구조에서는 대부분은 local minima가 아니라 saddle point일 가능성이 높음
- 만약 실제 local minima가 존재한다면 그것은 global minimum과 거의 차이가 없을 가능성이 높음(neural network의 대칭성)





Optimization Methods



그 밖에 Deep Learning을 가능하게 했던 것들

- Hardware
  - CPUs, GPUs!, ASICs https://youtu.be/-P28LKWTzrI
- Organized Large Datasets
  - ImageNet
- Algorithms and Research
  - Backprop, CNN, LSTM
- Software and Infrastructure
  - Git, AWS, Amazon Mechanical Turk, TensorFlow, …
- Financial Backing of large Companies
  - Google, Facebook, Amazon, …

Deep Learning in One Slide

- **What is it:**
  Extract useful patterns from data.
- **How:**
  Neural network + optimization
- **How (Practical):**
  Python + TensorFlow & friends
- **Hard Part:**
  Good Questions + Good Data
- **Why now:**
  Data, hardware, community, tools, investment
- **Where do we stand?**
  Most big questions of intelligence have not been answered nor properly formulated

Exciting progress:

- Face recognition
- Image classification
- Speech recognition
- Text-to-speech generation
- Handwriting transcription
- Machine translation
- Medical diagnosis
- Cars: drivable area, lane keeping
- Digital assistants
- Ads, search, social recommendations
- Game playing with deep RL

Deep Learning's Challenge – Toward the Real World

Deep Learning's Challenge – Unintended Consequences

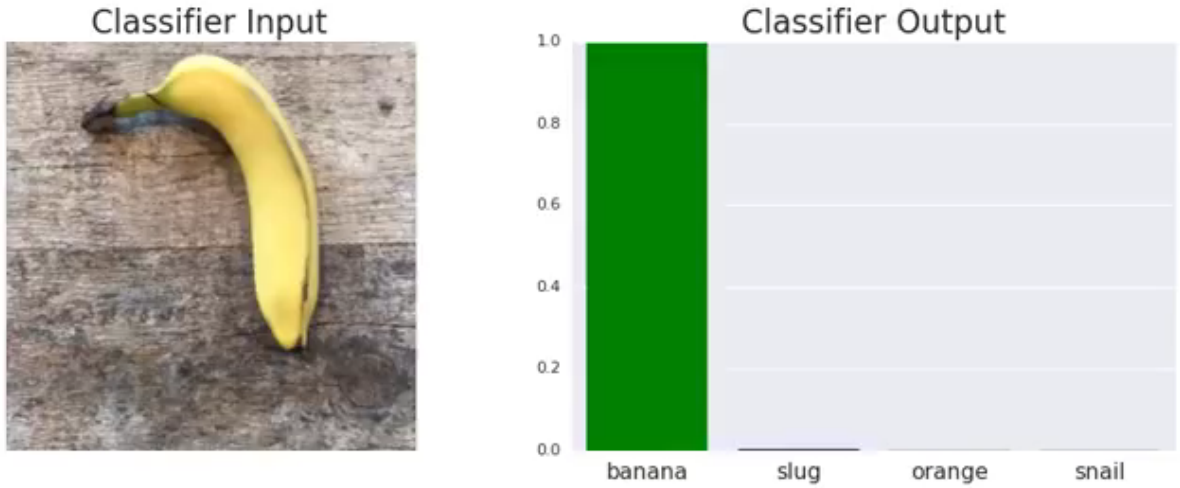
Human Play

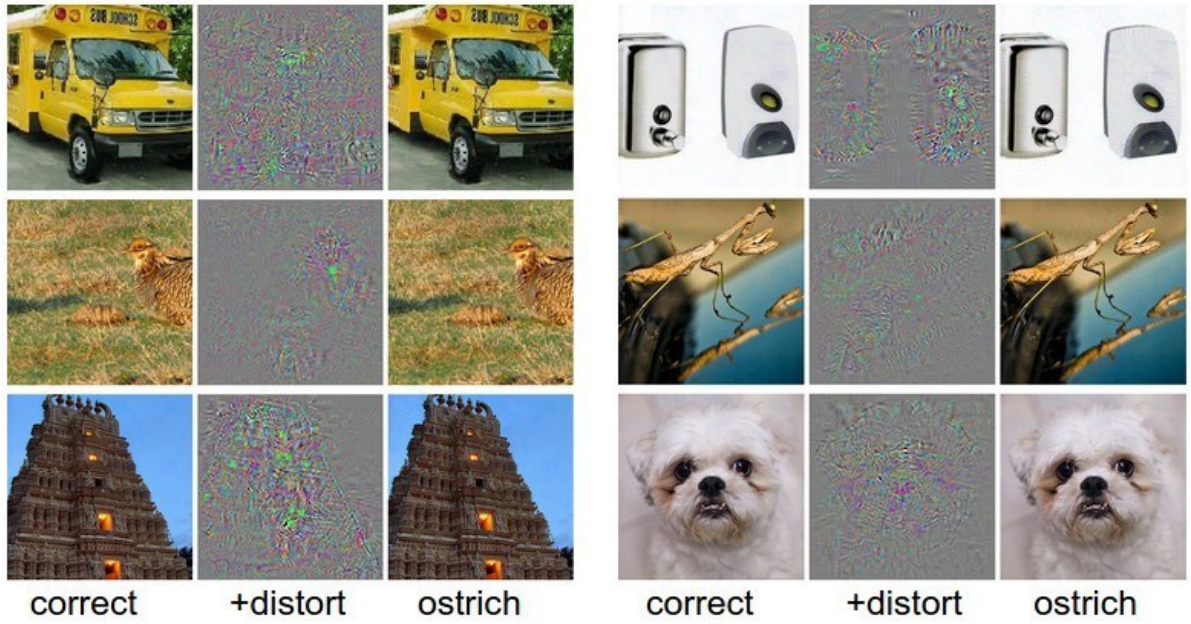
AI Play

Player gets reward based on:

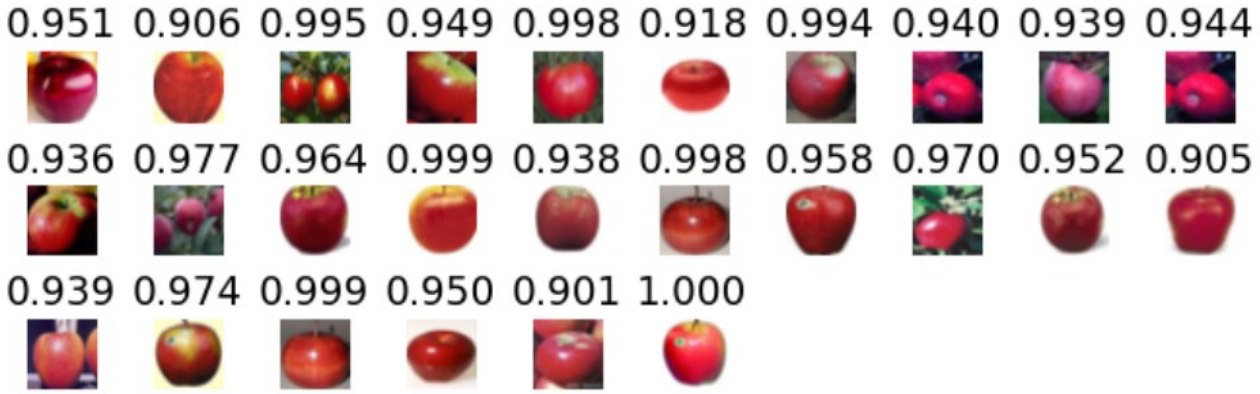
1. Finishing time
2. Finishing position
3. Picking up "turbos"

Adversarial Attack

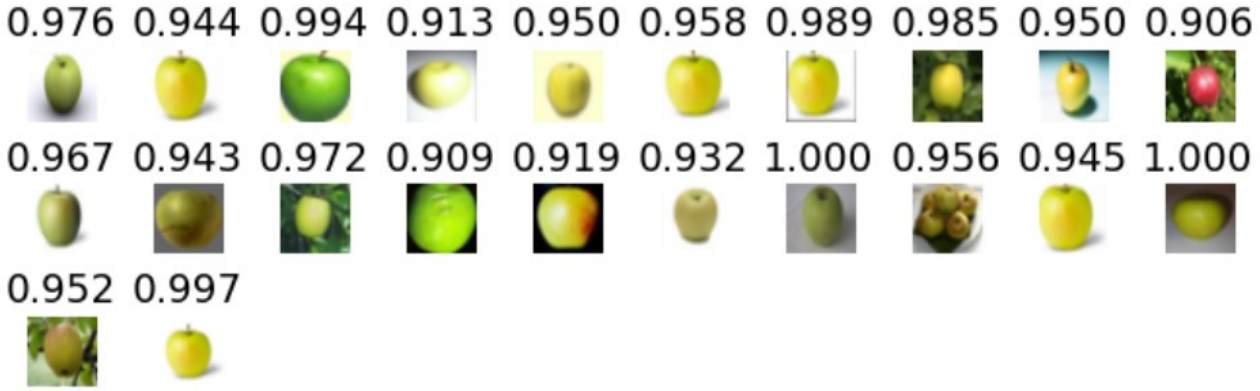




Uncertainty

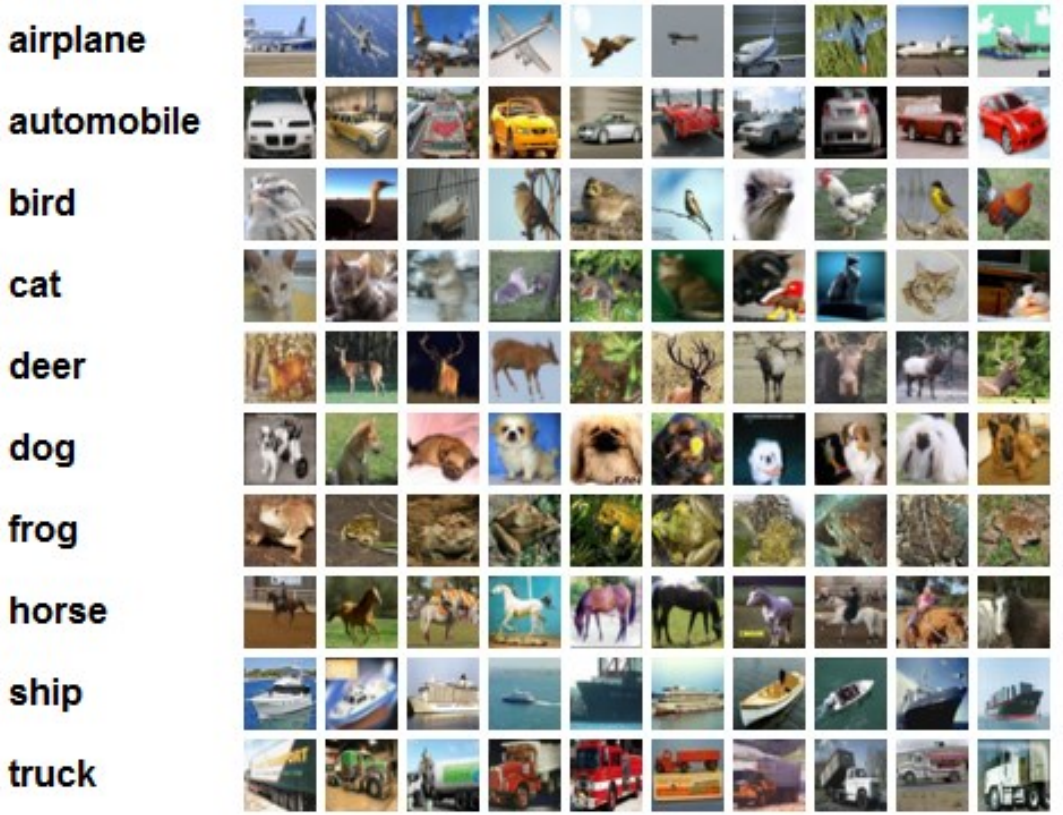


CIFAR-100's *apple* misclassified as CIFAR-10's *automobile* class with $p > 0.9$.



CIFAR-100's *apple* misclassified as CIFAR-10's *frog* class with $p > 0.9$.

Cifar 10 dataset

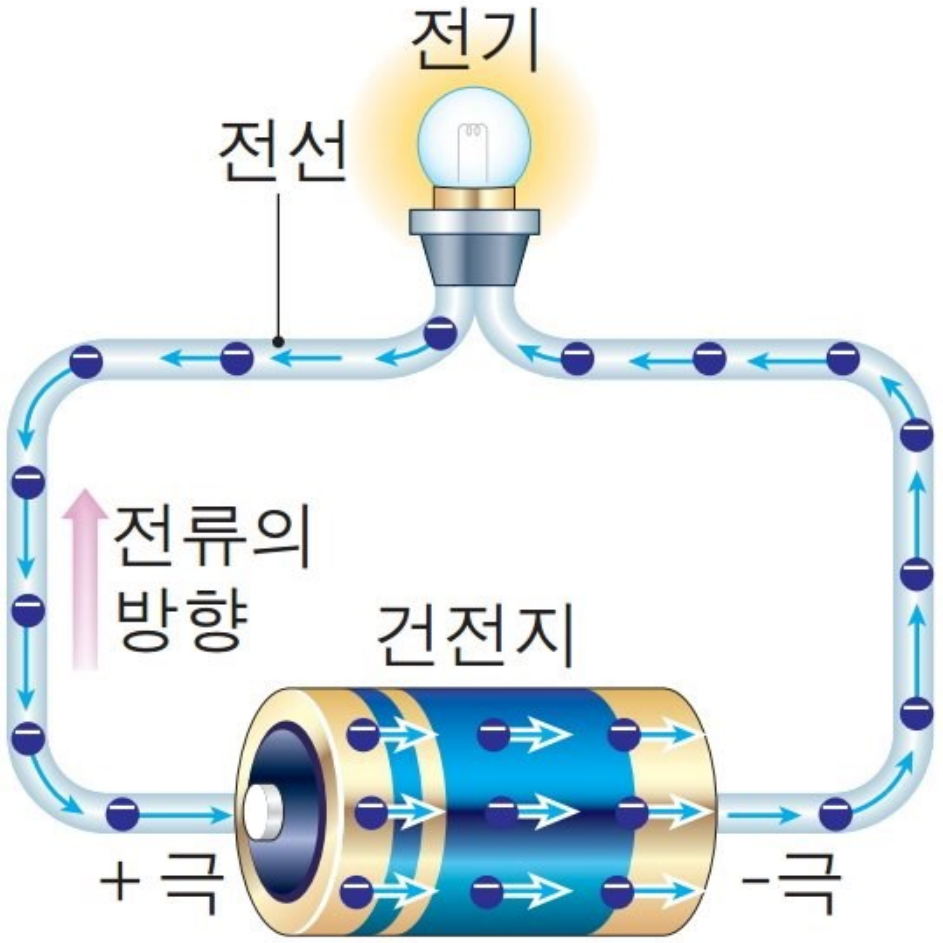


Limitations of Deep Learning

- Not sample efficient
- Compute intensive
- Poor at handling prior knowledge or uncertainty
- Not interpretable
- Suffers from hyper-parameter optimization
- Vulnerable to adversarial attack

Today's Artificial Intelligence?

L'Avion III de Clément Ader, 1897
(Musée du CNAM, Paris)



▲ 전자의 이동과 전류의 방향

From 중학교 3학년 과학3 교과서