

인공지능이란

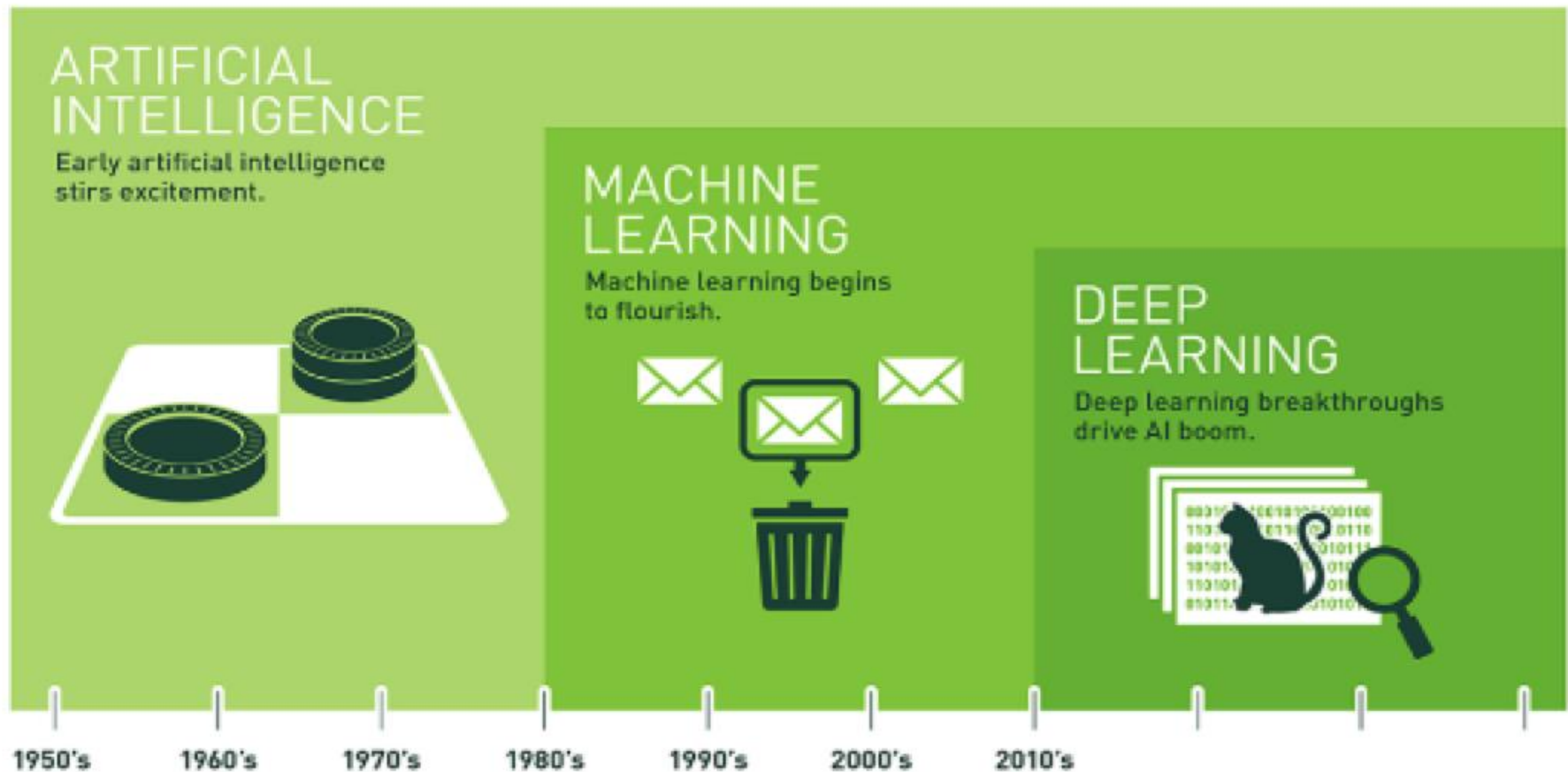
인공지능(AI; Artificial Intelligence)

- 사람의 손이 아닌 기계가 알아서 하면 인공지능이다.
- 전문가의 지식을 하드코딩 할 수도 있고(전문가 시스템)
- 데이터에서 로직을 찾을 수도 있다.(머신러닝)

머신러닝(ML; Machine Learning)

- 인공지능의 한 분야
- 데이터에서 가치를 찾아내는 것
- 아주 다양한 방법이 있다.
 - SVM, 의사결정트리, Random Forest, Bayesian, K-Means Clustering, K-NN, Neural Network

AI, ML, DL



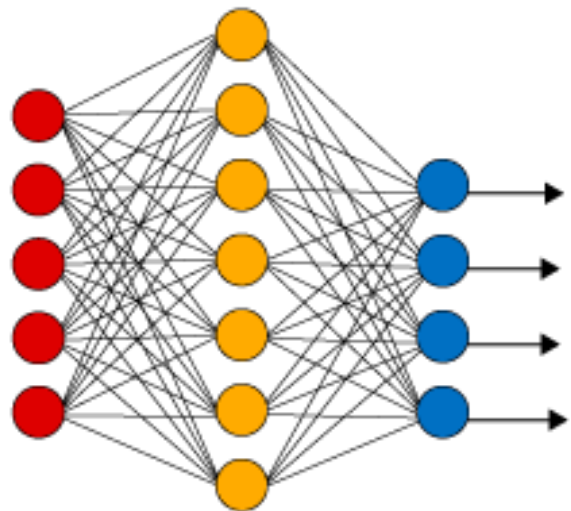
AI, ML, DL

- 인간이 고안한 알고리즘이건 기계가 학습한 알고리즘 이건, 기계가 스스로 처리하면 AI
- 기계가 학습하는 경우가 ML
- 그중 신경망을 사용하는 것이 DL
- 서로 다르지만, 그냥 $AI = ML = DL$ 이라 부른다.

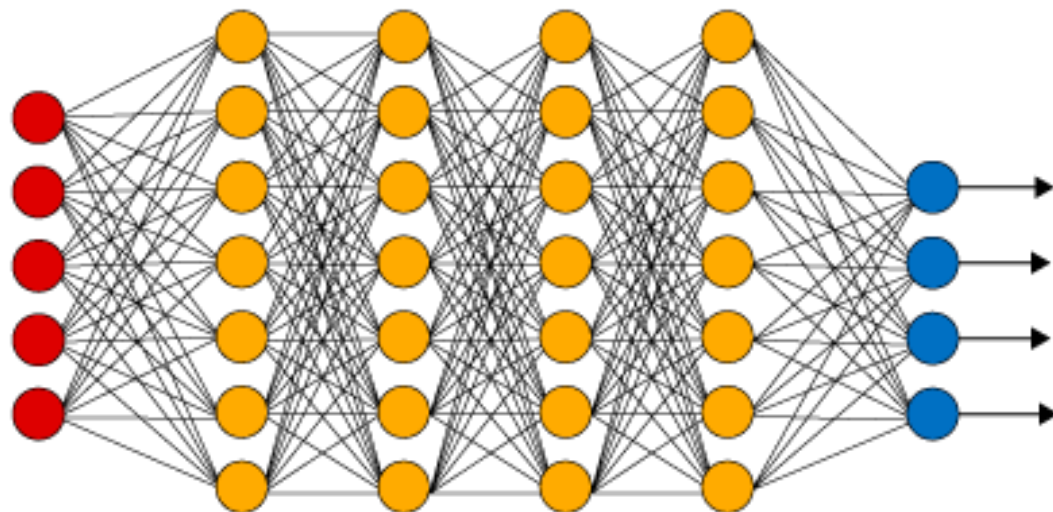
딥러닝(DL; Deep Learning)

- 신경망(NN; Neural Network)을 사용한 머신러닝 방법
- 쌓는 은닉층이 많아서(deep) DNN(Deep NN)이라 부른다

Simple Neural Network



Deep Learning Neural Network

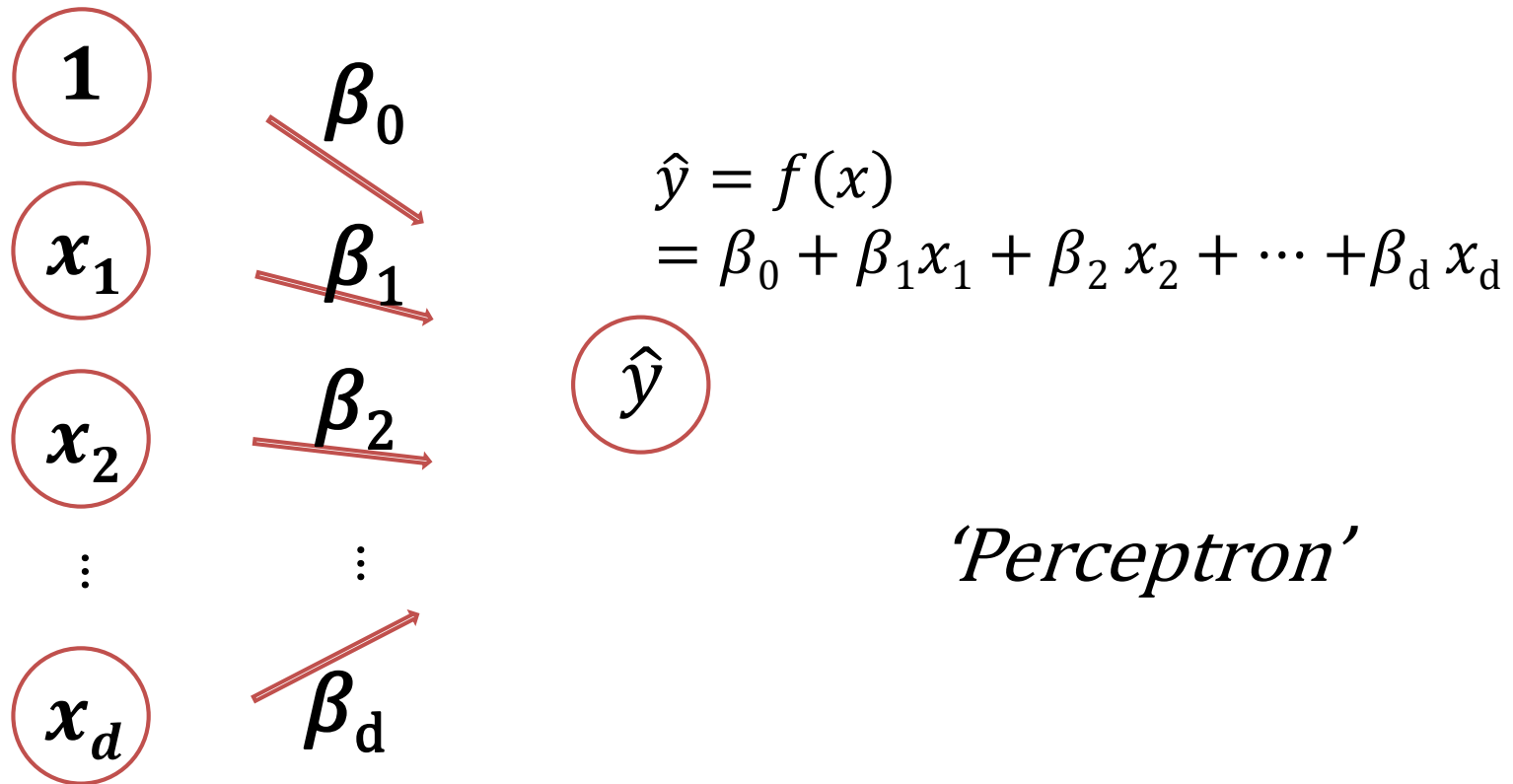


● Input Layer

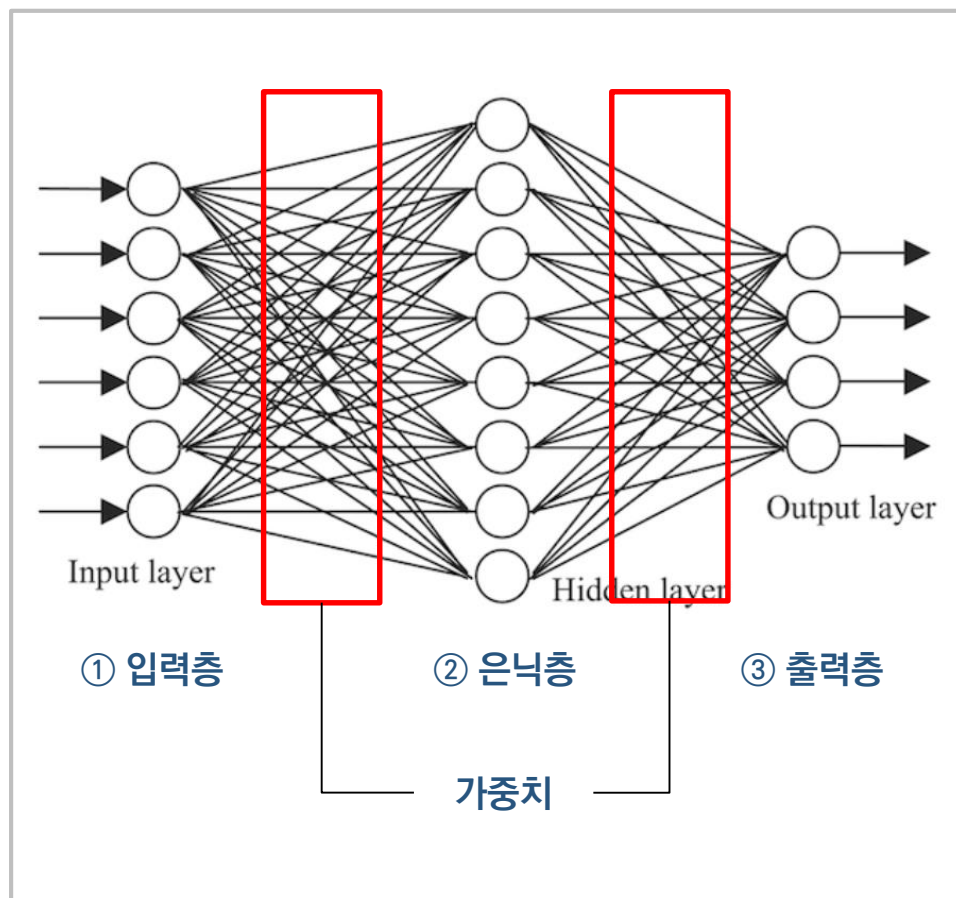
● Hidden Layer

● Output Layer

- 특징(feature)들의 선형 결합으로 y 의 값을 예측해보자.
- 1은 입력변수에 상관없이 존재하는 편향을 반영

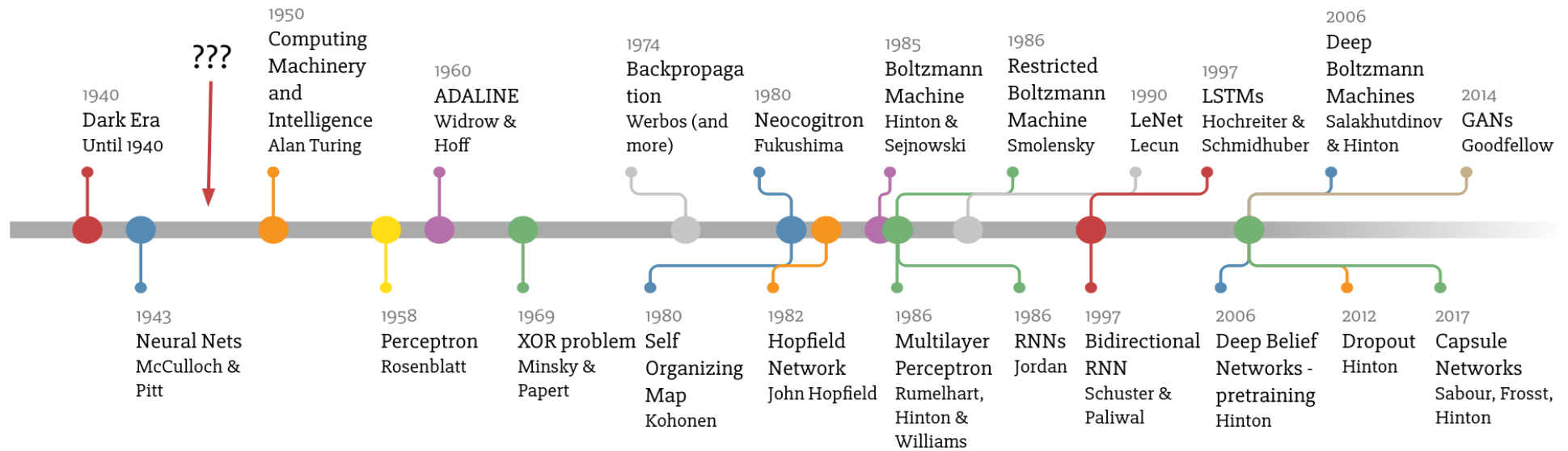


신경망(Neural Network)의 구조



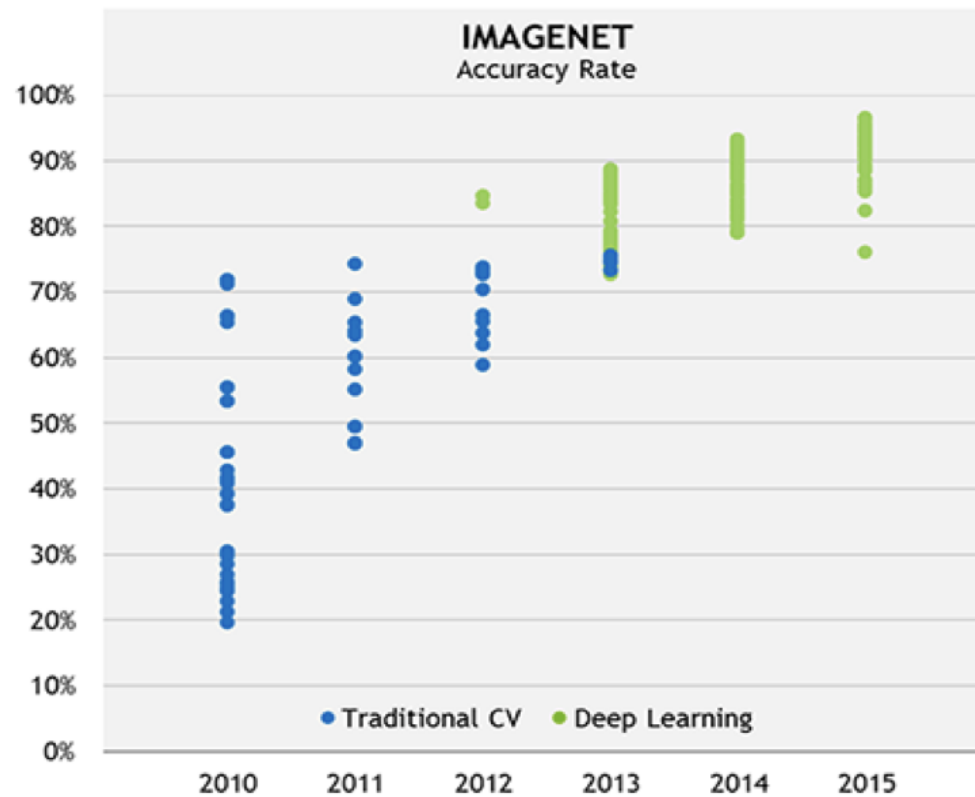
- ① 입력층 : 각 **설명 변수에** 대응되는 노드
(설명 변수의 특성 개수와 일치)
- ② 은닉층 : 입력층으로부터 전달되는 변수 값들의 **선형 결합을 비선형 함수로 처리**하여 출력층에 전달
- ③ 출력층 : **반응 변수에** 대응되는 노드
(반응 변수의 특성 개수와 일치)

Deep Learning Timeline

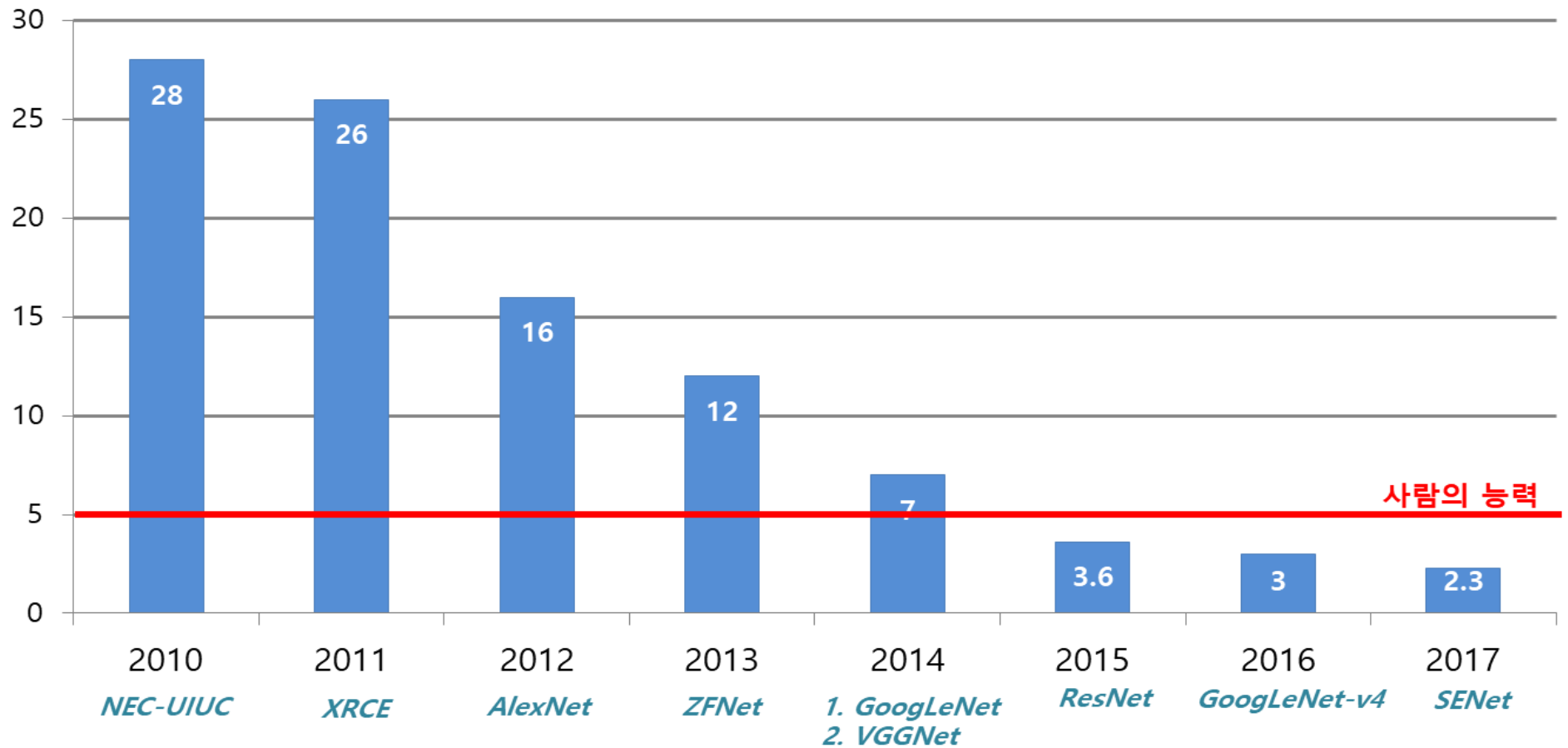


Made by Favio Vázquez

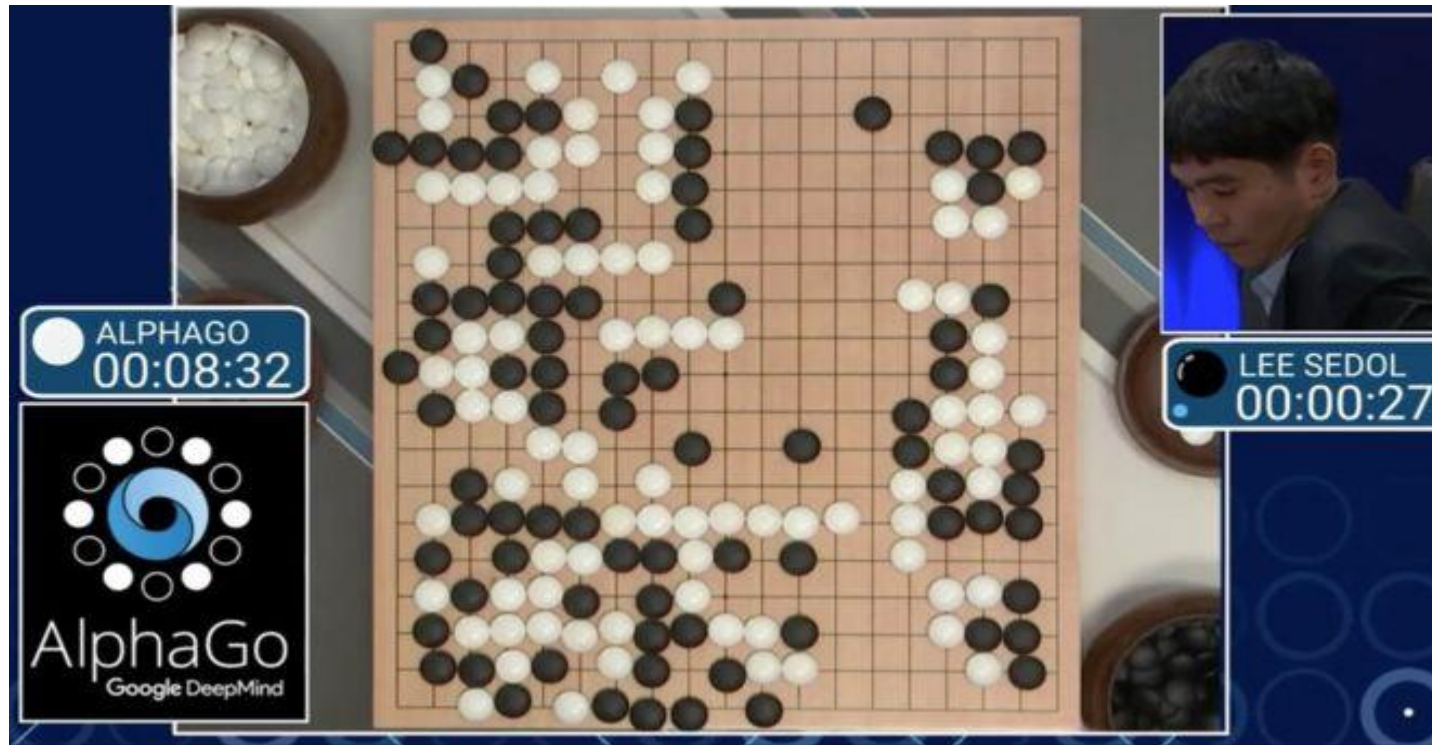
2012년 AI 부활



우승 알고리즘의 분류 에러율(%)



2016년 화려한 복귀



심층신경망(DNN)과 신경망(NN)

- DNN과 NN은 별 차이 없다.
- NN: AI에 대한 엄청난 기대와 그 만큼의 실망.
- 최근 뚜렷한 성과를 보이면서 다시 큰 관심.
- DNN: 실망했던 용어를 대신하여 붐을 일으키기 위한 용어.

딥러닝 장단점

- 장점 : 대상 함수의 내부를 몰라도 된다.
- 단점 : 비싸다
 - 많은 데이터, 많은 연산량

ML과 DL의 선택

- 기존의 방법으로 이미 풀린 문제는 ML
- 기존의 방법으로 못풀었는데 데이터가 있으면 DL
 - 바둑, 얼굴인식, 물체인식, 음성인식, 번역

딥러닝 실제 코드

```
model = build_model() // 모델 구성
```

```
for i in range(1000): // 모든 데이터에 대하여 1000번 반복  
    model.fit(datas)
```

```
model.save('my_model.h5') // 저장. 파일 크기는 보통 100메가  
단위
```


딥러닝 트렌드

- 2대 적용 분야 : 자연어, 영상
- 일부 작업은 이미 안정화 단계에 있다.
 - 영상분류, 영상인식
 - Keras의 배포본에 포함

```
model = keras.vgg16.VGG16()  
predicted = model.predict(image_data)
```

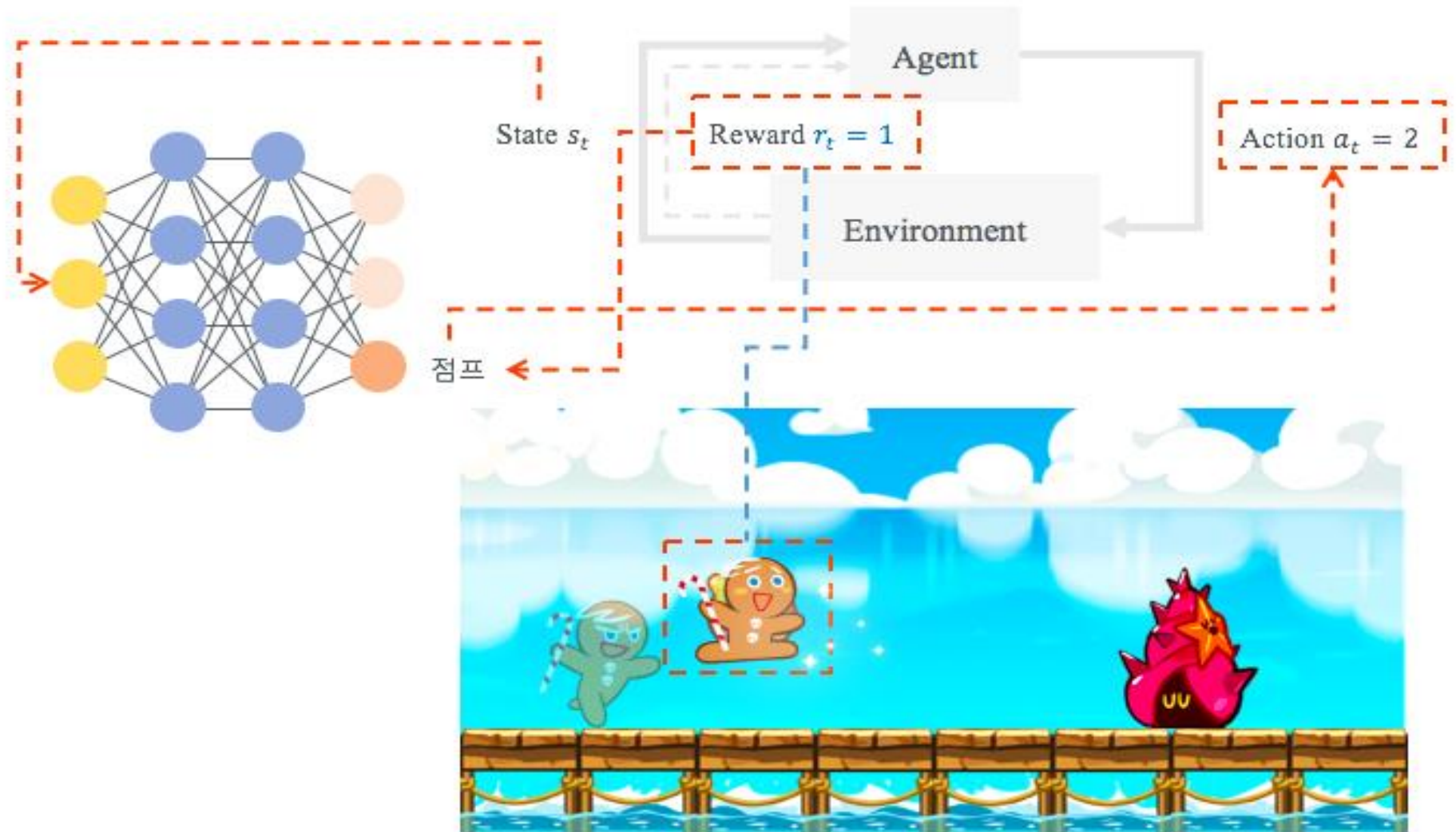
딥러닝의 큰 단점

- 입출력 데이터 쌍을 구하기 어렵다.
- 특히 출력 데이터. 레이블링 데이터(labeling data)

비지도 학습, 강화 학습

- 레이블링 데이터 문제를 해결하기 위한.
- 비지도 학습 : 모델 구조를 통해 레이블링 데이터 없이. GAN
- 강화 학습 : 환경과 동적으로 연동하여 레이블링 데이터를 취득.

비지도 학습, 강화 학습



개발도구

- 딥러닝 구현을 위해 python을 많이 쓰는 추세다 보니, framework 또한 python에 어울리는 tensorflow나 pytorch 등이 많이 쓰인다.



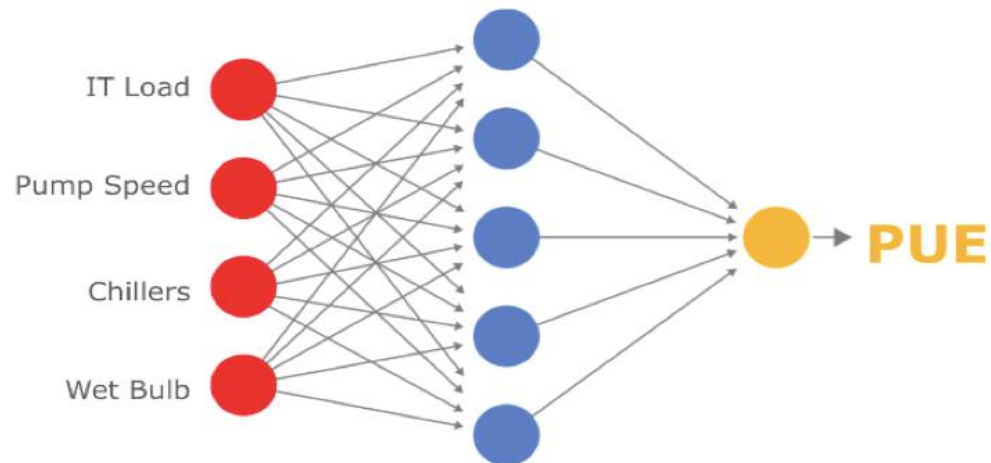
VS

PYTORCH

구글데이터센터 예시

DNN 적용

- 제어값을 입력으로 하고, PUE*를 출력으로 하여 학습
- 실제 데이터 사용

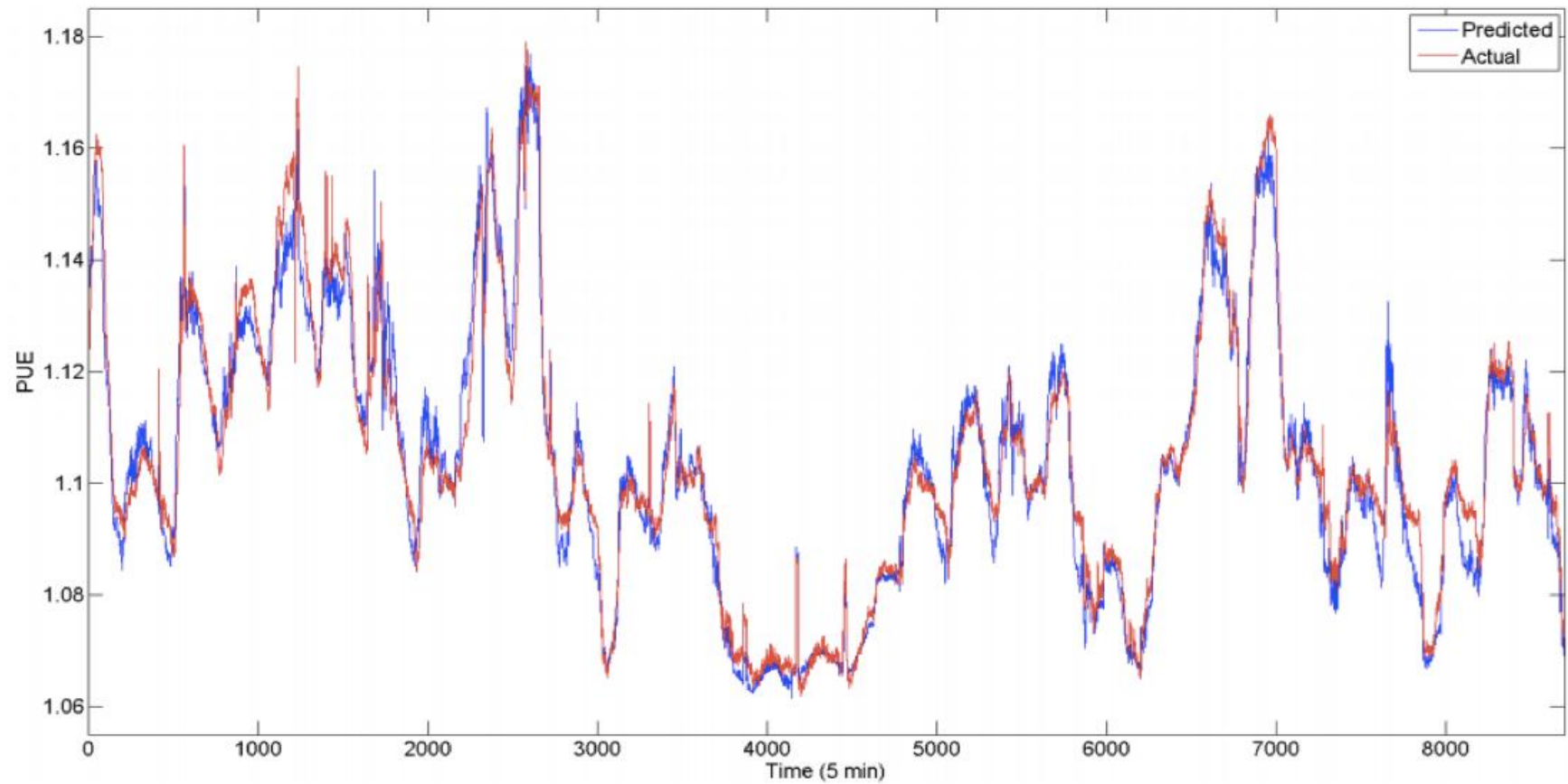


*PUE(Power Usage Effectiveness) : 에너지 사용 효율도

입력

1. Total server IT load [kW]
2. Total Campus Core Network Room (CCNR) IT load [kW]
3. Total number of process water pumps (PWP) running
4. Mean PWP variable frequency drive (VFD) speed [%]
5. Total number of condenser water pumps (CWP) running
6. Mean CWP variable frequency drive (VFD) speed [%]
7. Total number of cooling towers running
8. Mean cooling tower leaving water temperature (LWT) setpoint [F]
9. Total number of chillers running
10. Total number of drycoolers running
11. Total number of chilled water injection pumps running
12. Mean chilled water injection pump setpoint temperature [F]
13. Mean heat exchanger approach temperature [F]
14. Outside air wet bulb (WB) temperature [F]
15. Outside air dry bulb (DB) temperature [F]
16. Outside air enthalpy [kJ/kg]
17. Outside air relative humidity (RH) [%]
18. Outdoor wind speed [mph]
19. Outdoor wind direction [deg]

예측 결과



학습된 DNN

- 제어값과 PUE의 관계를 학습
- 임의의 제어값에 대한 PUE를 알 수 있다.
- 학습된 DNN은 시뮬레이터로 사용할 수 있다.

시뮬레이션이 가능하면

- 임의의 제어값에 대한 PUE를 미리 알 수 있다.
- 다양한 입력에 대한 시뮬레이션으로, 최선의 PUE를 찾을 수 있다.

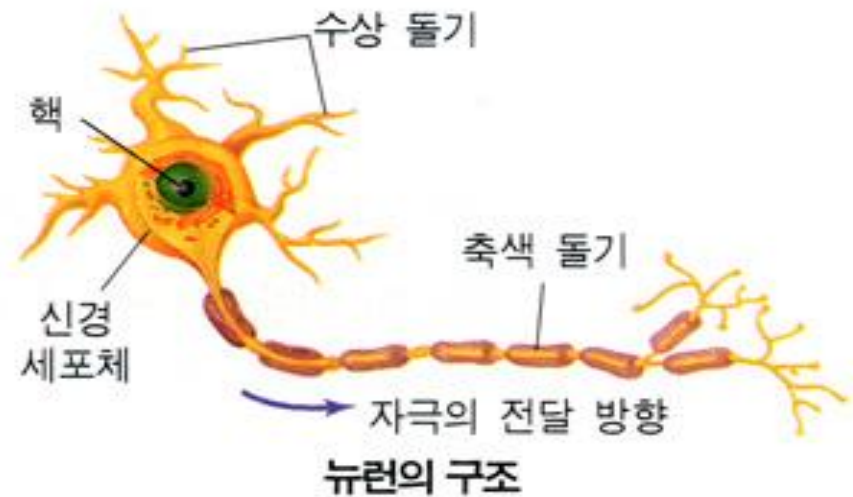
신경망

신경세포

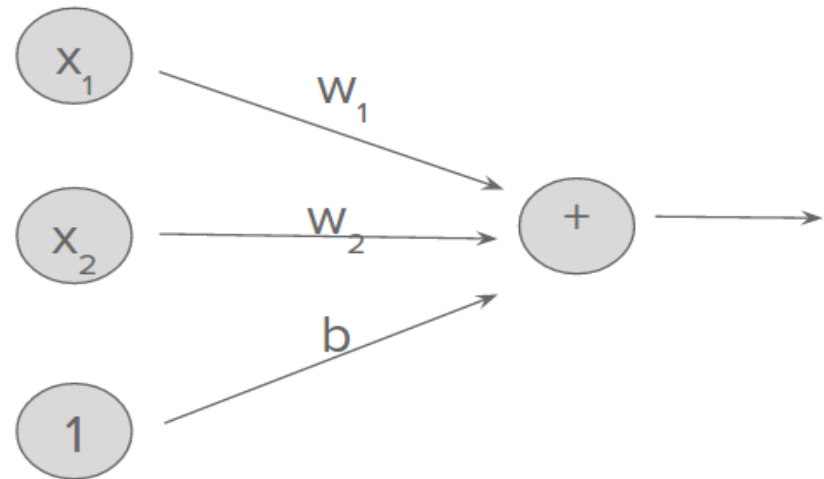
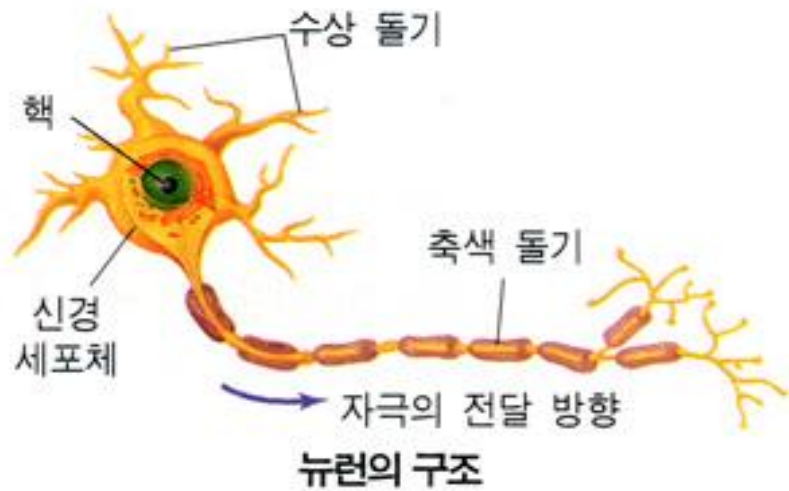
여러개의 수상돌기에서 자극이 합해져서

그 값이 어느 값 이상일 경우

축색돌기로 자극을 발생시킨다.

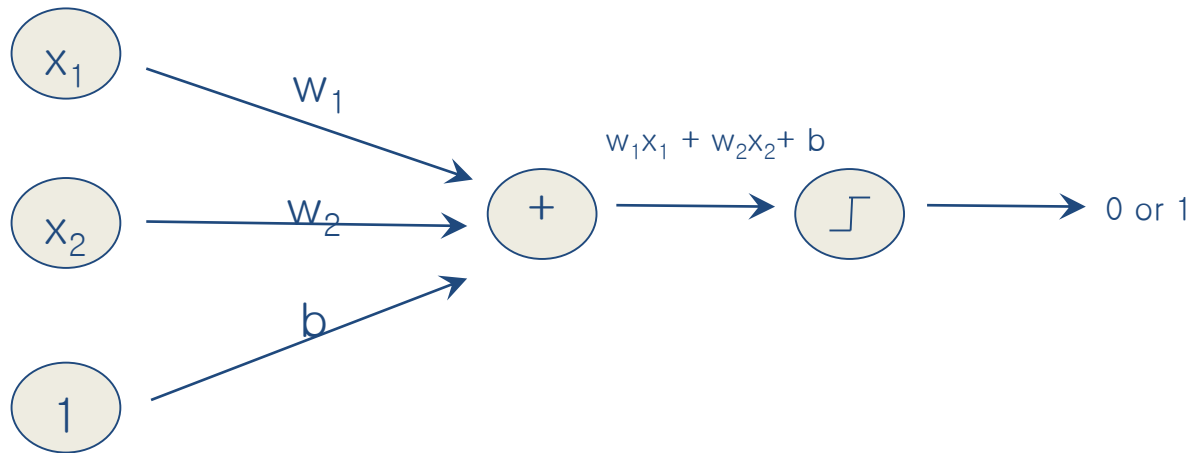


신경세포와 유사



퍼셉트론(Perceptron)

선형분리의 문제를 학습할 수 있다.

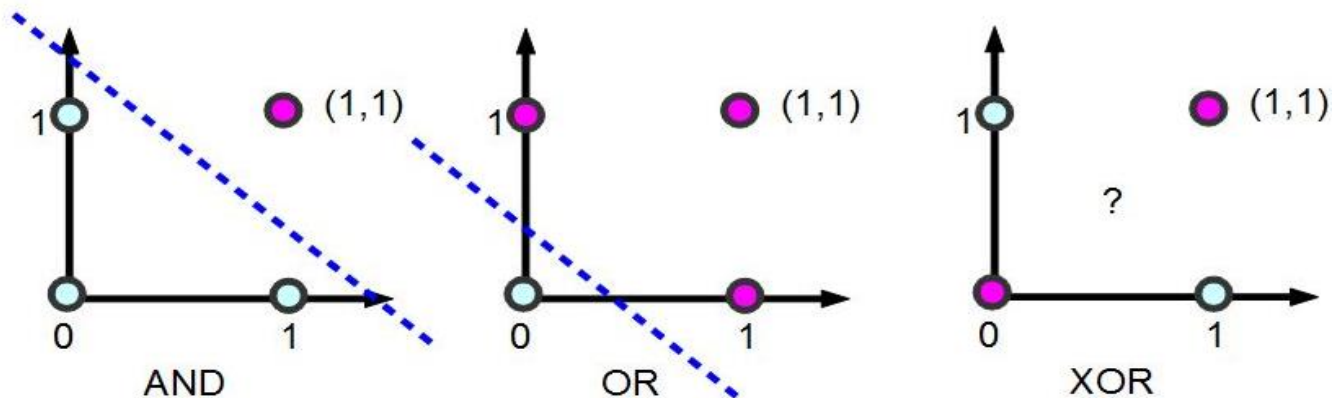


퍼셉트론의 한계

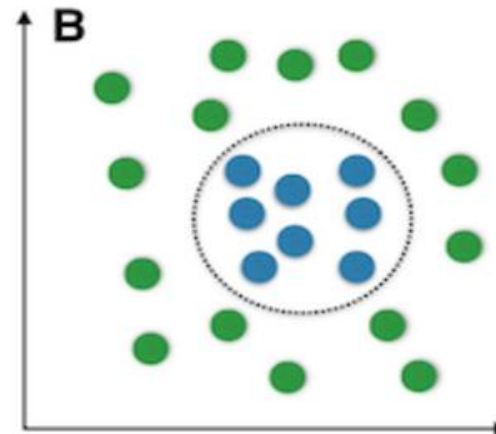
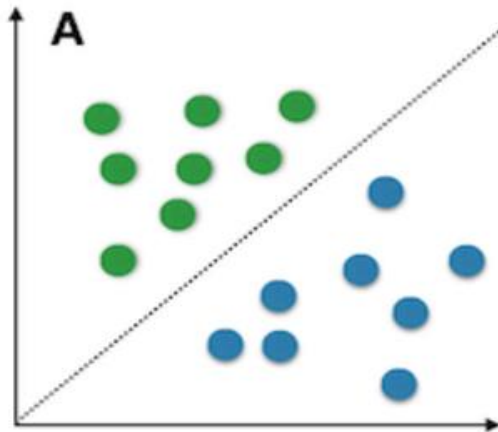
하지만 간단한 XOR도 못한다.

선형분리가 불가능한 것은 풀지 못한다.

XOR는 선형분리로 풀지 못하는 문제이다.

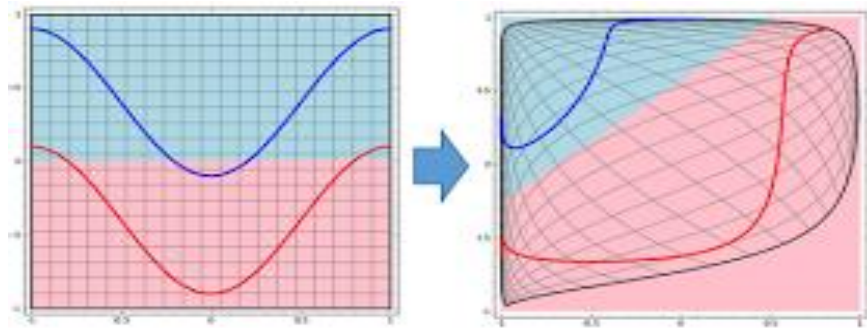


선형 분리 여부



선형 분리 불가 문제의 해결법

- 입력 차원을 늘린다.(사과 예)
- 입력을 비선형 변환하여 선형분리 가능하도록 한다.

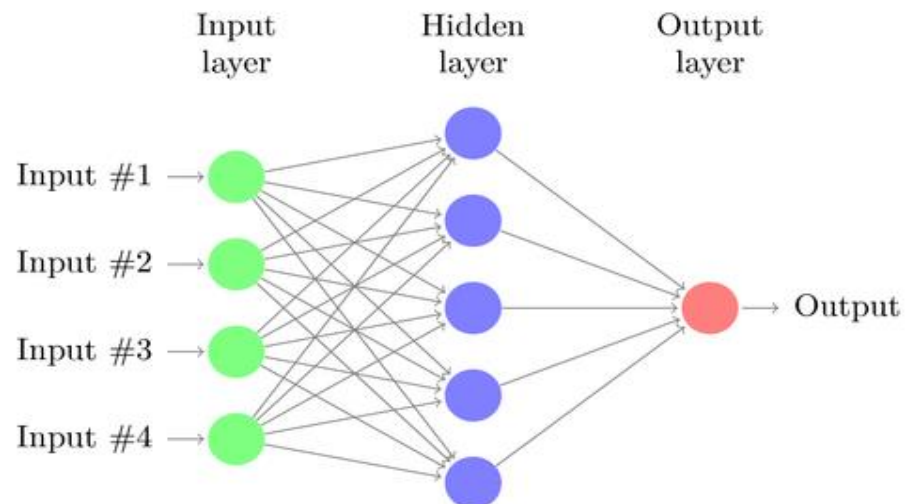


- 혹은 MLP (다음 페이지)

MLP(Multi Layer Perceptron)

입력과 출력 사이에 층이 더 있다.


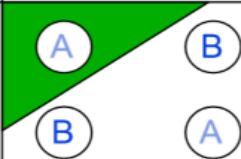
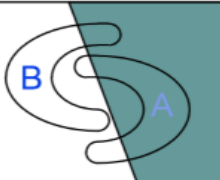
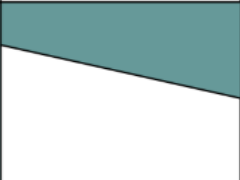
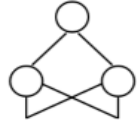
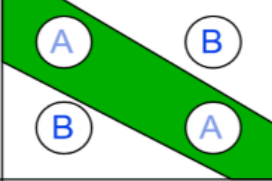
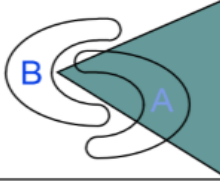

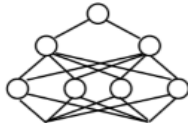
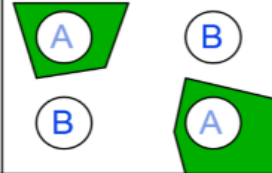

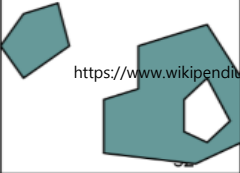
개별 perceptron의 결과를 다음 층의 입력으로 사용하고 결과적으로 선형 분리의 제약을 극복한다.



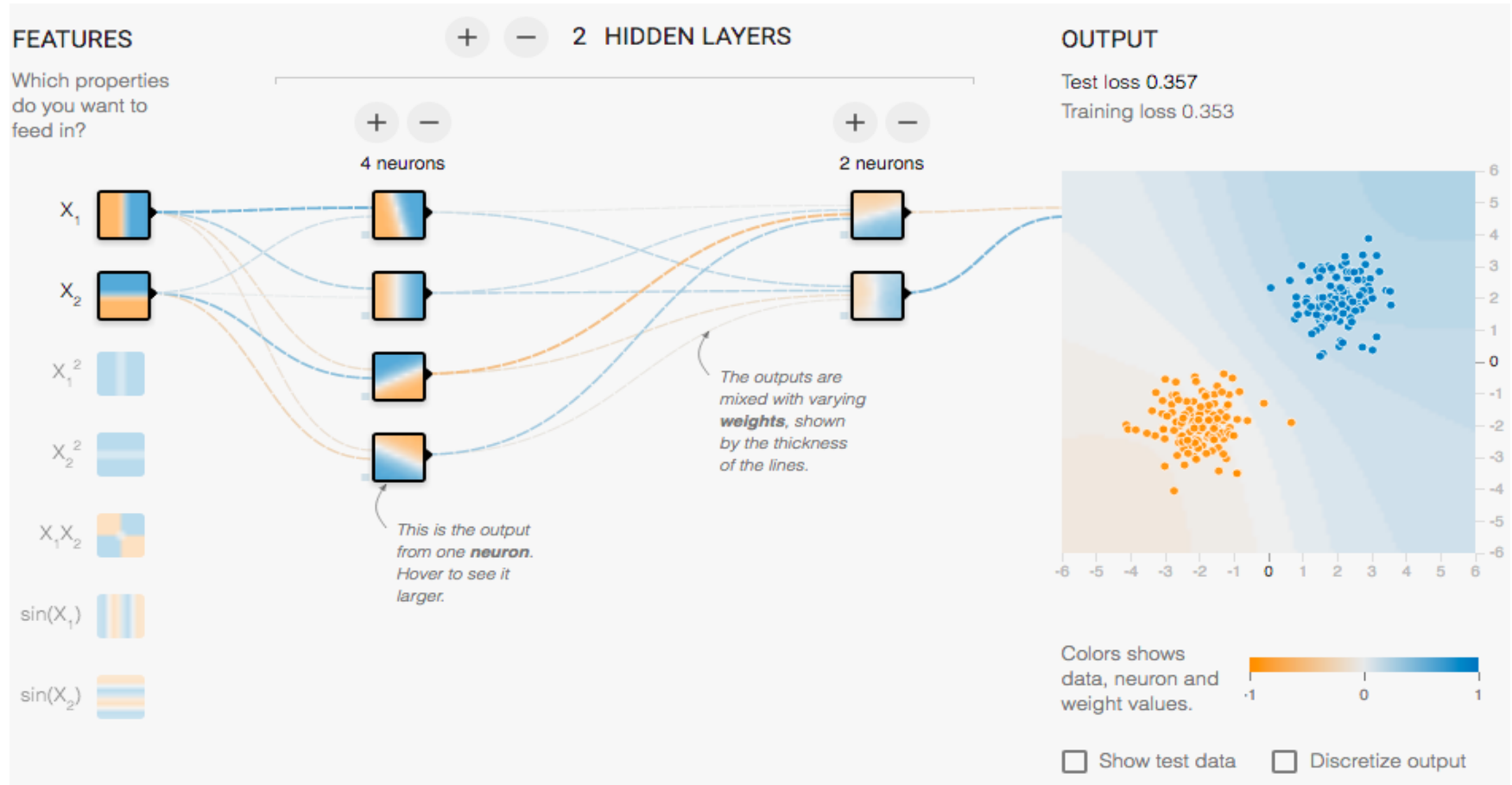
퍼셉트론의 능력

1개의 퍼셉트론은 1개의 선형 분리를 할 수 있다.

퍼셉트론의 결과를 다른 퍼셉트론의 입력으로 하면 여러개의 선으로 분리를 할 수 있다.

Structure	Types of Decision Regions	Exclusive-OR Problem	Classes with Meshed regions	Most General Region Shapes
Single-Layer 	Half Plane Bounded By Hyperplane			
Two-Layer 	Convex Open Or Closed Regions			
Three-Layer 	Arbitrary (Complexity Limited by No. of Nodes)			

온라인 시뮬레이션



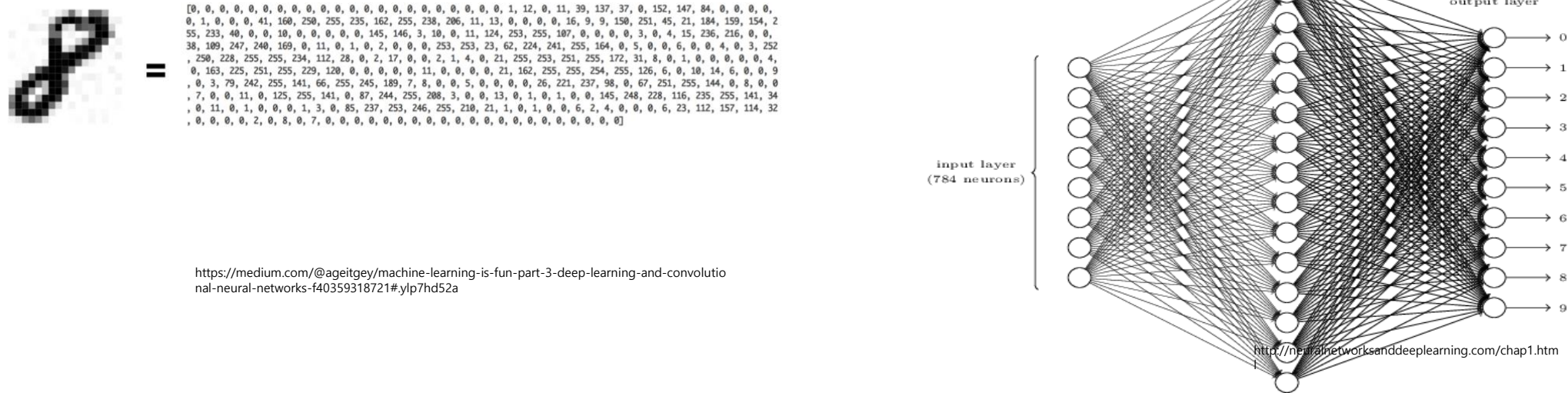
<http://playground.tensorflow.org>

숫자 인식의 예

이미지를 구성하는 pixel의 각 값으로 구성된 입력벡터를 NN의 입력으로 한다.

그리고 학습시 해당 출력 노드만 1로 하고, 나머지는 0으로 학습.

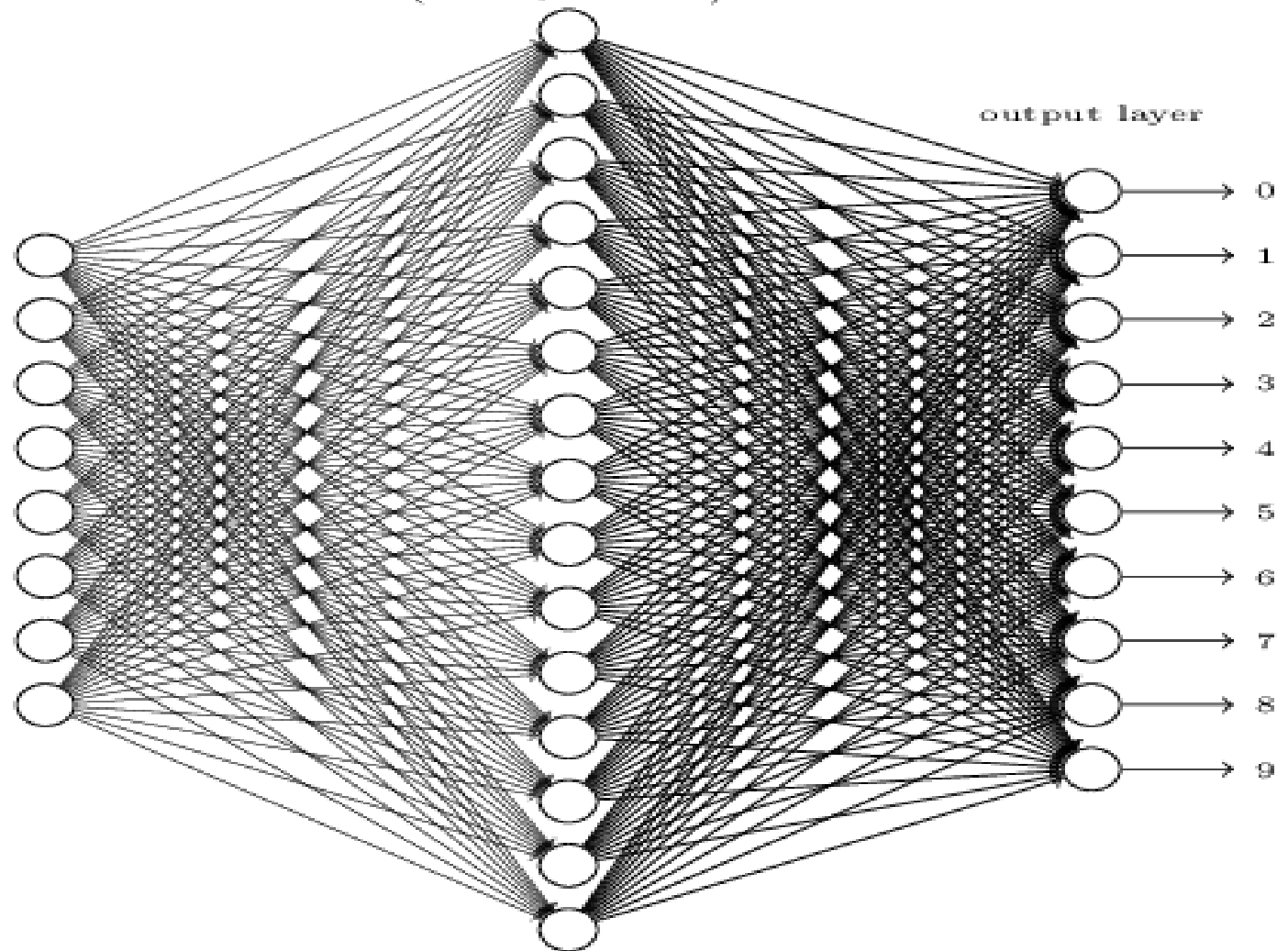
test 시에는 출력 노드중 최대 값을 가진 것으로 인식.



hidden layer
($n = 15$ neurons)

output layer

input layer
(784 neurons)



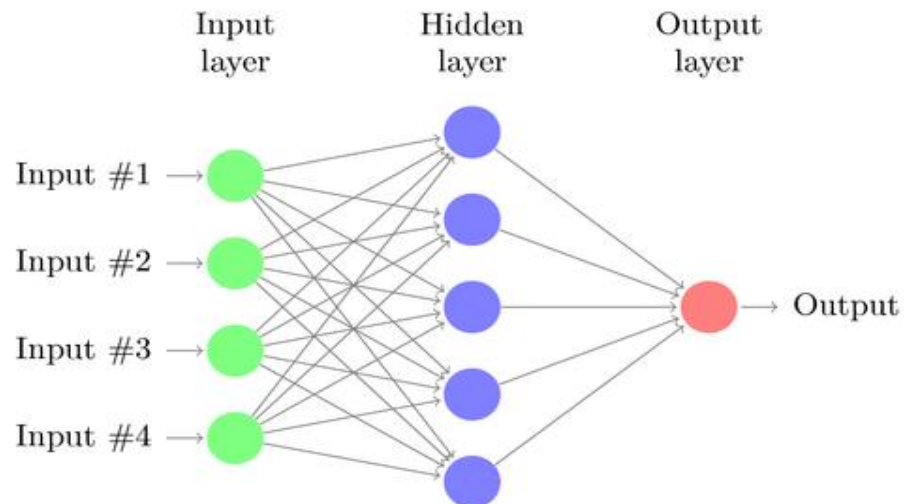
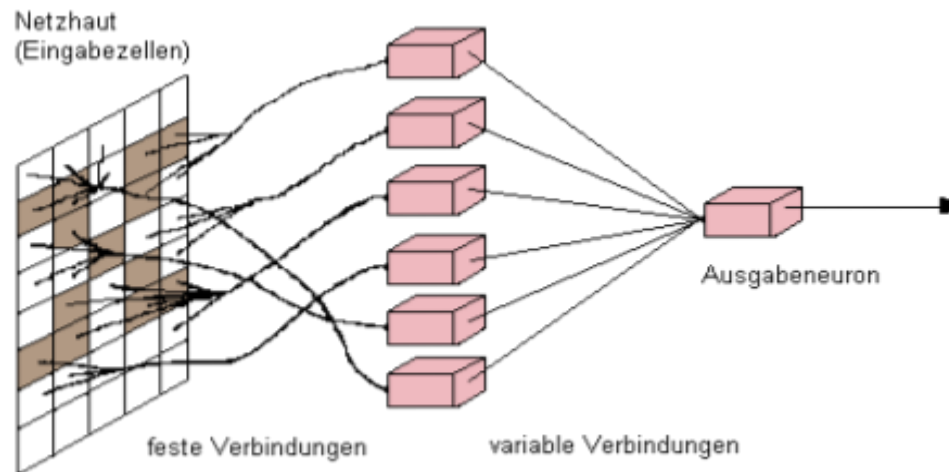
DNN

목차

- 인공지능이란. AI, ML, DL
- 인공지능의 흐름. imagenet, 겨울
- 인공지능 분야: ComputerVision, 자연어
- 해결할 문제유형: 회귀, 분류
- 데이터타입에 따라: 숫자형, 범주형, 이미지, 음성

입력과 출력 사이에 층이 더 있다.

개별 perceptron의 결과를 다음 층의 입력으로 사용하고 결과적으로 선형 분리의 제약을 극복한다.



<https://m.blog.naver.com/cyclonedhk/22071858248>
3

함수 근사화

DNN의 능력

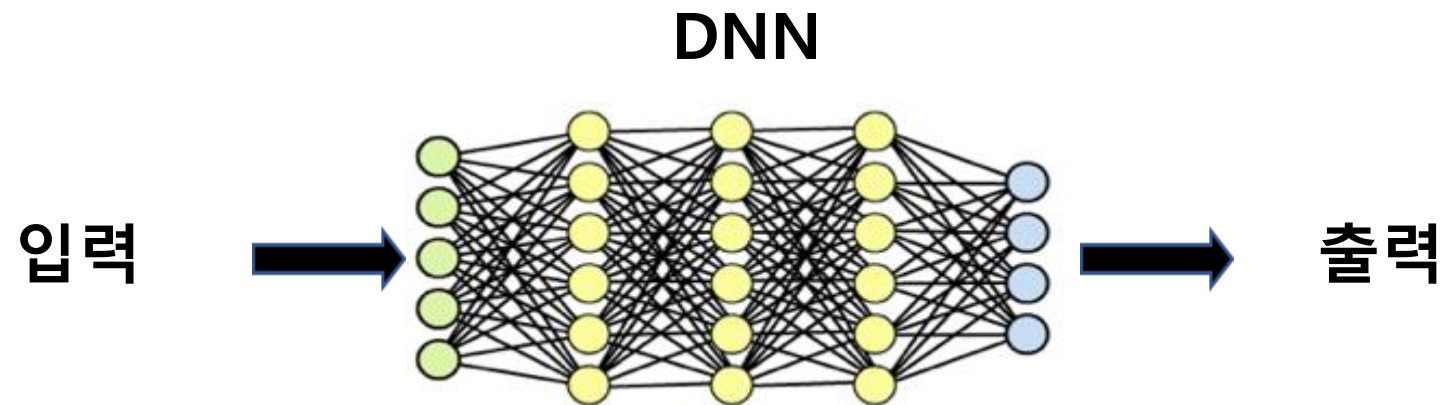
Universal Approximator

어떠한 함수도 근사화 할 수 있다.

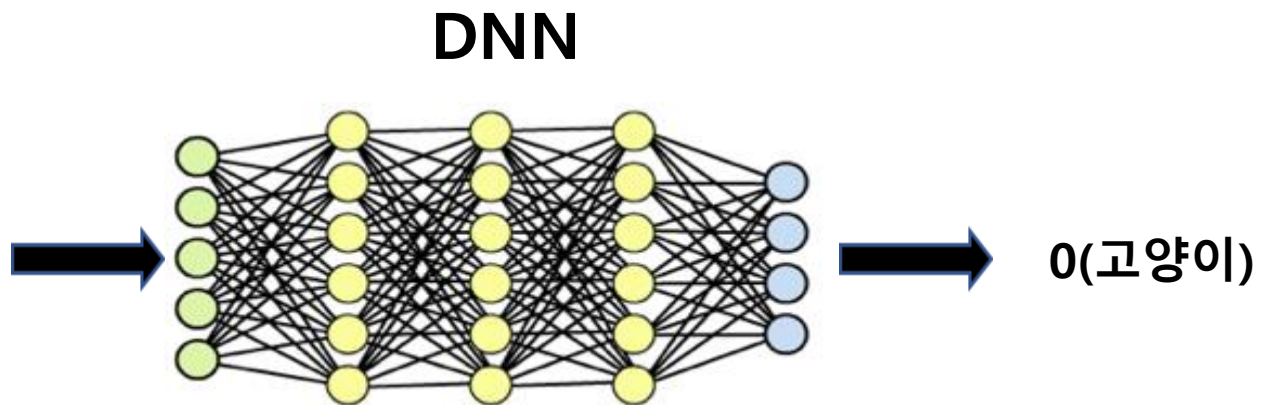
DNN의 함수 근사화 능력

- DNN은 임의의 함수를 근사화 할 수 있다.
- 함수의 내부를 모르더라도.
- 입력과 출력 데이터로.

함수

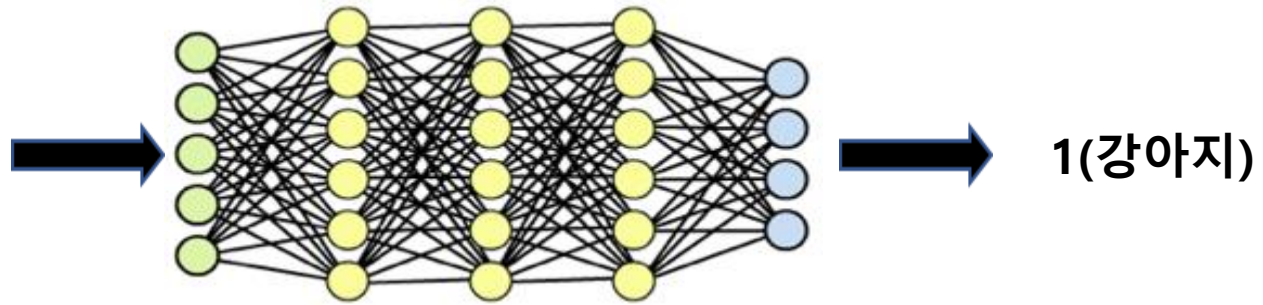


학습

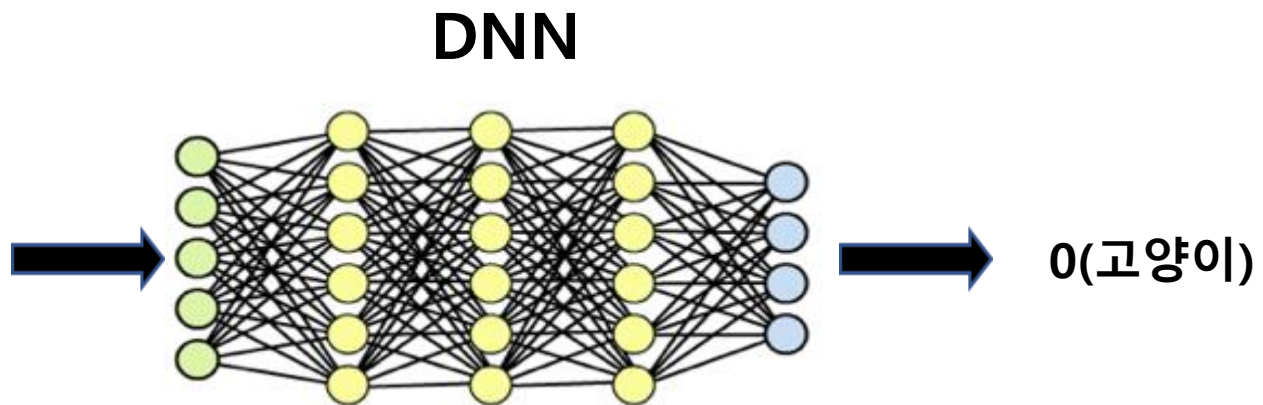


학습

DNN

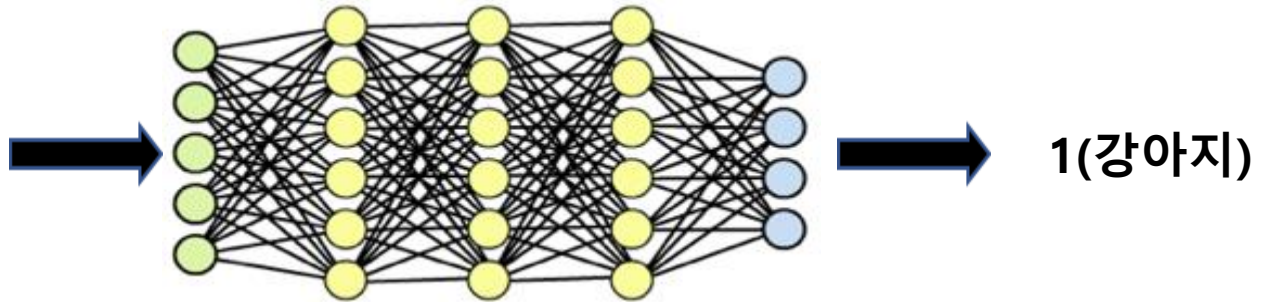


학습

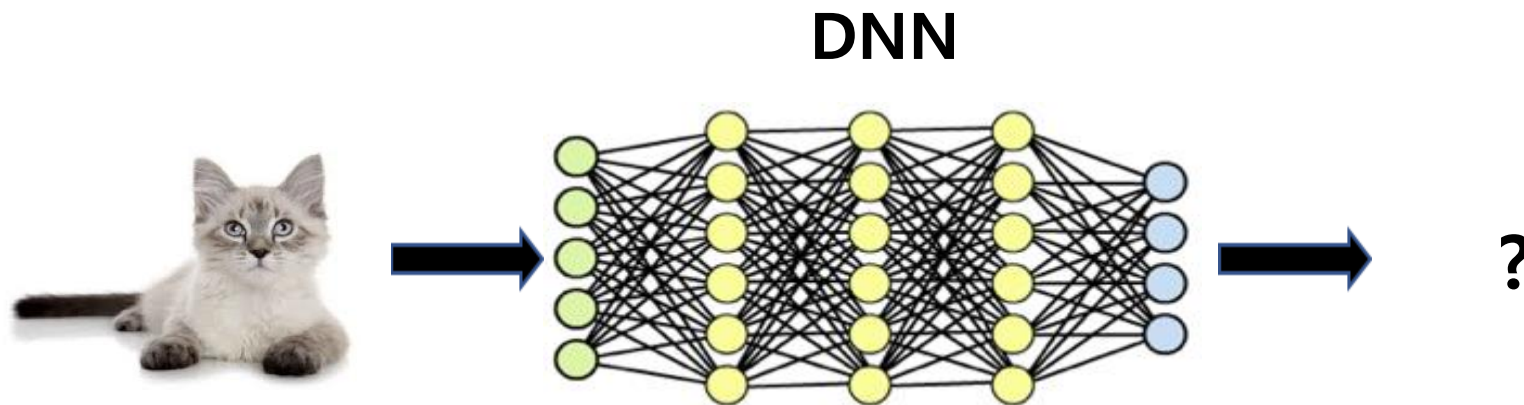


학습

DNN



학습



처음 보는 입력이라도 출력을 낸다. 의미 있는.

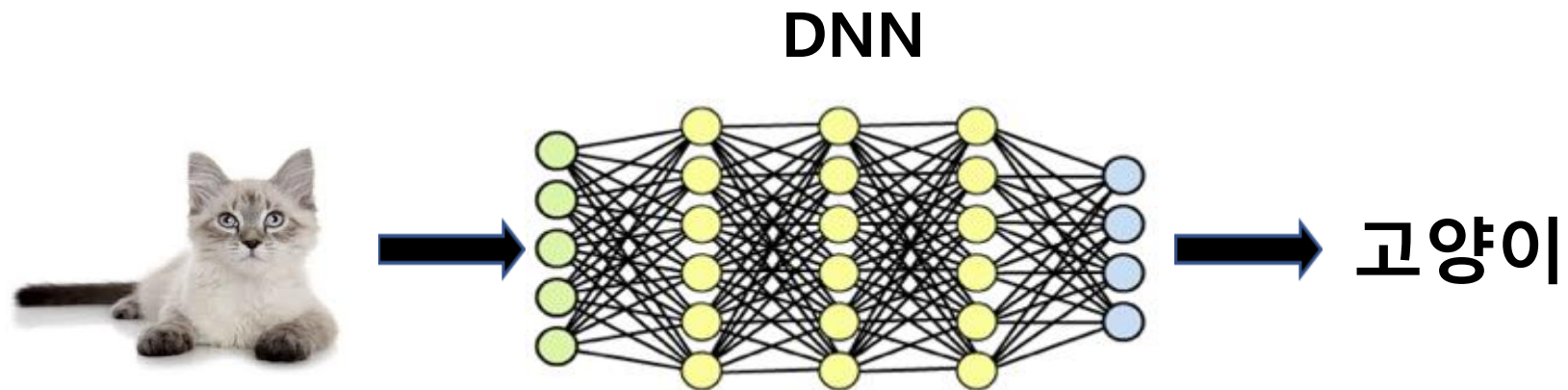
함수를 근사화 한 것이다

- 고양이, 강아지 구분 함수
- 어떻게 구분하는 지 정의하지 않았다.
- 정의하기는 힘들어도, 그 함수는 존재한다.
- 단순한 입출력 쌍 데이터로 그 함수를 근사화 하였다.

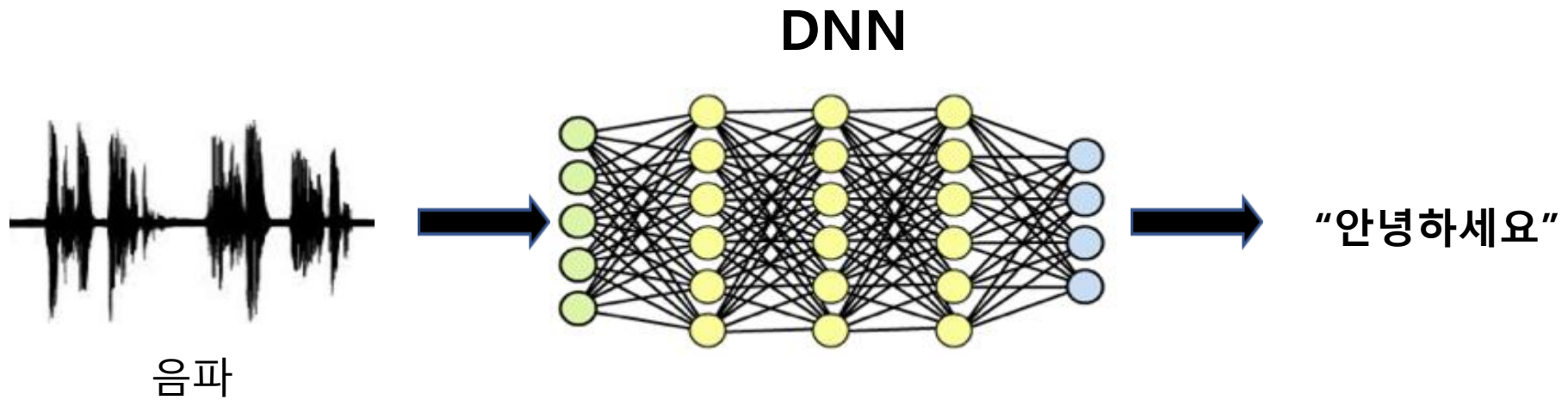
단순, 하지만 강력한 방법

- 단순히 입출력 쌍을 반복하여 학습시킨다.
- 하지만 로직을 찾아내기 어려운 문제에는 아주 효과적이다.
- 얼굴 인식, 물체 인식 같은.

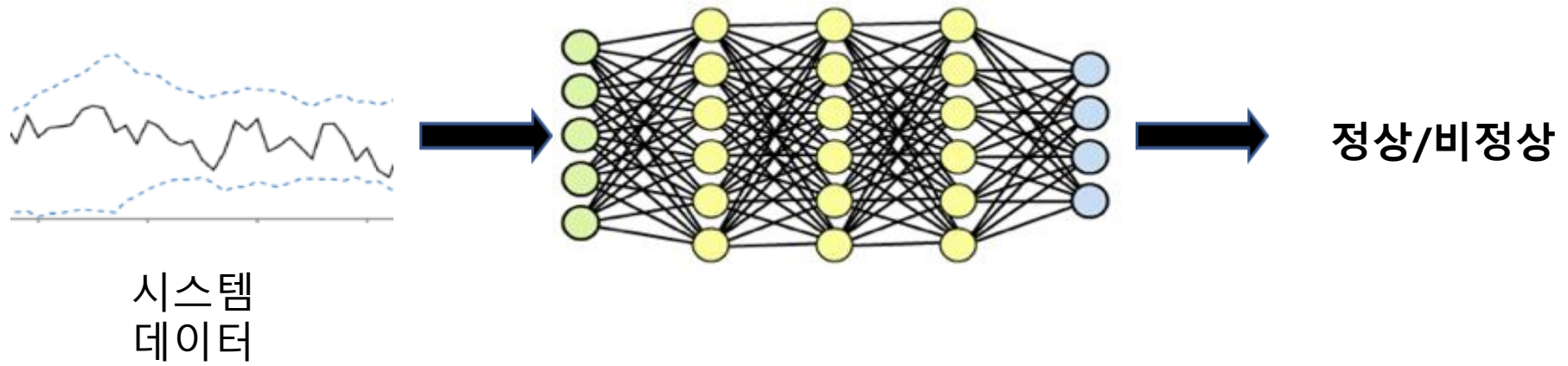
함수 예



함수 예

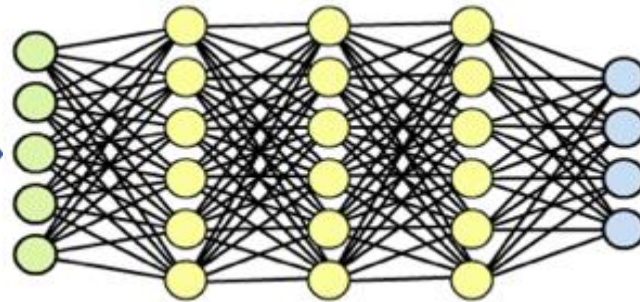


함수 예



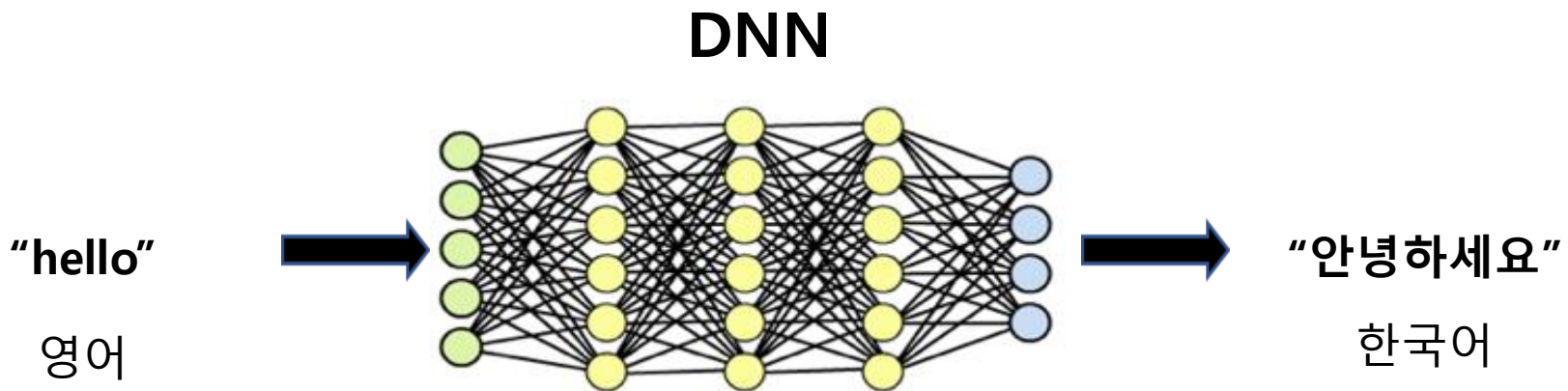
함수 예

DNN



다음 수

함수 예

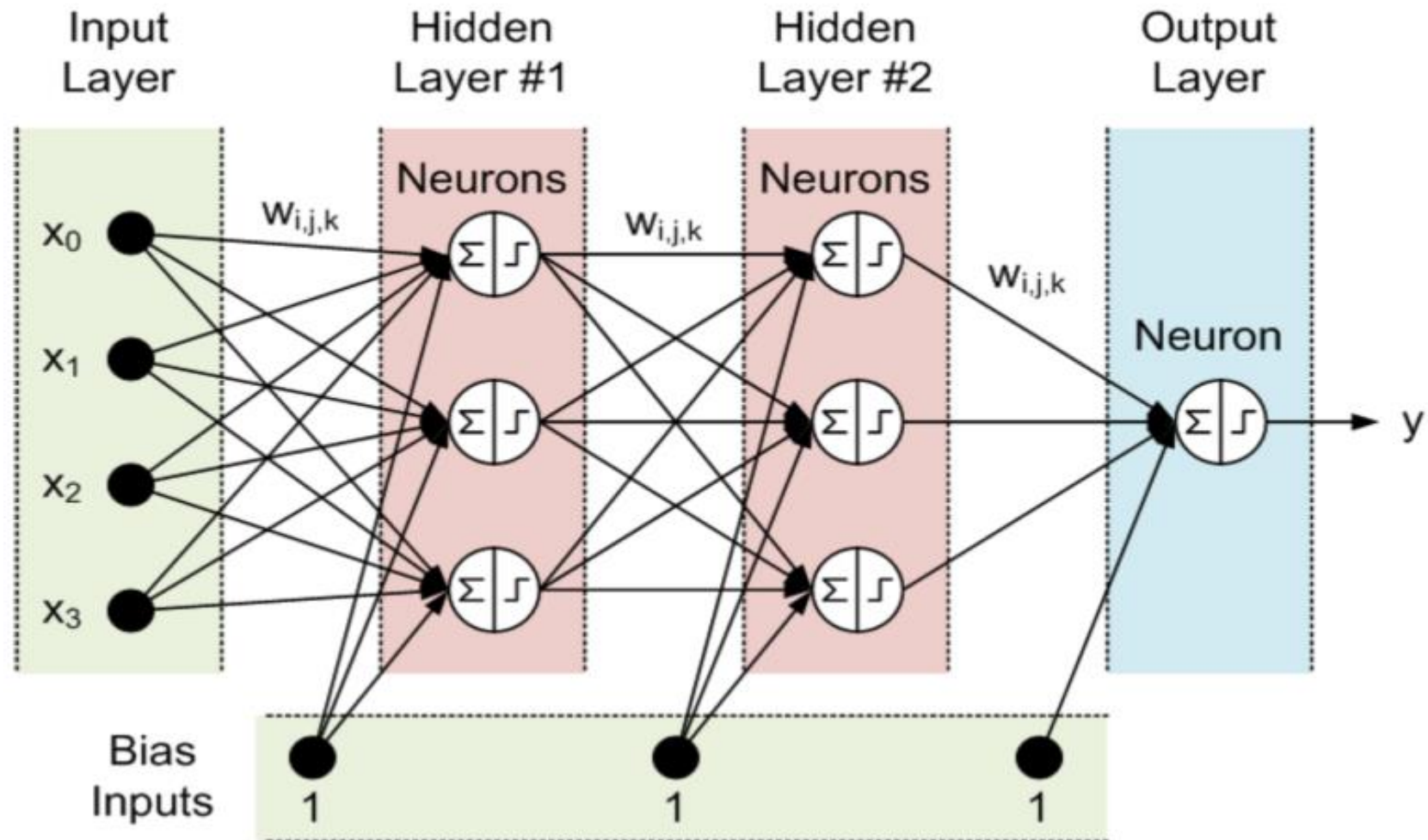


딥러닝

- DNN은 함수 근사화 능력이 있다.
 - 입출력 쌍을 계속 제공하여 DNN내부의 웨이트를 업데이트 한다.
 - 이를 위해 내부적으로 BP와 GD 알고리즘을 사용.
 - 충분한 입출력 데이터와 컴퓨팅 파워가 필요.
-
- 이를 반복하여 함수를 근사화하는 것을 DNN의 학습, Deep Learning이라 한다.

BP(Back Propagation), GD(Gradient Descent)

은닉층을 사용하는 이유



The universal approximation theorem found that a neural network with **one hidden layer** can approximate **any continuous function** (George Cybenko, 1989)

➤ 간단 퀴즈

- DNN은 무엇의 약자입니까
- Deep 은 왜 붙였을까요