# Machine Learning (ML) & Deep Learning (DL) Basics

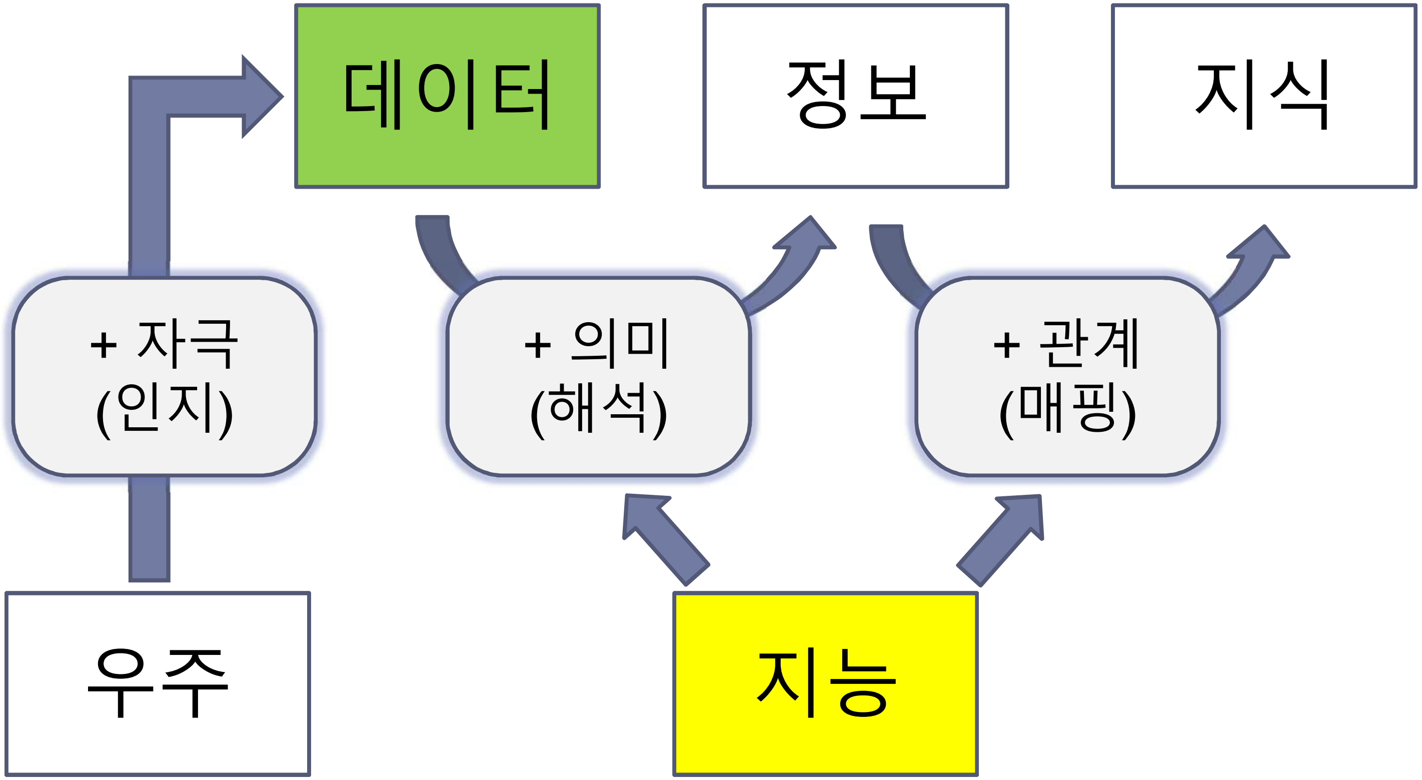
Saehwa Kim

Information and Communications Engineering
Hankuk University of Foreign Studies

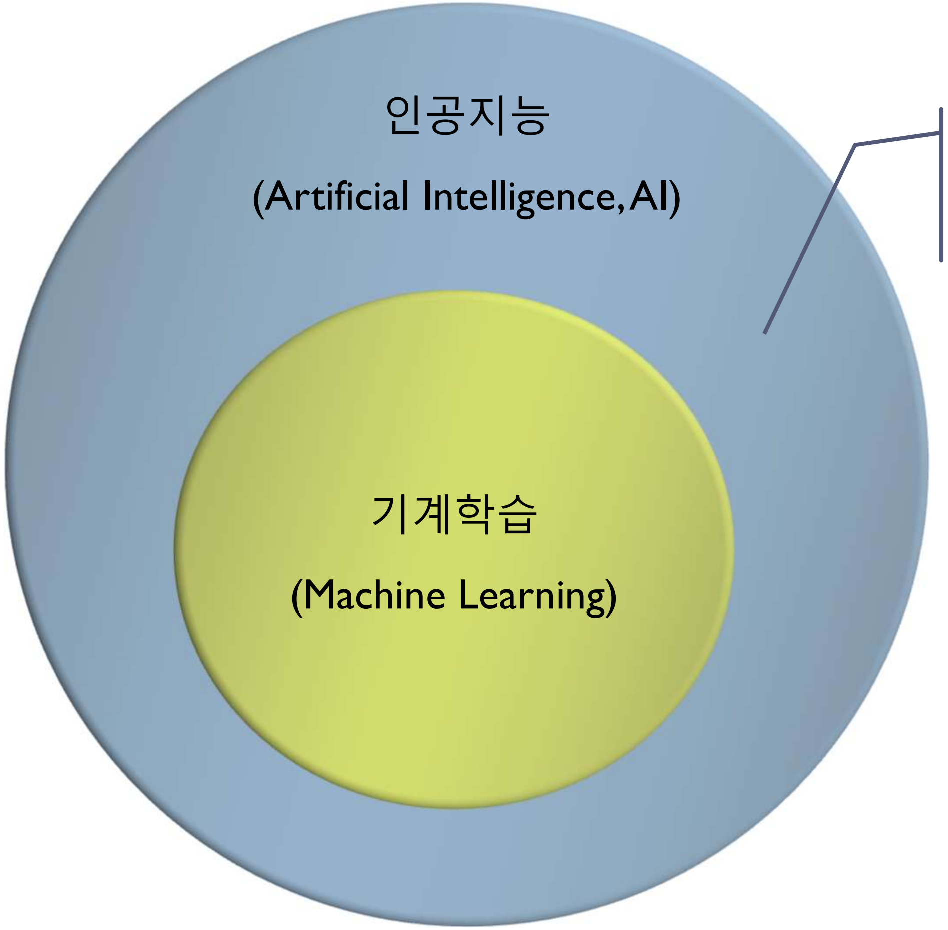
# Contents

- Machine Learning Basics
- Deep Learning Basics
- Bias vs. Variance Tradeoffs
- Performance Metrics for Binary Classifier

# 데이터와 지능



# 인공지능과 기계학습

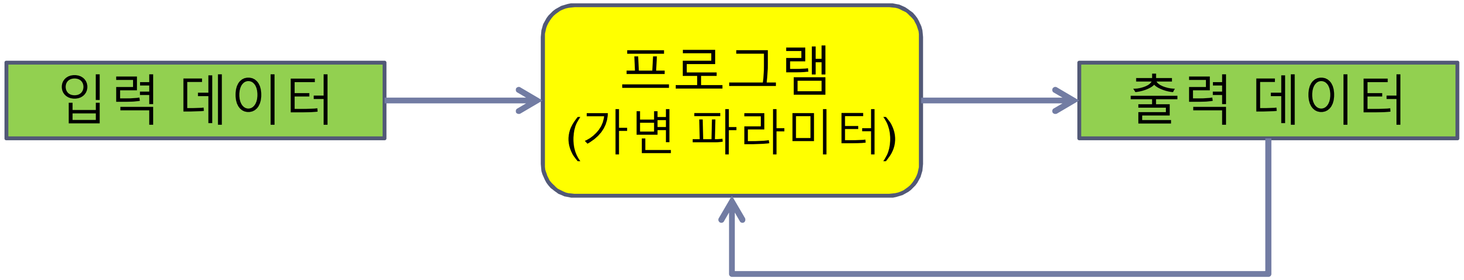



기계학습이 아닌 인공지능:
(예) 지식 창고(knowledge base),
Rule-based system

# 전통적인 프로그래밍 대 기계학습

전통적인 프로그래밍


기계학습

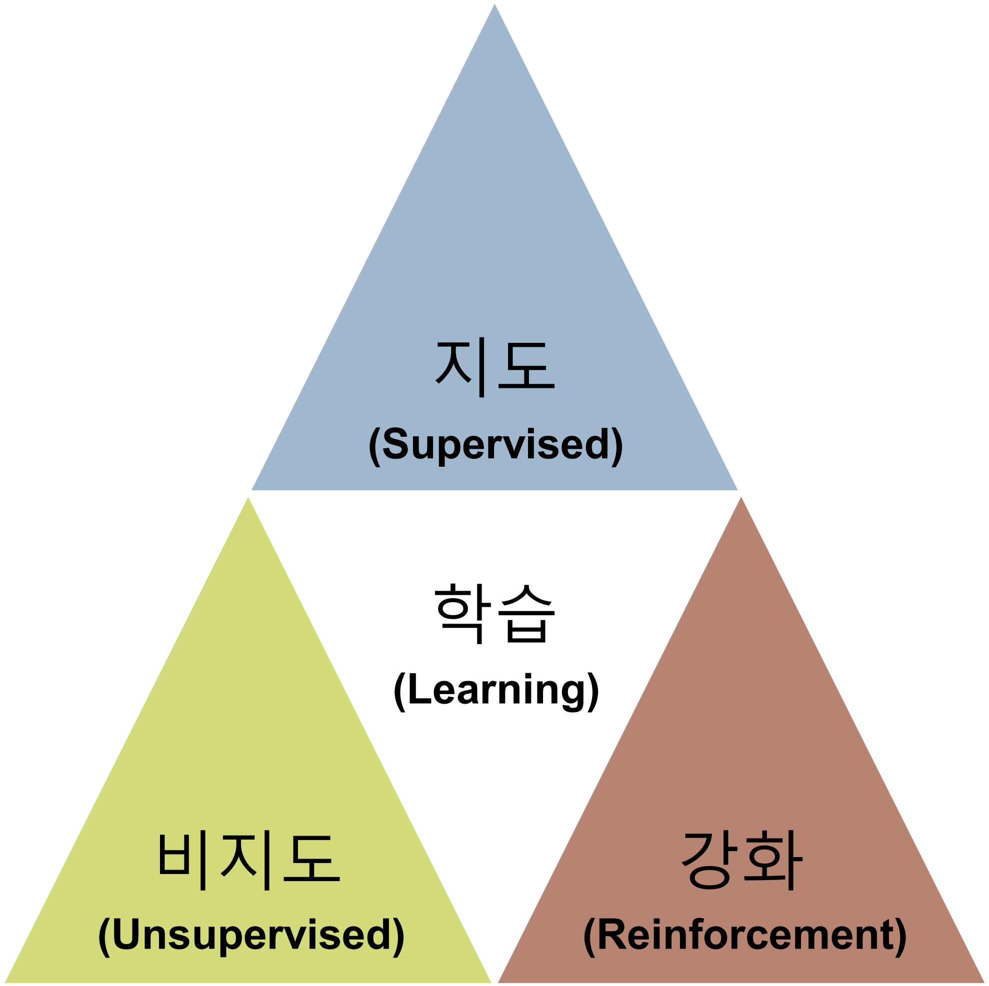


# Machine Learning: Training + Testing

- Training (훈련)

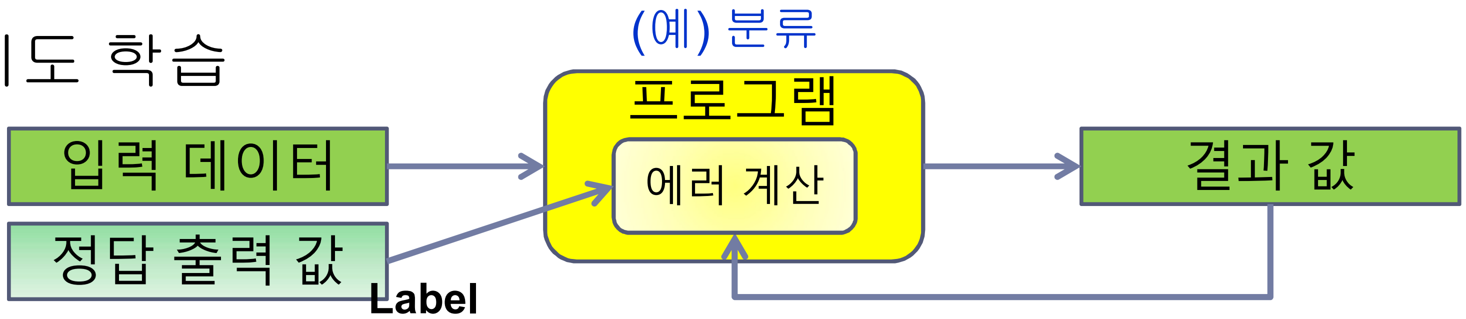

- Testing (시험) == Inference (추론)

# 3 Kinds of Machine Learning (1)

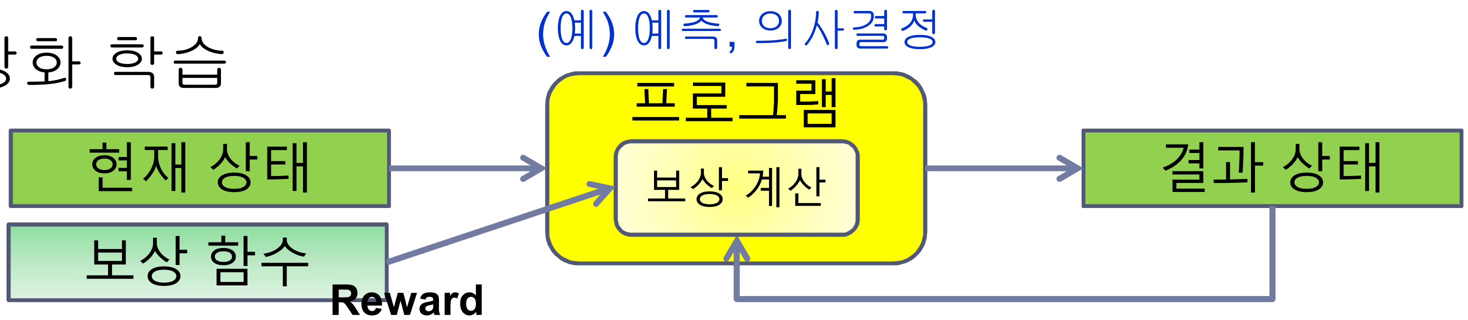


# 3 Kinds of Machine Learning (2)

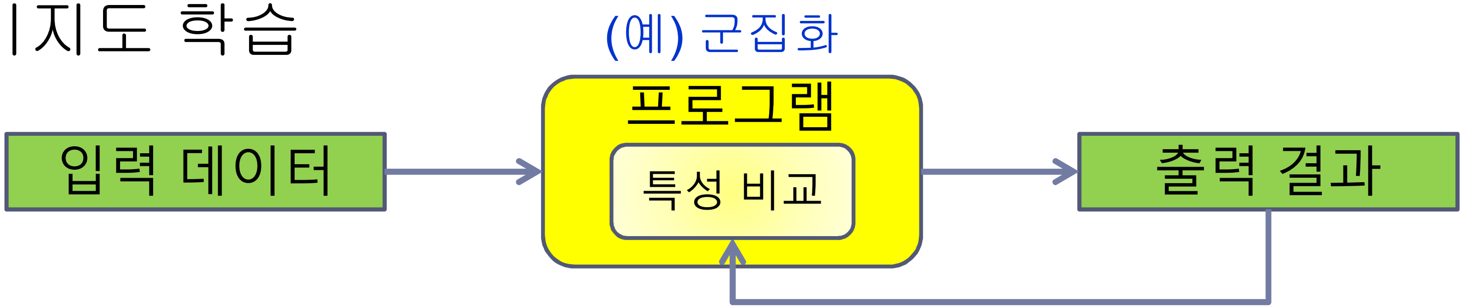
지도 학습




강화 학습




비지도 학습

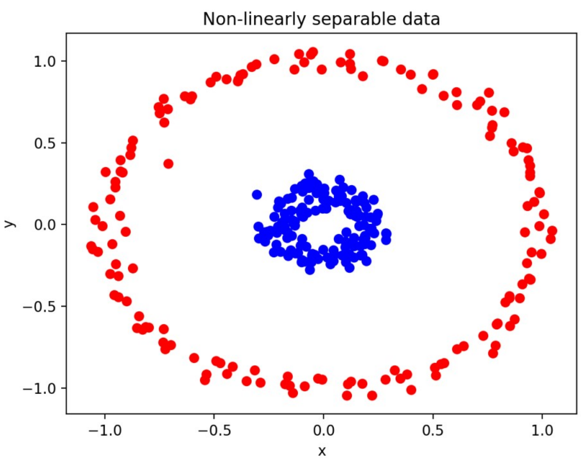


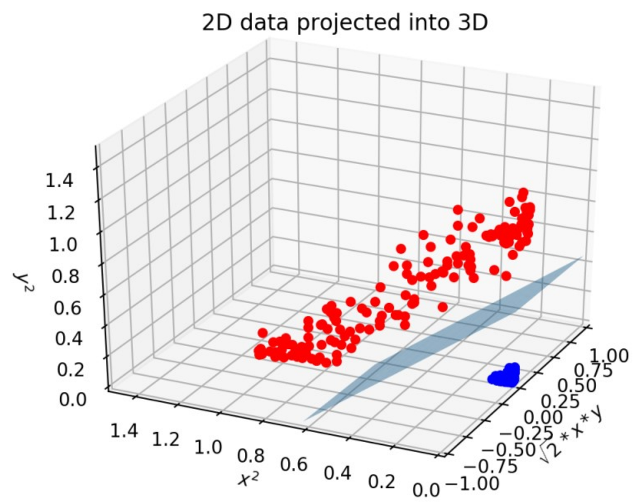
# 3 Kinds of Machine Learning (3)

- Supervised Learning
  - Linear Regression = Least Squares Regression
    - Continuous value prediction
  - Logistic Regression = Sigmoid Regression = Minimum Cross Entropy Regression
    - Discrete binary classification
  - Softmax Regression = Multinomial Logistic Regression
    - Discrete multiclass classification
  - Support Vector Machine (SVM, 지지 벡터 기계)
  - Decision Tree (의사 결정 나무)
  - Random Forest (무작위 숲)
- Unsupervised Learning
  - Variational Auto Encoder (VAE)
  - Principal Component Analysis (PCA)
  - K-Means Clustering (K-평균 군집화)
- Reinforcement Learning
  - Temporal Difference, SARSA, Q-Learning, …

# 지도 학습 (1)

- (예) Support Vector Machine (SVM, 지지 벡터 머신)
  - 분류 또는 회귀 분석에 사용 가능한 초평면(영어: hyperplane) (집합)
  - 선형으로 분리할 수 없는 점들을 분류하기 위해 커널(kernel) 함수를 사용
    - 커널(kernel)은 원래 가지고 있는 데이터를 더 높은 차원의 데이터로 변환

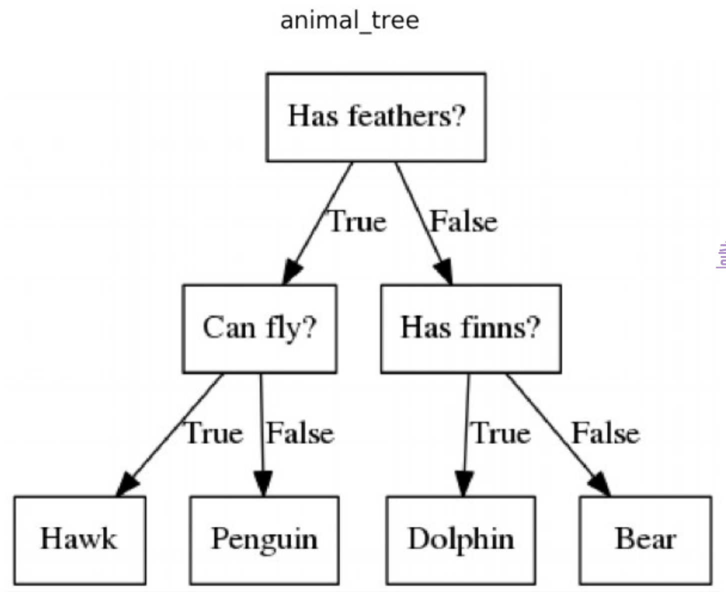



- 출처: http://hleecaster.com/ml-svm-concept/

# 지도 학습 (2)

- (예) Decision Tree (의사결정 나무)




출처: https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53

- (예) Random Forest (무작위 숲)
  - 앙상블 머신러닝 모델
  - 다수의 의사결정 트리를 만들고, 그 나무들의 분류를 집계해서 최종적으로 분류

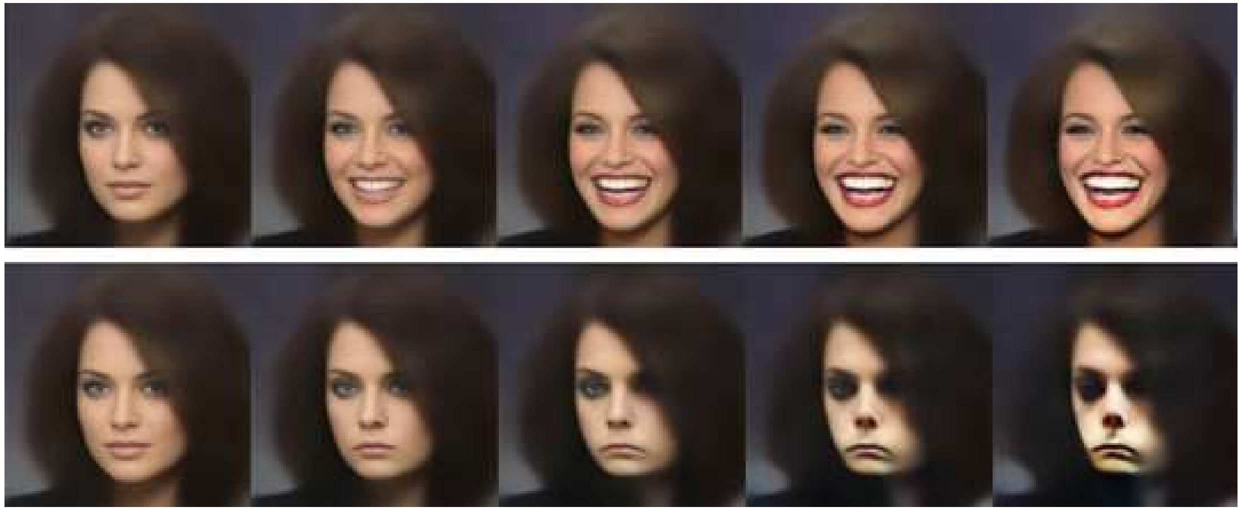
# 비지도 학습 (1)

- (예) VAE (Variational Auto Encoder)
  - A continuous space of faces

https://gaussian37.github.io/deep-learning-chollet-8-4/

# 비지도 학습 (2)

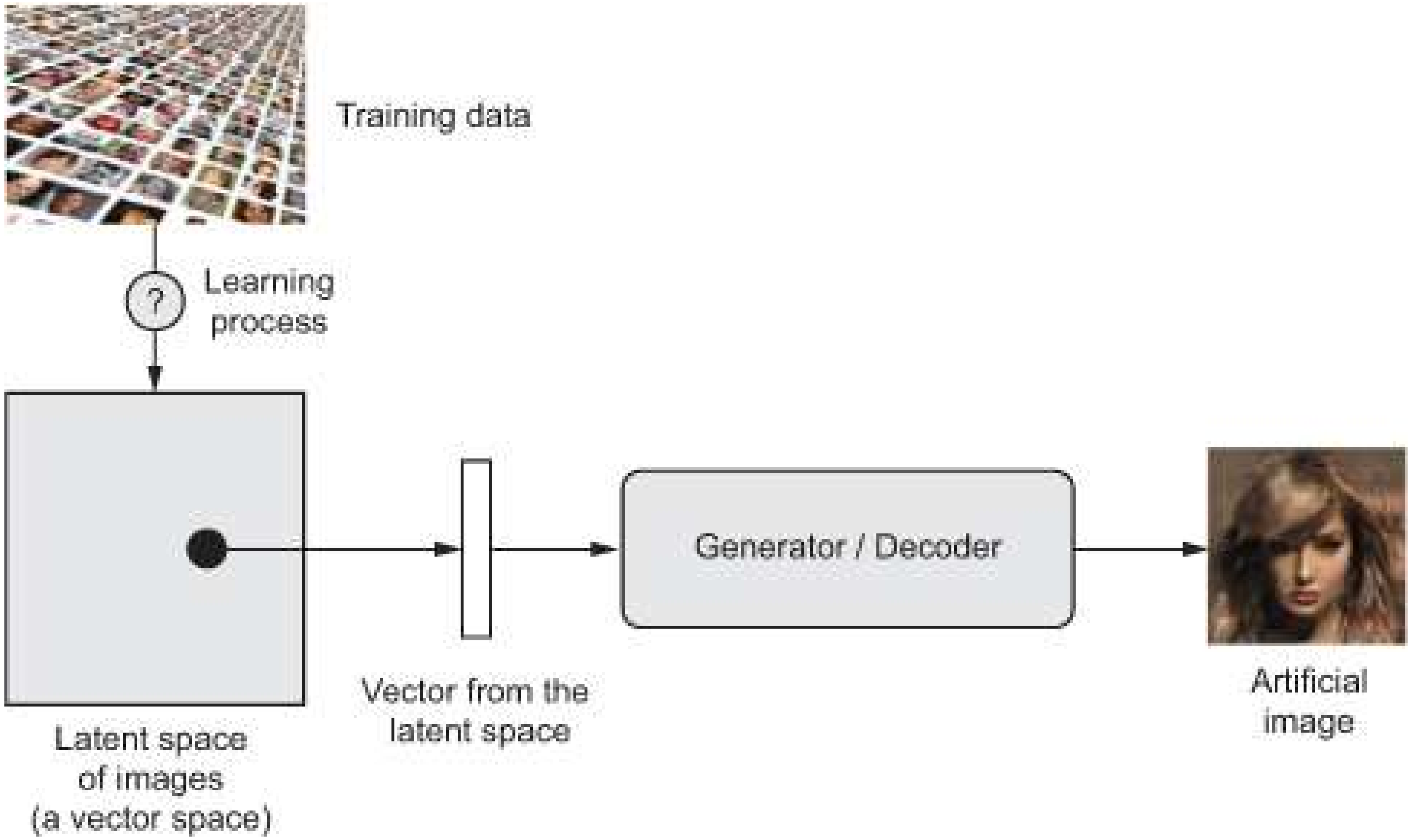
- (예) VAE (Variational Auto Encoder)
  - Smile Vector



https://gaussian37.github.io/deep-learning-chollet-8-4/

# 비지도 학습 (3)

- (예) VAE (Variational Auto Encoder)




https://gaussian37.github.io/deep-learning-chollet-8-4/

# 비지도 학습 (4)

- (예) principal component analysis (PCA)



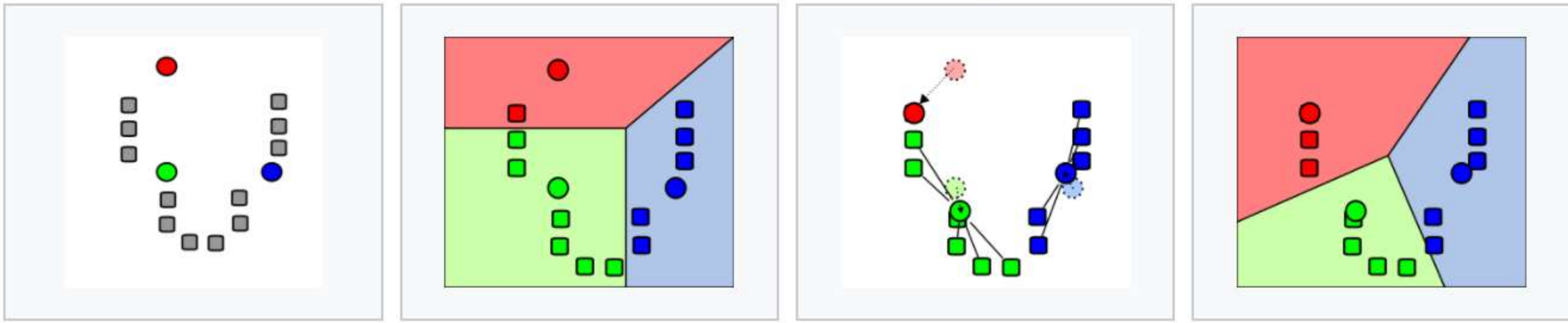
https://www.oreilly.com/library/view/deep-learning-for/9781788295628/02af7a13-e42d-4adc-96af-35e1120706b0.xhtml

# 비지도 학습 (5)

https://en.wikipedia.org/wiki/K-means_clustering

- (예) K-Means Clustering
  - 주어진 데이터를 k개의 클러스터로 묶는 알고리즘
  - 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작
    - 각 그룹의 중심 (centroid)과 그룹 내의 데이터 오브젝트와의 거리의 제곱합을 비용 함수로 정함
    - 비용 함수 값을 최소화하는 방향으로 각 데이터 오브젝트의 소속 그룹을 업데이트 해 클러스터링

**Demonstration of the standard algorithm**



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

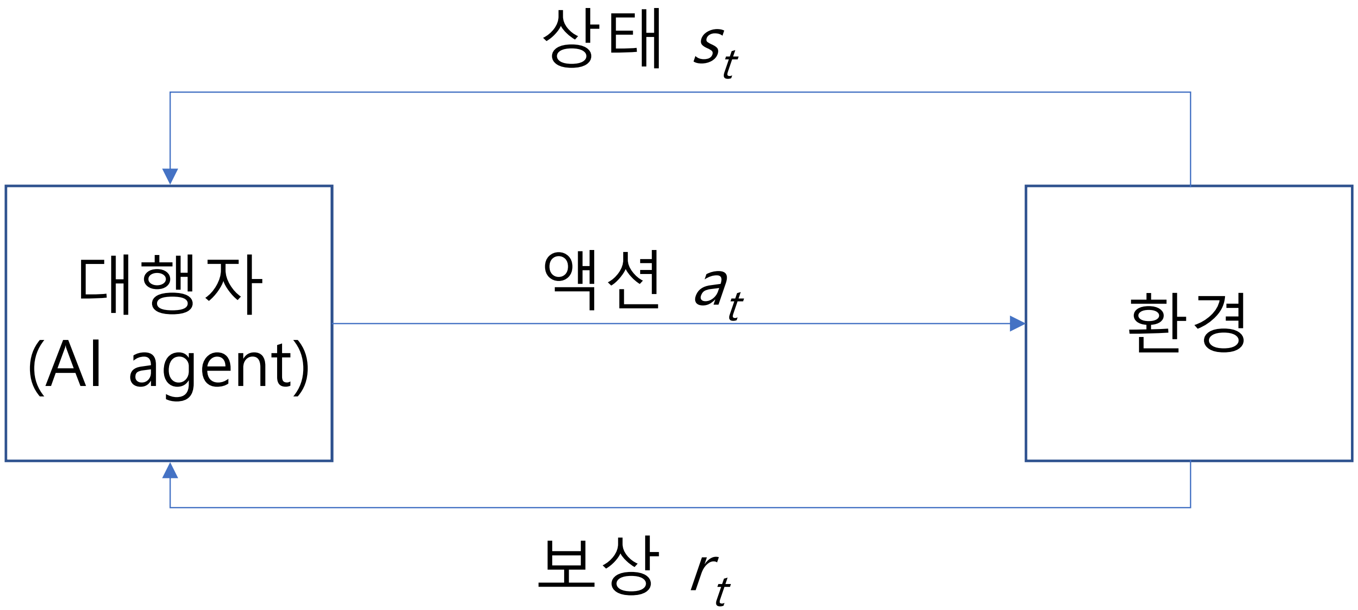
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

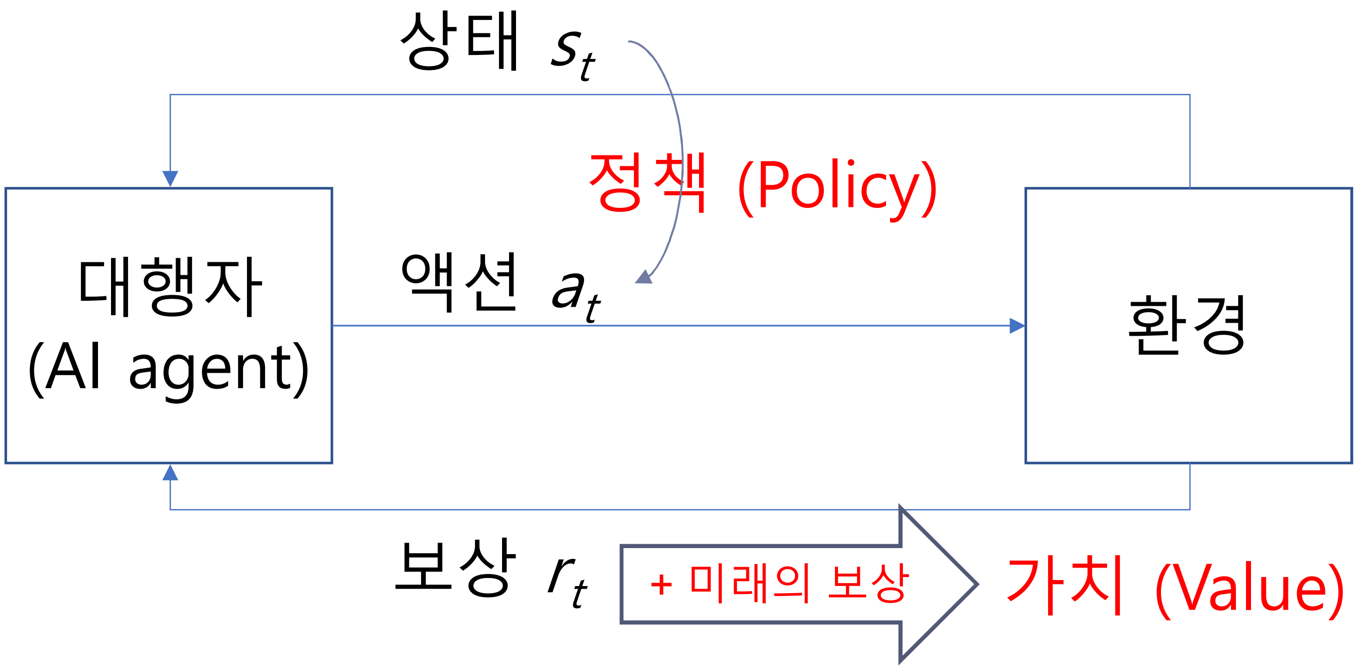
# Reinforcement Learning (1)

- 인간 지식 입력이 필요하지 않음
  - 지도/비지도 학습은 빅데이터가 필요함
- 대행자(AI agent)가 스스로 경험을 통해 얻을 수 있는 **총 보상** 값이 최대가 되도록 **환경**의 **상태**에 따라 특정 **액션**을 취하도록 학습함



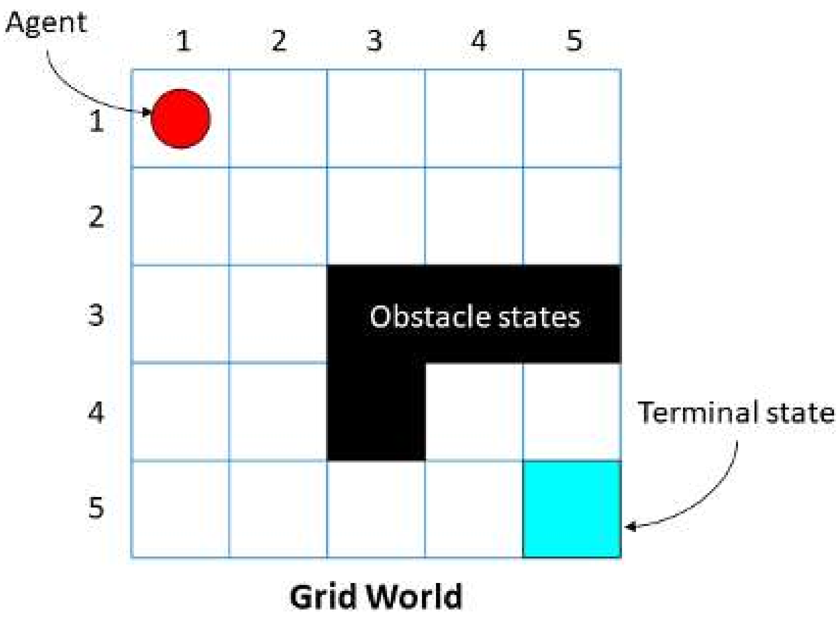
# Reinforcement Learning (2)

- 정책 (Policy)
  - 특정 상태로부터 특정 액션을 취하는 확률 분포
- 가치 (Value)
  - 특정 상태로부터 얻을 수 있는 총 보상 값: 즉각적인 보상 + 미래의 보상



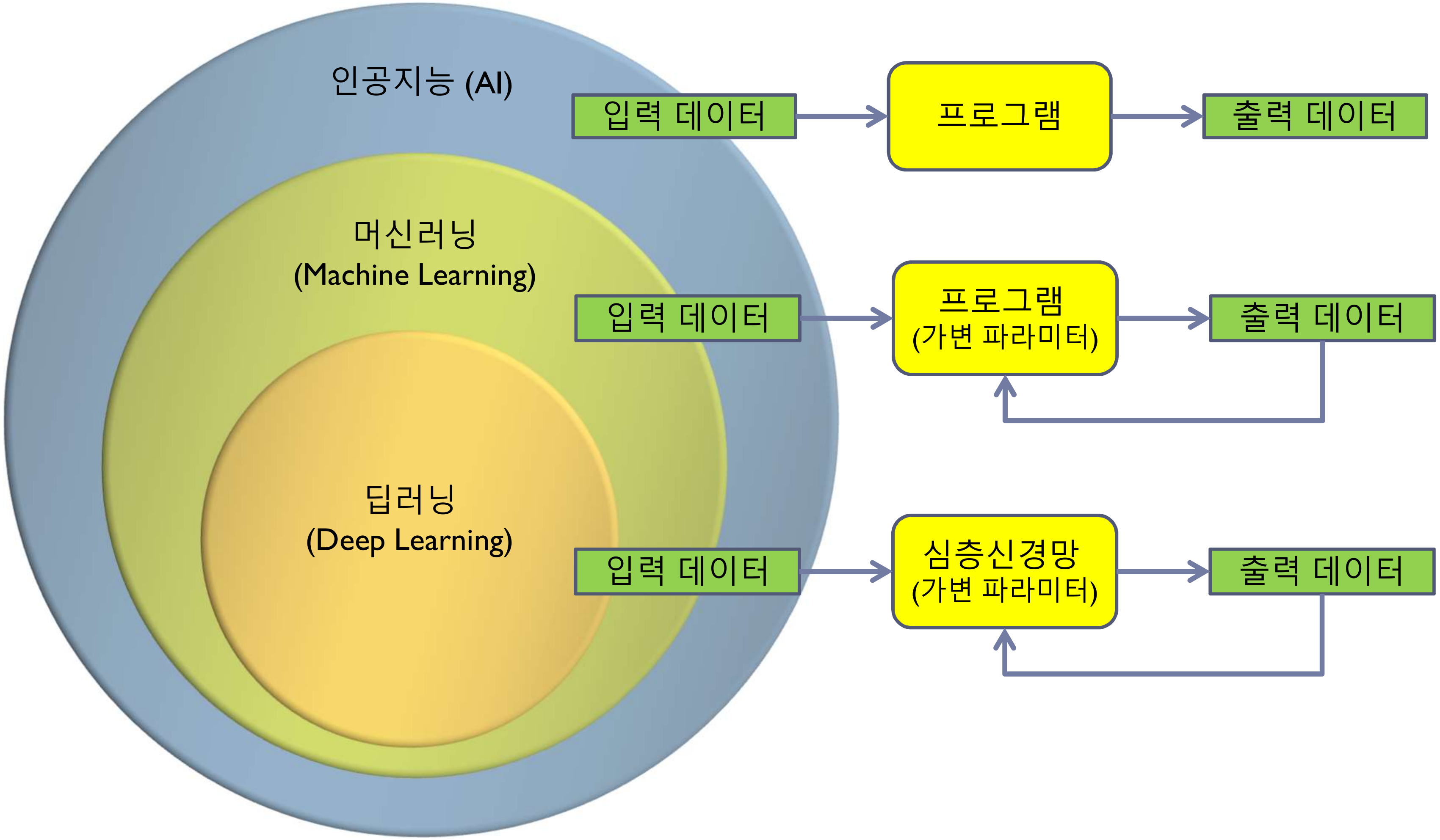
# Reinforcement Learning (3)

- Q Learning 등
  - Q (quality) 값: 특정 상태와 "특정 액션"으로부터 얻을 수 있는 총 보상 값
- Grid world 등의 간단한 문제만 해결 가능

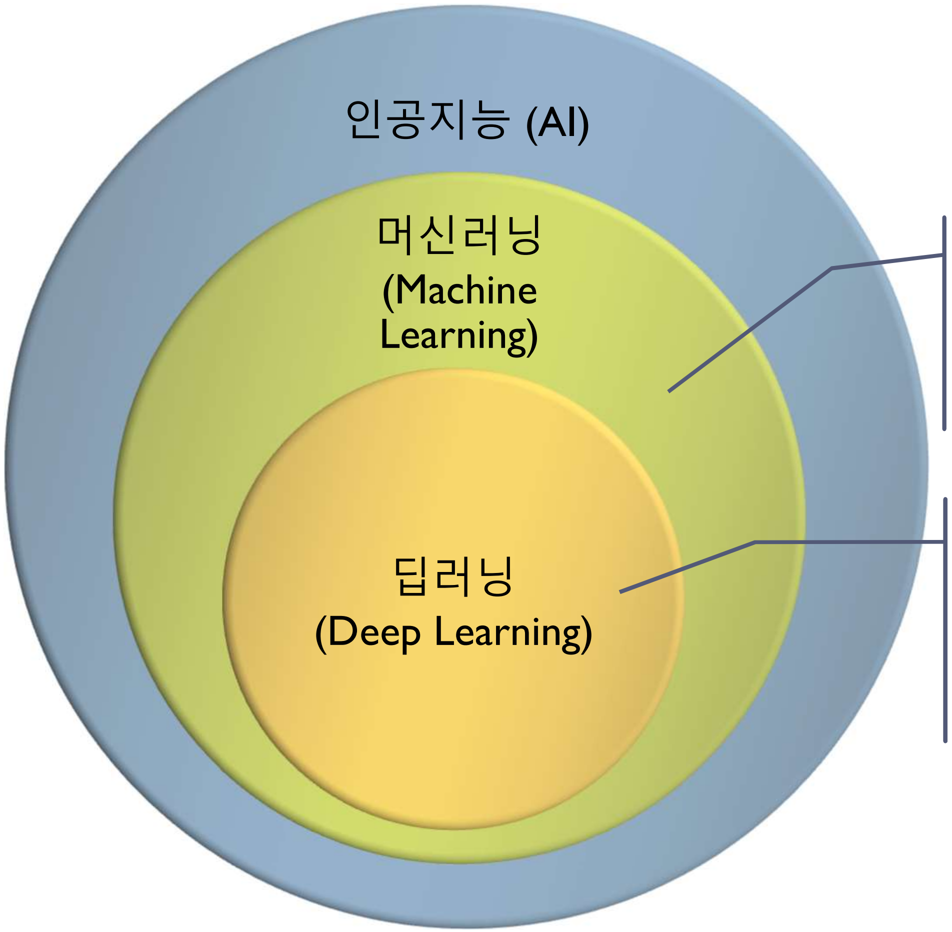



(이미지 출처: https://www.mathworks.com/help/reinforcement-learning/ug/create-custom-grid-world-environments.html)

# 머신러닝과 딥러닝


인공지능 (AI)
머신러닝
(Machine Learning)
딥러닝
(Deep Learning)
입력 데이터
프로그램
출력 데이터
입력 데이터
프로그램
(가변 파라미터)
출력 데이터
입력 데이터
심층신경망
(가변 파라미터)
출력 데이터

# 머신러닝과 딥러닝




심층학습이 아닌 기계학습 (Shallow Learning):
(예) SVM (Support Vector Machine), Perceptron neural network

≅ 심층 신경망 (Deep Neural Network)

주의: 신경망 중 딥러닝이 아닌 것도 있음

# 딥러닝이 아닌 머신러닝 예: 추천 서비스

- http://www.bloter.net/archives/238299

FeatureFu | 링크드인 | 머신러닝

## 링크드인, 자체 머신러닝 기술 오픈소스로 공개

이지현 | 2015.09.08 공유 0 댓글 0

언어 선택 Google 번역에서 제공 가+ 가-

링크드인이 머신러닝 기술 '피처푸'를 오픈소스 라이브러리로 공개했다.

링크드인은 피처푸를 '피처 엔지니어링(Feature engineering)'을 도와주는 도구라고 설명한다. 피처 엔지니어링이란 현상, 지식, 특성 등을 미리 알고, 머신러닝 모델을 적용하는 방식을 말한다. 링크드인은 9월4일 깃허브 계정을 통해 "피처푸는 기존에 있던 데이터를 따로 수정하지 않고 새로운 데이터를 생성하고 관리할 수 있다"라며 "분류, 클러스터링, 정규화 등에 이용할 수 있다"라고 설명했다.

Sample use cases:

1. Feature normalization

   "(min 1 (max (+ (* slope x) intercept) 0))" : scale feature x with slope and intercept, and normalize to [0,1]

2. Feature combination

   "(- (log2 (+ 5 impressions)) (log2 (+ 1 clicks)))" : combine #impression and #clicks into a smoothed CTR style feature

3. Nonlinear featurization

   "(if (> query_doc_matches 0) 0 1)" : negation of a query/document matching feature

4. Cascading modeling

   "(sigmoid (+ (+ (..) w1) w0))" : convert a logistic regression model into a feature

5. Model combination (e.g. combine decision tree and linear regression)

   "(+ (* model1_score w1) (* model2_score w2))" : combine two model scores into one final score

Expr: A super fast and simple evaluator for mathematical s-expressions written in Java.

▲피처푸 활용 예제(사진:깃허브)

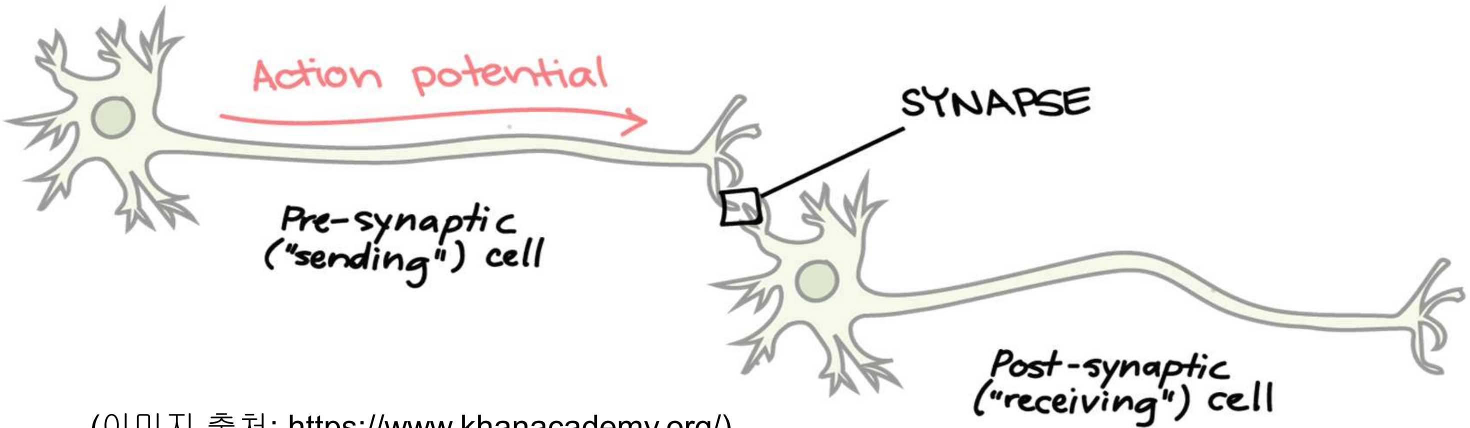
빙 자오 링크드인 소프트웨어 개발자는 "대규모 추천 시스템은 서로 다른 팀이 관리하고 모델링하느라 시간과 노력이 많이 든다"라며 "피처푸를 이용하면 새로운 특징을 적용하느라 주 혹은 월 단위로 걸리지 않으며 더 신속한 개발과 배포가 가능하다"라고 설명했다고 〈테크크런치〉는 9월4일 보도했다.

피처푸에는 수학적 S-표현식을 분해하고 평가하는 기술이 들어 있다. 자바로 쓰여진 코드라서 메이븐, 그래들에서도 이용할 수 있다. 피처푸를 적용한 예제는 링크드인 깃허브 계정에서 볼 수 있다.

- 피처푸 소개 페이지
- 피처푸 깃허브 페이지

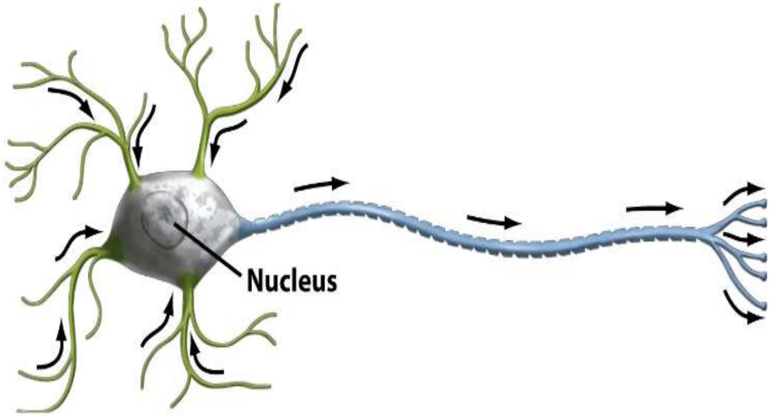
# 인간의 뇌 구조: Neural Network

- Neuron (신경세포)
  - 사람 뇌에는 약 $10^{11}$ (천억) 개의 neuron 존재
  - 개수가 사람에 따라 차이가 거의 없음
    (새로 생성되지 않음)
- Synapse (Neuron 간의 연결 부분)
  - 한 neuron에 보통 1천~1만 개의 synapse가 붙음
  - 성인 뇌에 $10^{14}$~$10^{15}$ (백조~ 천조) 개의 synapse 존재
  - 뇌 활동에 따라 생성/소멸
  - 학습과 기억의 원천

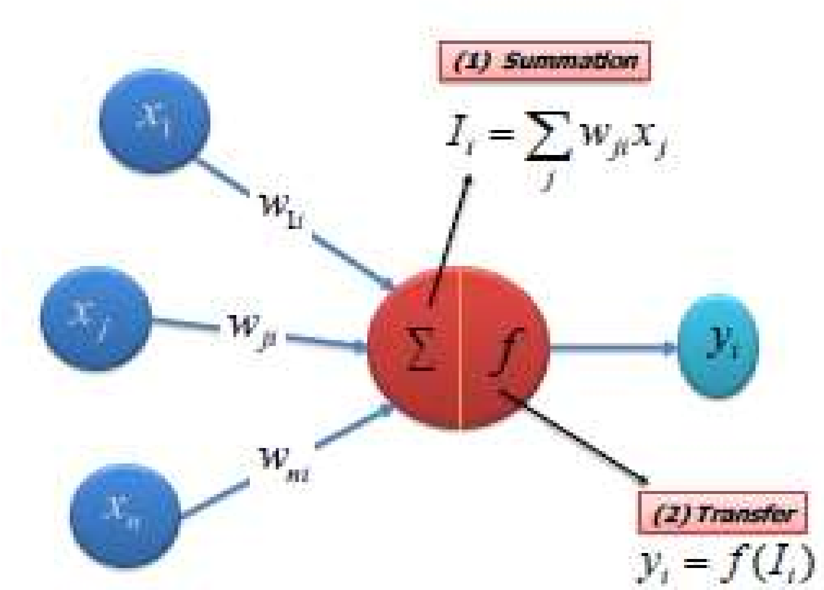



(이미지 출처: https://www.khanacademy.org/)

# Artificial Neural Network (인공 신경망)



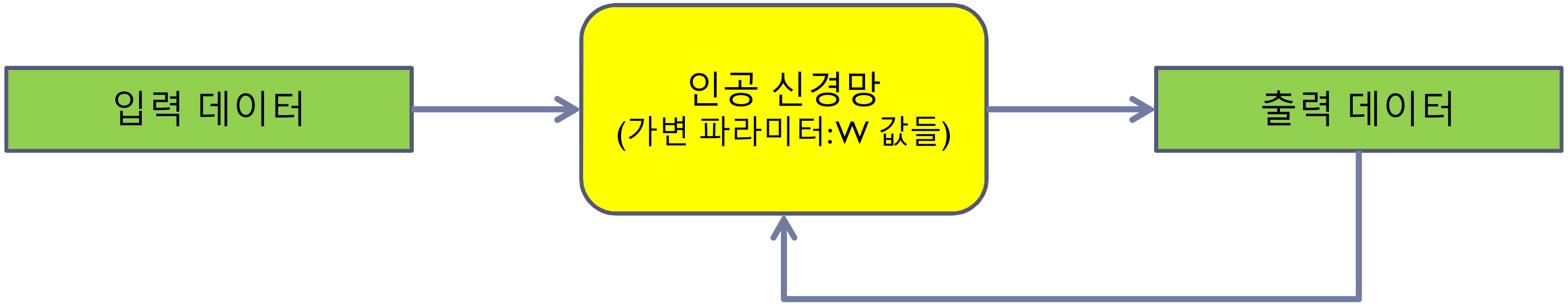

Feedforward Input Data




**Activation 함수**

(이미지 출처: http://www.saedsayad.com/artificial_neural_network.htm)

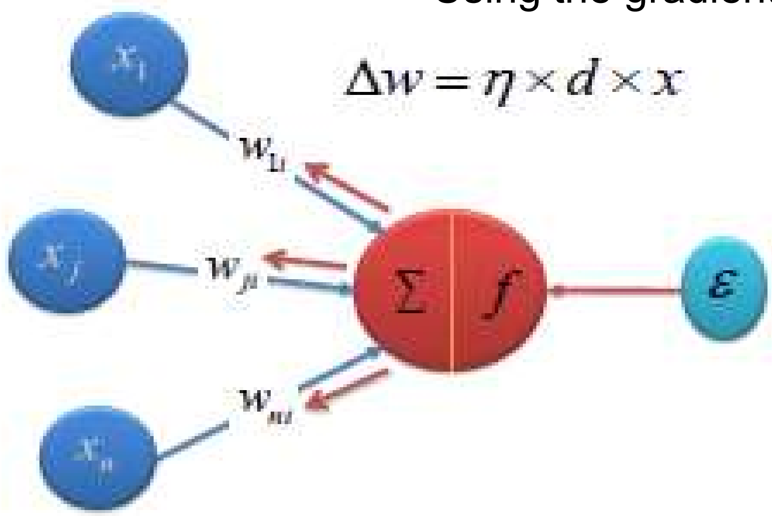
# Artificial Neural Network에서의 학습




Backward Error Propagation

Using the gradient descent algorithm

$$\Delta w = \eta \times d \times x$$

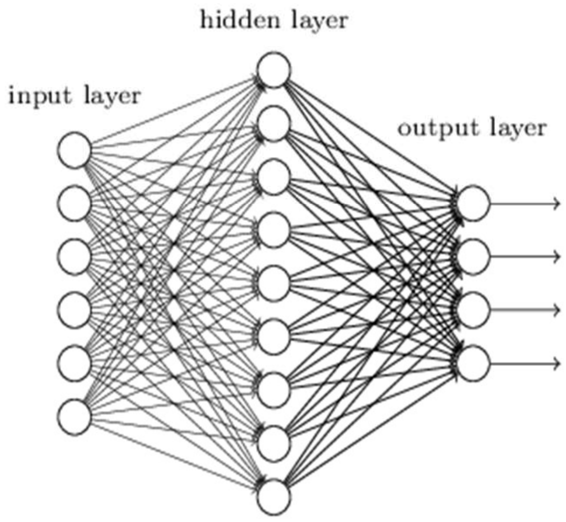



(이미지 출처: http://www.saedsayad.com/artificial_neural_network.htm)

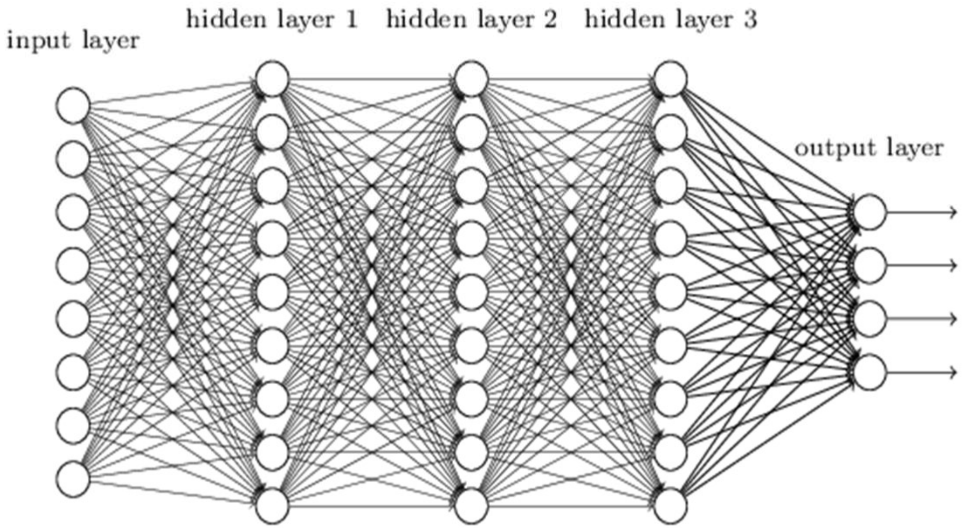
# Shallow (Non-Deep) vs. Deep Neural Network

Deep neural network: 은닉 계층(hidden layer)이 두 개 이상 있는 신경망

"Non-deep" feedforward neural network

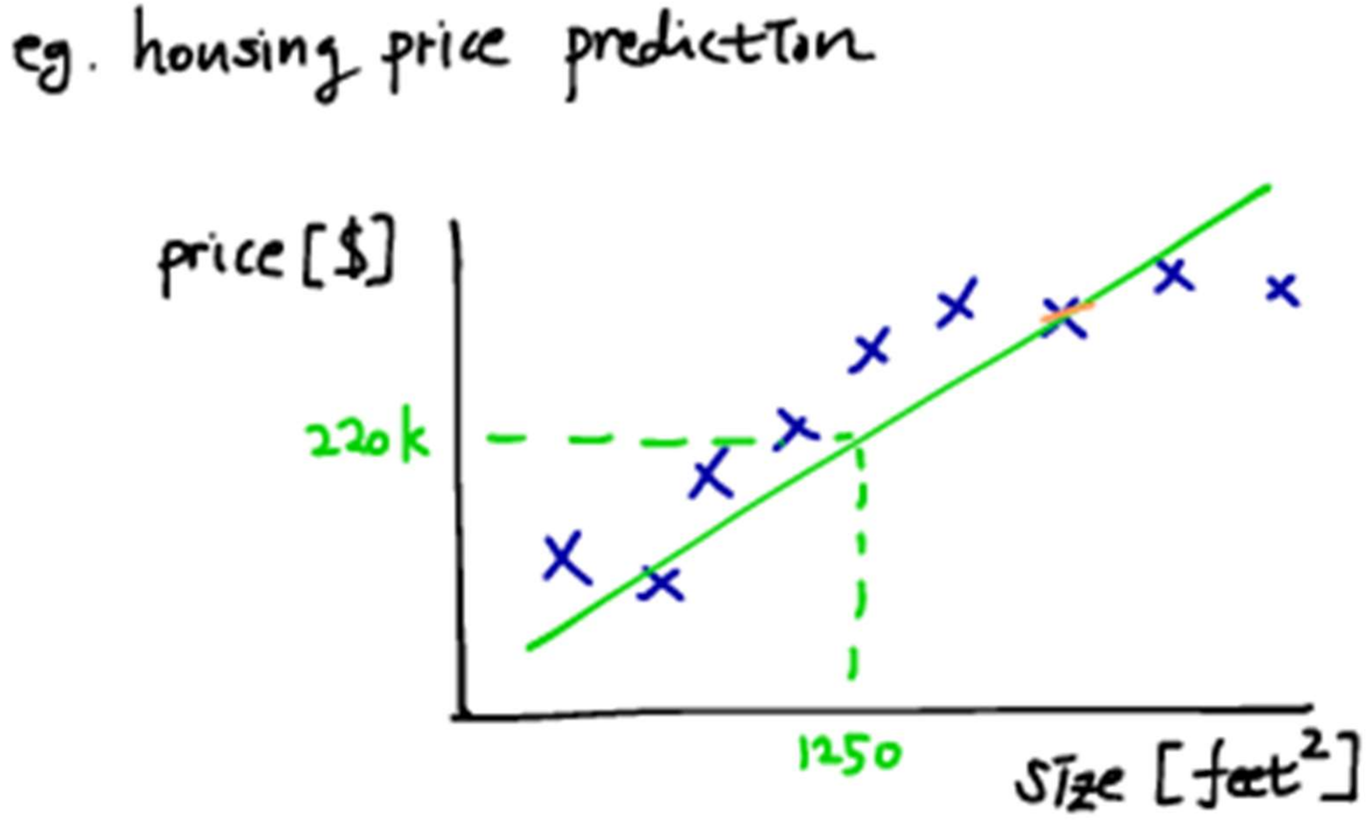



Deep neural network




출처: http://neuralnetworksanddeeplearning.com/chap5.html

# Linear Regression (선형 회기) (1)




https://wikidocs.net/4213

# Linear Regression (선형 회기) (2)

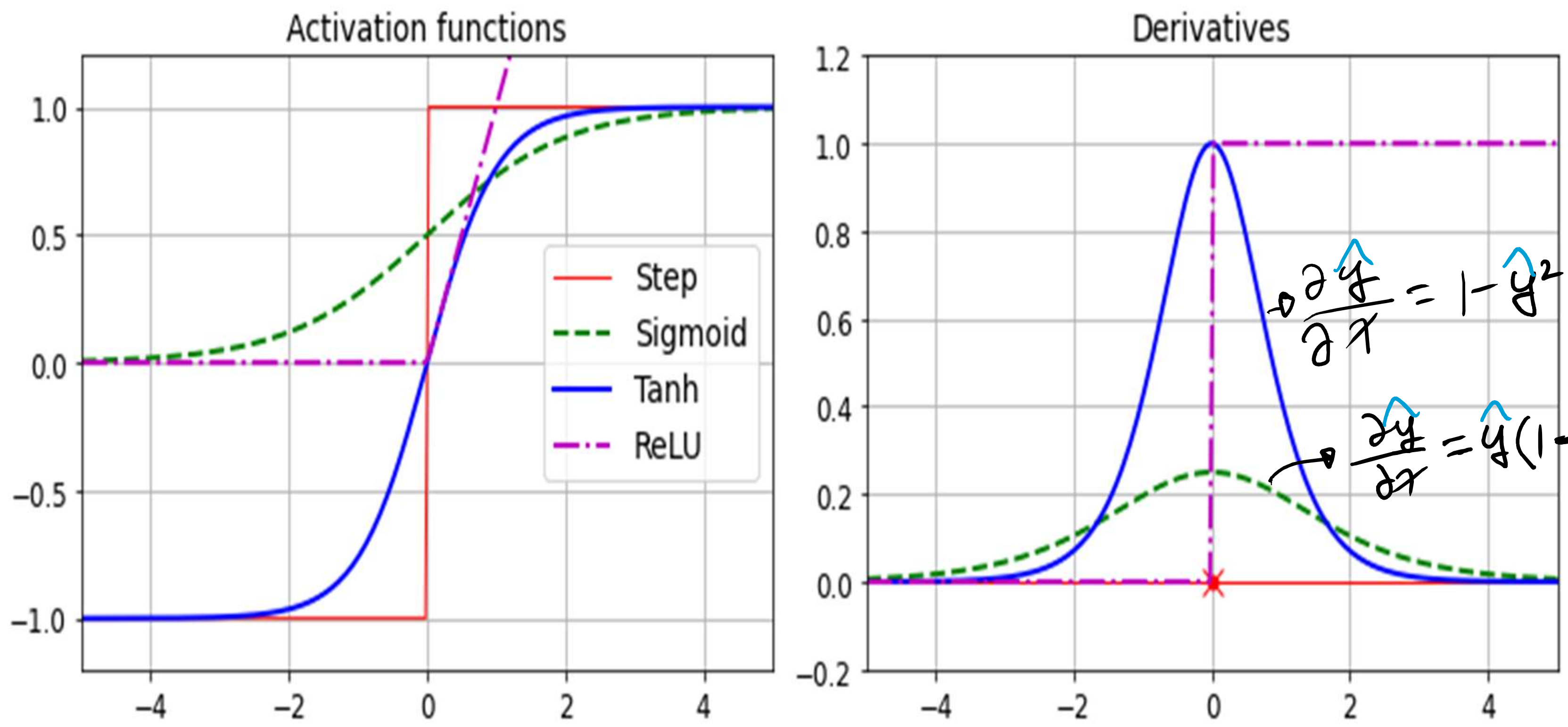
- Hypothesis function h:
  - $h_\theta(x)=\theta_0+\theta_1 x$
- Cost function J:
  - Mean-Squared-Error (MSE)
    - == LSE (least squared error)
  - 예측값과 실제값의 차이

$\hat{y}^{(i)}$ $y^{(i)}$

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

https://wikidocs.net/7635

# Activation Functions




$$\frac{\partial \hat{y}}{\partial x} = 1 - \hat{y}^2$$

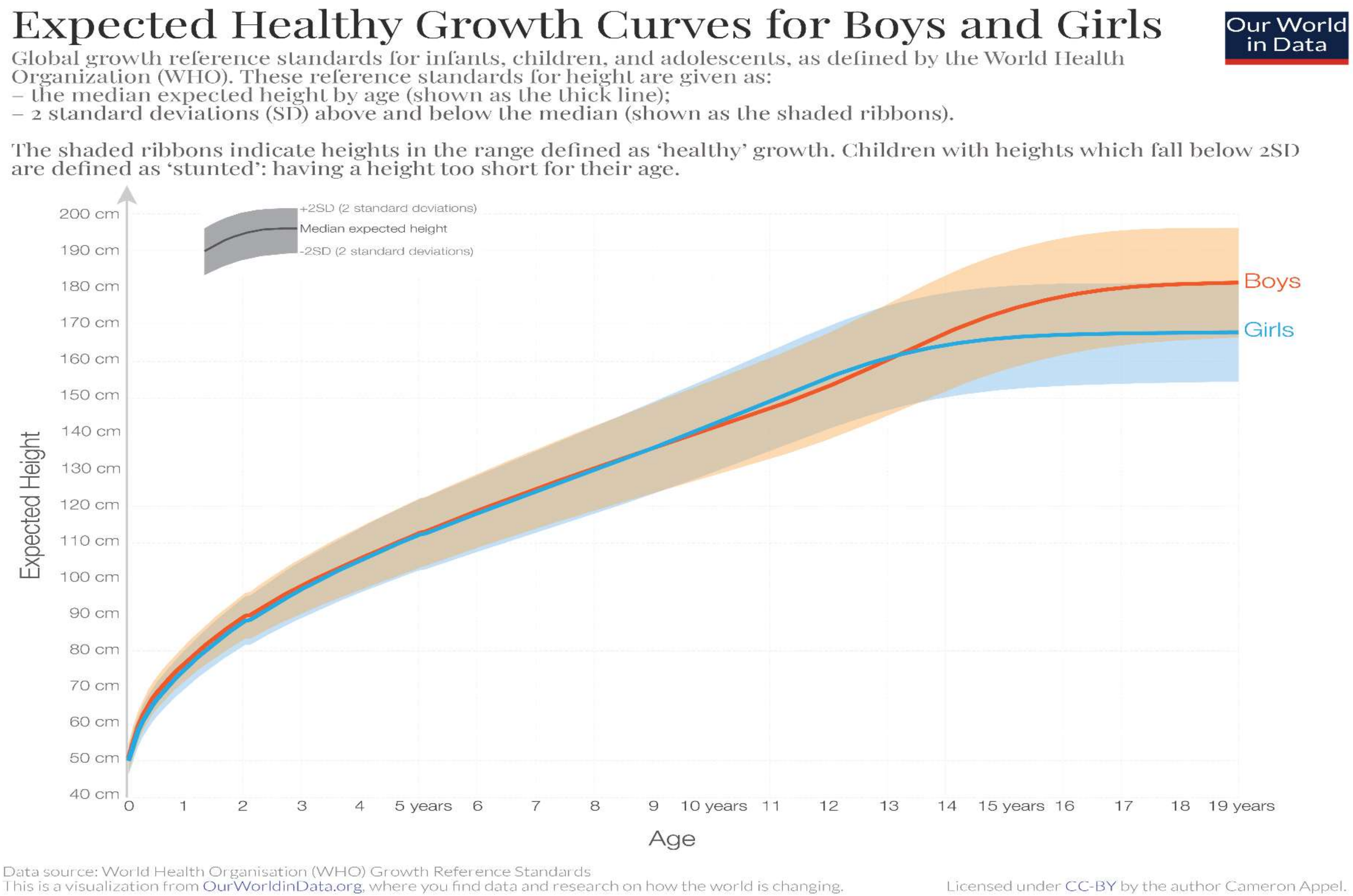
$$\frac{\partial \hat{y}}{\partial x} = \hat{y}(1-\hat{y})$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1}$$

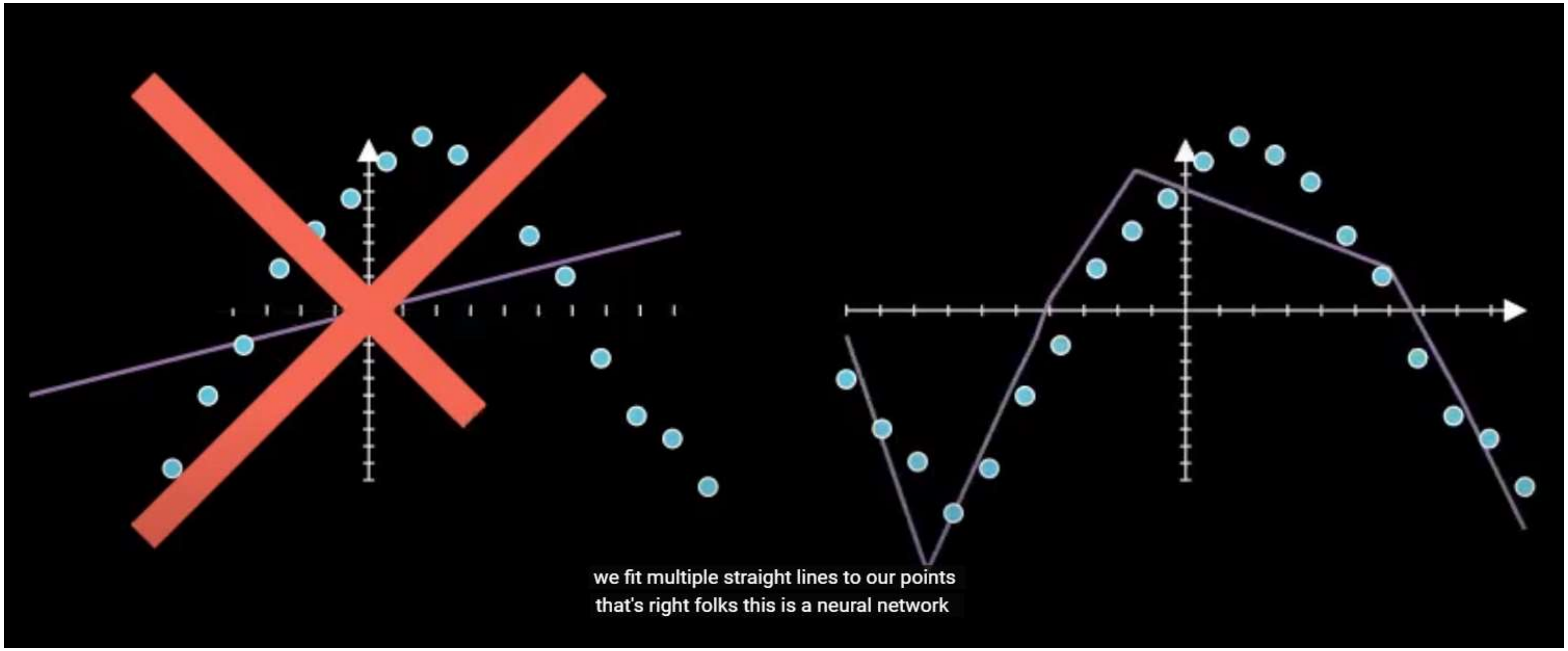
$$s(x) = \frac{1}{e^{-x}+1} = \frac{e^x}{1+e^x}$$

**Every activation function has an area of linearity and nonlinearity.**

# Nonlinearity Relation Example: Human Age to Height



# Nonlinearity



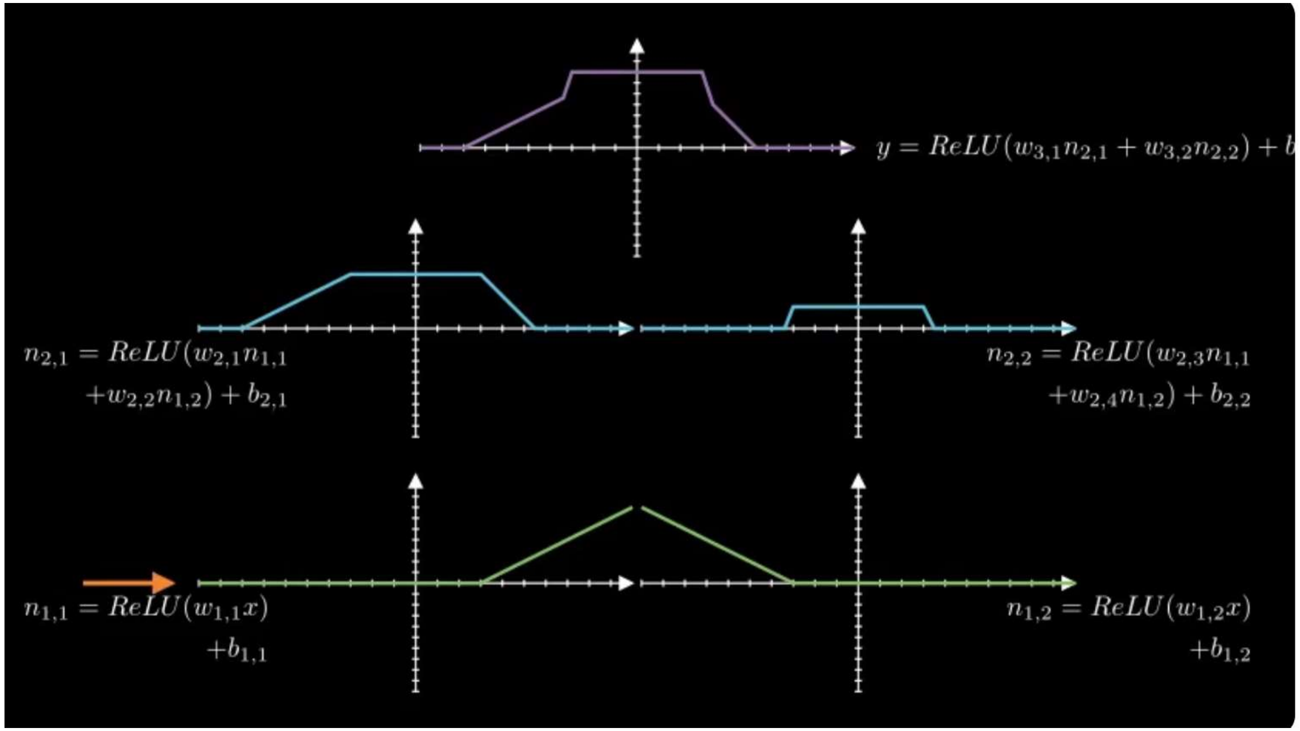

**https://www.youtube.com/watch?v=FBpPjjhJGhk**

# Nonlinearity

output

Two neurons $2^{nd}$ layer

Two neurons $1^{st}$ layer




L layers and N neurons in each layer can express N^L linear segments.

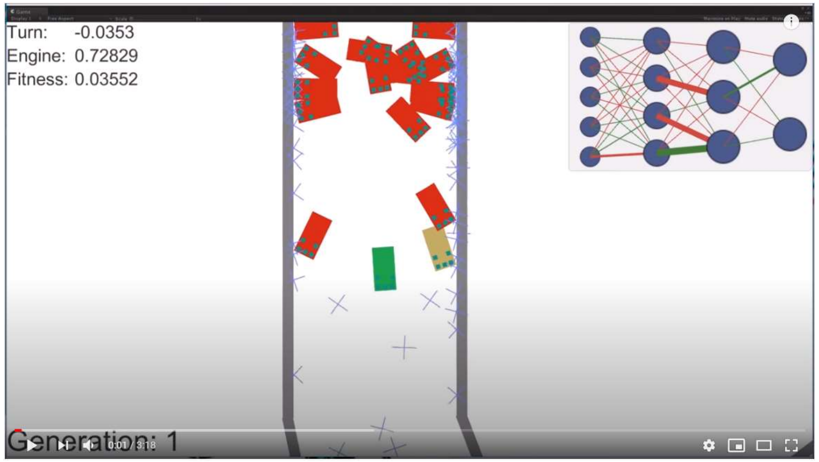
https://www.youtube.com/watch?v=FBpPjjhJGhk

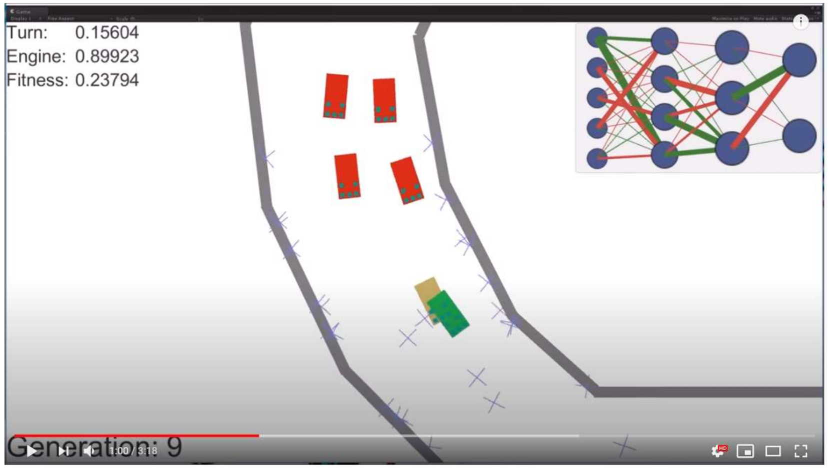
# Examples

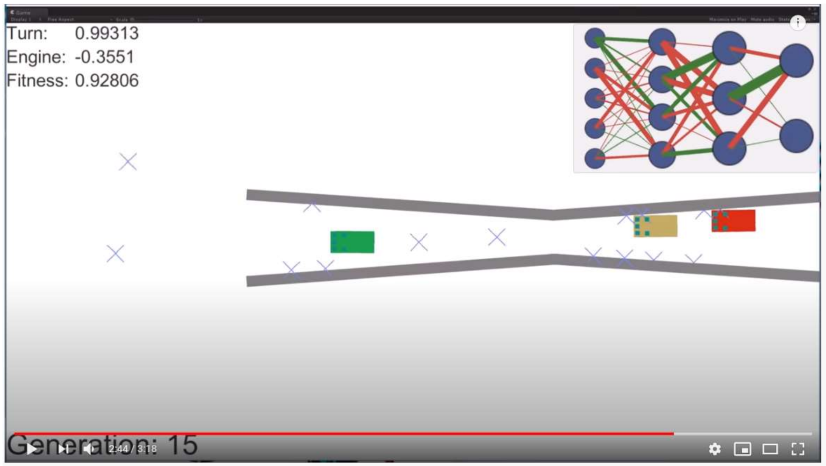
- Deep neural network (강화학습) 예 1:
  - https://www.youtube.com/watch?v=Aut32pR5PQA
- Deep neural network (강화학습) 예 2:
  - https://www.youtube.com/watch?time_continue=12&v=V1eYniJ0Rnk
- 비 기계학습 예:
  - http://www.etnews.com/20180114000011

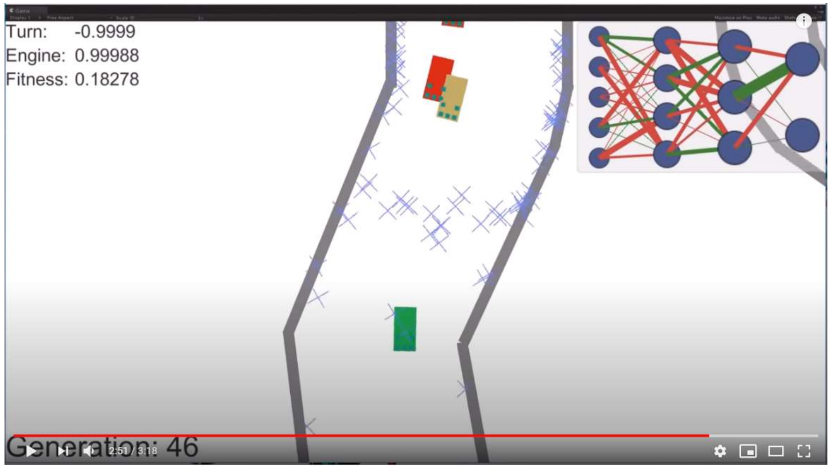
# 심층 강화학습 예: AI Car

- https://www.youtube.com/watch?v=Aut32pR5PQA



# Performance Metrics for Continuous Value Predictor

- Root Mean Square Error (RMSE), 평균 제곱근 오차
- Mean Absolute Error (MAE), 평균 절대 오차

$$RMSE(X,h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right)^2}$$

$$MAE(X,h) = \frac{1}{m}\sum_{i=1}^{m}\left|h(x^{(i)}) - y^{(i)}\right|$$

# Performance Metrics for Binary Classifier

- Confusion Matrix (오차 행렬)
- Type 1 and Type 2 Errors
- Accuracy (정확도)
- Precision (정밀도)
  - == Positive predicted value (양성 예측 값)
- Recall (재현도)
  - == Sensitivity (민감도)
  - == True Positive Rate (TPR, 진짜 양성 비율)
- Specificity (특이도)
  - == True Negative Rate (TNR, 진짜 음성 비율)
  - 1 – FPR (False Positive Rate, 가짜 양성 비율)
- F1 score

# Confusion Matrix (1)

| | | Predicted | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | Negative | True Negative (TN) | False Positive (FP) |
| | Positive | False Negative (FN) | True Positive (TP) |

# Confusion Matrix (2)

**Is it ‘A’?**

| | | Predicted | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | Negative | F D C True Negative (TN) D | B False Positive (FP) |
| | Positive | A False Negative (FN) A | A A True Positive (TP) A |

# Type 1 and 2 Errors

- https://en.wikipedia.org/wiki/F-score

| | | Predicted: Negative | Predicted: Positive |
|---|---|---|---|
| Actual | Negative | True Negative (TN) | False Positive (FP)<br>**Type I Error** |
| | Positive | False Negative (FN)<br>**Type II Error** | True Positive (TP) |

# Accuracy, Precision, Recall, and F1

Is it 'A'?

| Is it 'A'? | | Predicted: Negative | Predicted: Positive |
|---|---|---|---|
| Actual | Negative | F D C True Negative (TN) D | B False Positive (FP) |
| | Positive | A False Negative (FN) A | A A True Positive (TP) A |

Accuracy = (TP + TN) / (TP + TN + FP + FN) $= \frac{7}{10} = 0.7$

Precision = TP / (TP + FP) $= \frac{3}{4} = 0.75$

Specificity = TN / (FP + TN)

Recall = TP / (TP + FN) $= \frac{3}{5} = 0.6$

F1 = Harmonic average of Precision and Recall = 2/ (1/Precision + 1/Recall)

$$= \frac{2}{\frac{4}{3} + \frac{5}{3}} = \frac{2}{3} = 0.67$$

# Receiver Operating Characteristic (ROC)

- FPR vs. TPR

TNR = Specificity (특이도)

FPR

| | | Predicted | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | Negative | True Negative (TN) | False Positive (FP) |
| | Positive | False Negative (FN) | True Positive (TP) |

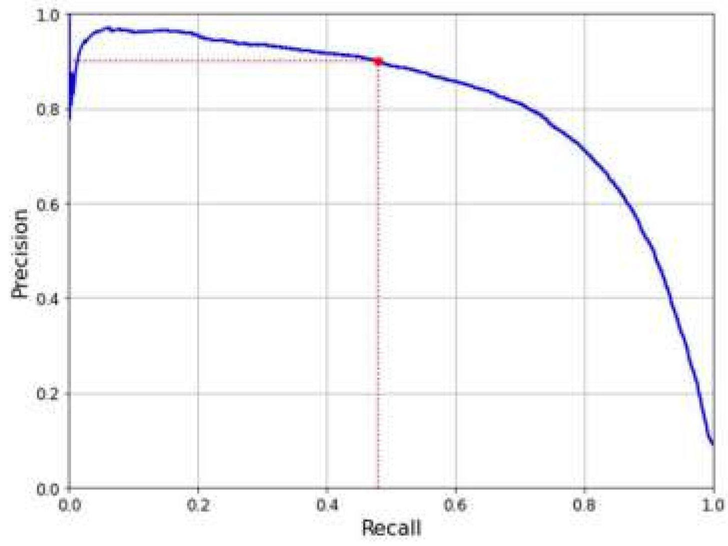
FNR

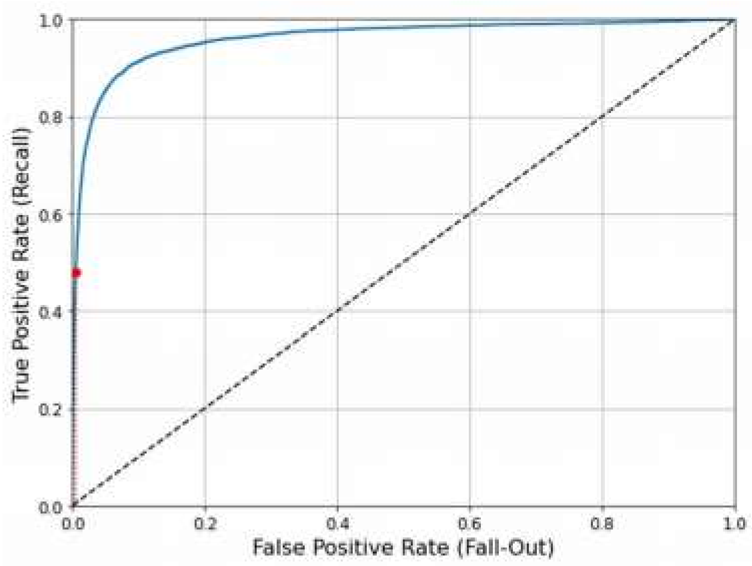
Precision

TPR
= Recall
= Sensitivity

# PR (Precision/Recall) vs. ROC Curves

- PR is preferred when
  - FP is more important than FN.
  - Positive class is rare.
- Area under the curve (AUC)
  - A perfect classifier's AUC is 1.
  - A random classifier's AUC is 0.5.



# Confusion Matrix for Multi-Class

- Total TP = (7+2+1) = 10
- Total FP = (8+9)+(1+3)+(3+2) = 26
- Total FN = (1+3)+(8+2)+(9+3) = 26

  == Total FP

Hence,

- Accuracy = Precision = Recall = 10/36 = 0.28

- Commonly
  - Accuracy: dataset-wide
  - Precision and recall: class-specific

| Class | Precision | Recall |
|---|---|---|
| Apple | 0.64 | 0.29 |
| Orange | 0.17 | 0.33 |
| Mango | 0.08 | 0.17 |

True Class

| Predicted Class | Apple | Orange | Mango |
|---|---|---|---|
| Apple | 7 | 8 | 9 |
| Orange | 1 | 2 | 3 |
| Mango | 3 | 2 | 1 |

https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826