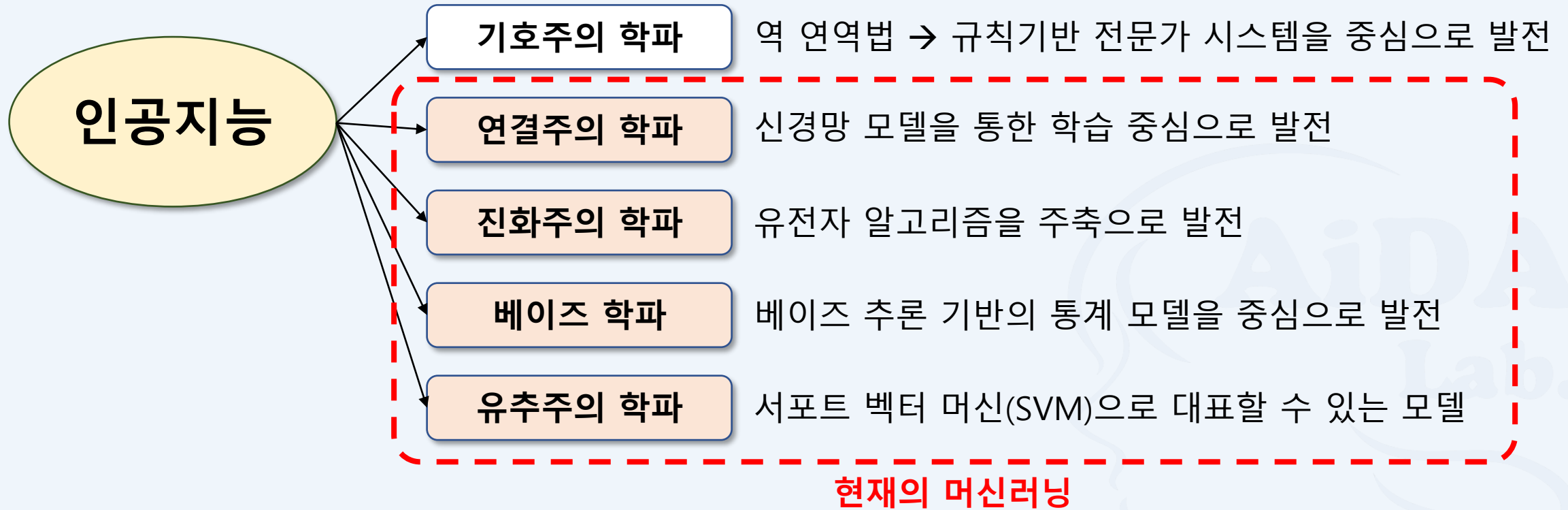


# 2021 인공지능 소수전공

31~32차시: 신경망 모델

2021.07.28 20:30~22:15

Seokhwan Yang



- 기호주의

- 흠의 귀납문제 :

- 우리가 본 것에서 시작한 일반화를 → 보지 못한 것에까지 적용하는 일을..

- 어떻게 하면 정당화 할 수 있을까?

- 정당화 할 수 있는 근거가 없다면 → 우리가 본 몇 가지 사례를 법칙으로 발전시킬 수 없음

- 기호주의란 경험으로 얻은 지식, 사고 체계를 논리적으로 귀결 시키는 과정

- 이러한 과정을 시스템으로 구현하는 것이 기호주의의 머신 러닝

- 기호주의 머신 러닝을 위한 하나의 예
- 데이트 신청의 성공 패턴

주일, 데이트 종류는  
결정적인 조건이 아님

날씨, TV프로그램 중  
중요한 조건이 있다고 추측

1~3 중에서 날씨가 쌀쌀한 경우가 있다면?  
4의 경우, 날씨가 온화했다면?  
1~3 중에서 TV 프로그램이 좋았다면?  
4의 경우, TV 프로그램이 따분했다면?

경우	주일	데이트 종류	날씨	오늘 밤 TV 프로그램	데이트 성사 여부
1	주중	저녁식사	온화	따분함	승낙
2	주말	클럽	온화	따분함	승낙
3	주중	클럽	온화	따분함	승낙
4	주말	클럽	쌀쌀함	좋음	거절
<b>5</b>	<b>주말</b>	<b>클럽</b>	<b>쌀쌀함</b>	<b>따분함</b>	<b>?</b>

- 이처럼 데이터의 패턴을 분석, 그 결과를 예측할 수 있도록 학습하는 것이 기호주의의 머신 러닝

## • 기호주의 머신 러닝의 문제점

- 우리가 발견한 패턴이 실제로 존재하는가? → 통계적 검증 필요, 수많은 데이터와 경우의 수
- 데이터에 적합한 단순한 가설을 선택한다면? → 사람이 편한 것이지 정확도, 성능 향상은 없음
- 기호주의 머신 러닝은 아는 것이 너무 적은 상태에서 학습 시작 → 결승점 도달 실패 확률 높음
- 역연역법을 통해서 논리적으로 예측하는 방안 → 나름대로 좋은 성과를 거둠. 그러나...
  - 너무 많은 규칙을 관리해야 함 → 계산량 문제 → 해결? → 의사결정트리 (스무고개놀이) 등
  - 잡음(무관한 데이터)에 쉽게 오류를 일으킴
  - 가장 큰 문제: 실제 개념은 규칙의 모음으로 간결하게 정의되는 일이 거의 없다는 사실

- 연결주의

- 심리학자 “헵”의 규칙을 기반으로 만들어진 유형

- 헵: 신경과학자보다 먼저 신경세포의 연결방식을 제안한 심리학자
    - 헵의 규칙이란?
      - 시냅스의 앞과 뒤에서 동시에 신경세포가 흥분할 때, 해당 시냅스의 효율이 강화됨
      - 적당한 추측을 기반으로 심리학과 신경과학의 착상들을 통합해 놓음

- 연결주의 머신 러닝의 개념

- 각 개념(데이터)와 기억은 두뇌에서 세포의 모임으로 나타난다
- 개념(데이터)은 모든 곳에 조금씩 저장되어 있다.
- 두뇌는 수십억의 신경세포가 동시에 동작하며 많은 계산을 수행한다. 그러나 각 신경세포는 1초에 1000번 정도 반응하므로 계산이 느리다(병렬시스템)
- 신경세포에는 수천개의 신경 접합부가 있다. 등..

→ 두뇌가 어떻게 만들어지는가 이해해야 두뇌를 시뮬레이션(모의 실험)할 수 있으며 인공지능(머신 러닝)은 두뇌를 재 구축함으로써 구현 가능하다

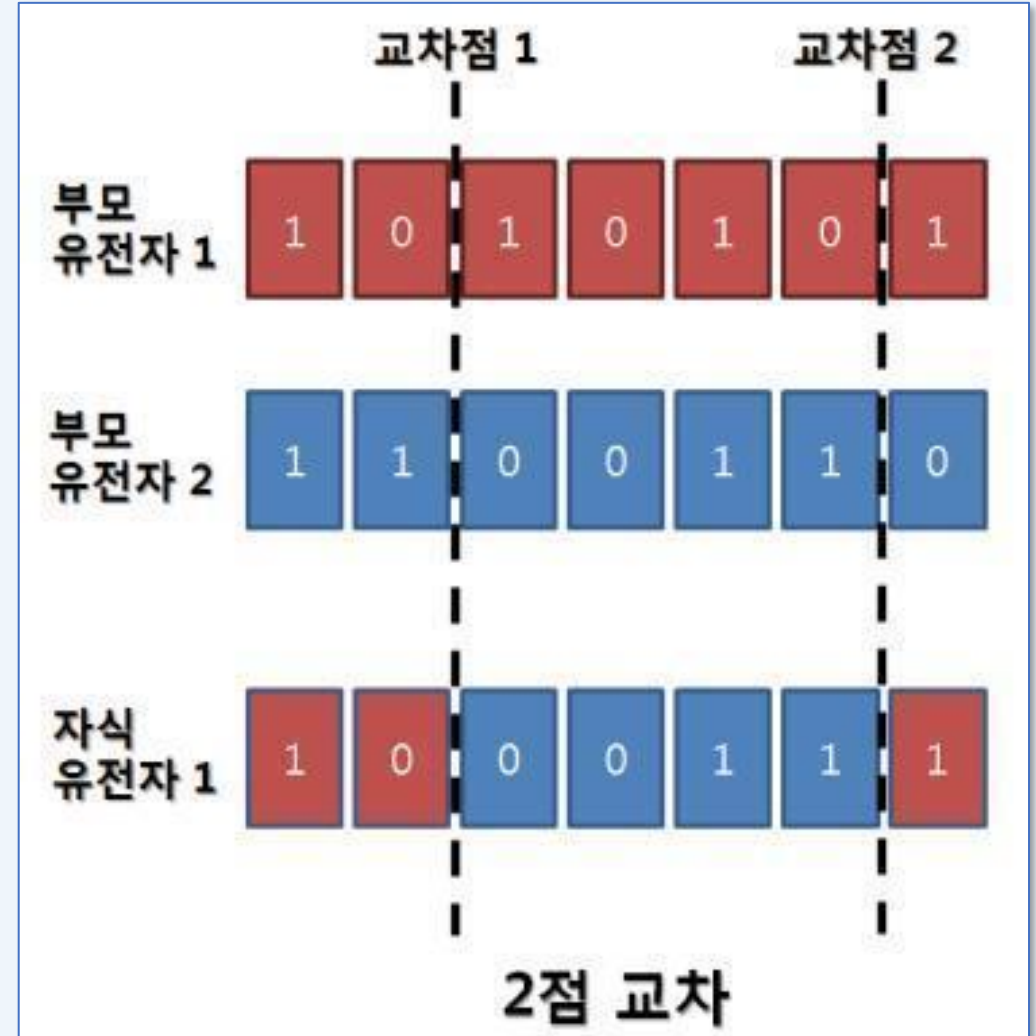
- 연결주의 머신 러닝의 문제점

- 두뇌, 신경의 구조, 작용 등에 대하여 아직 모르는 부분이 너무 많음
- 연결주의 등장 당시의 기준으로 컴퓨터의 성능이 지나치게 낮음 (구현 불가능)
- 병렬처리해야 하는 데이터가 너무 많음 등...



## • 진화주의

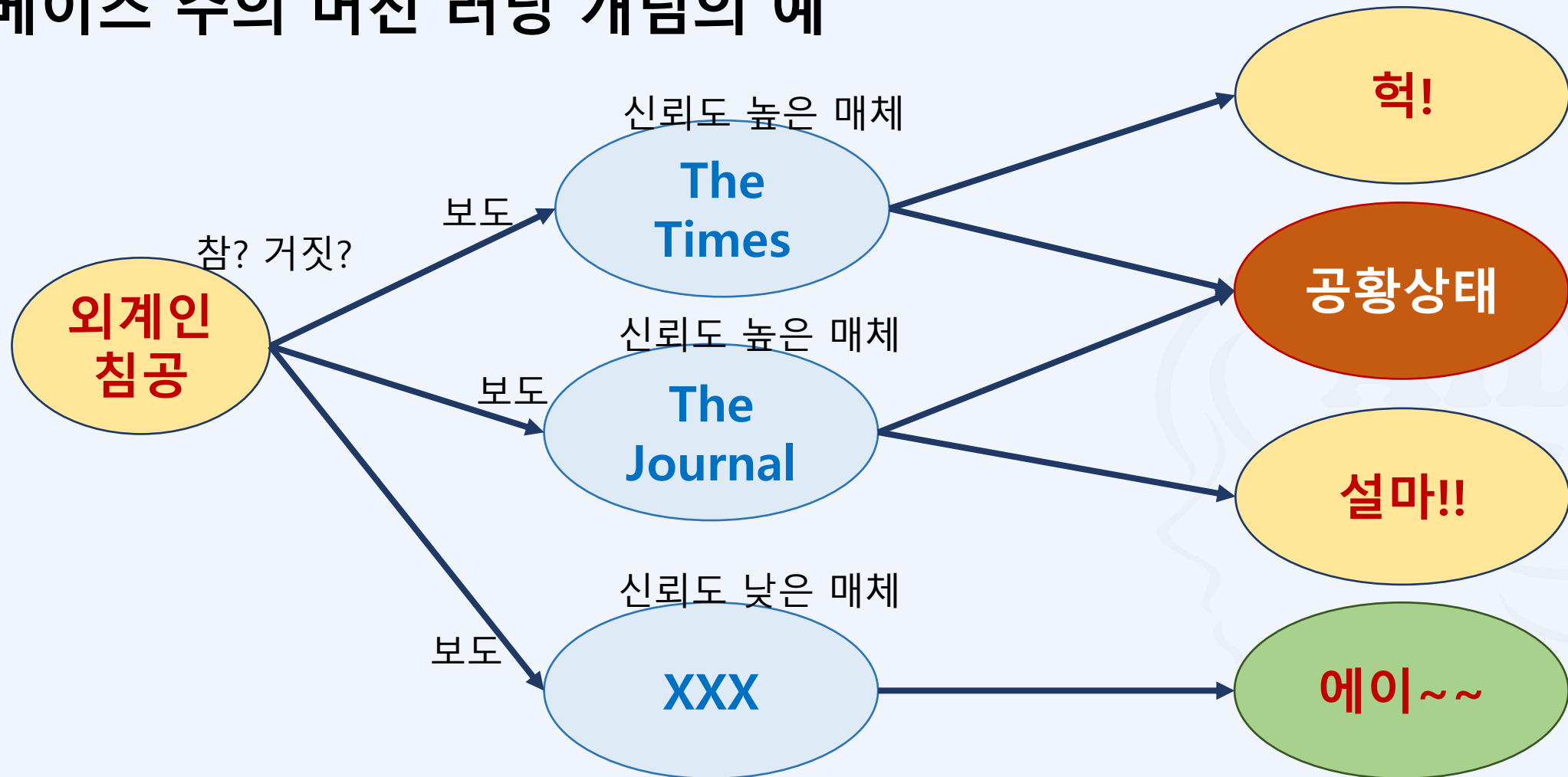
- 진화론에서 출발하여 진화, 돌연변이 등을 통해 학습을 수행하는 유전자 알고리즘 중심 연구
- 연결주의(신경망) 연구자였던 홀랜드가 생물학자 겸 통계학자인 로널드 피셔의 "자연선택의 유전 이론" 논문을 접한 후 제안한 이론을 기반으로 함
- 가장 적응력이 높은 유전자만 살아남고 살아남은 유전자가 가장 정확한, 또는 적절한 결과를 도출할 가능성이 크다



- 베이지 주의

- 통계학의 일부인 베이지 정리를 기반으로 한 머신 러닝
- 어떤 원인에서 어떤 결과가 일어날 가능성이 더 높을 수록  
→ 그 결과가 나타났을 때 그것이 원인일 가능성이 더 높다 (확률적인 기반)
- 원인과 결과, 즉 **인과관계에 대한 추론을 기반으로 학습, 예측**을 진행함
- 그런데... 인간은 언어 추리가 연관되면 베이지 추론(베이지 정리를 기반으로 하는 추론)을 매우 잘 하는 것은 아니다. 인간은 원인의 사전 확률을 무시하는 경향이 있다.

## • 베이즈 주의 머신 러닝 개념의 예



- 유추주의

- 사물, 현상에 대한 유추를 기반으로 학습을 진행하는 연구
- 통계학에서 먼저 알고리즘화 되기 시작했으며 컴퓨터 과학 전 분야에서 많은 연구가 진행되고 있는 분야
- 신경망, 기호주의, 유전자 알고리즘 등 다양한 머신 러닝 모델에도 영향을 끼침

- 유추주의 개념의 예시

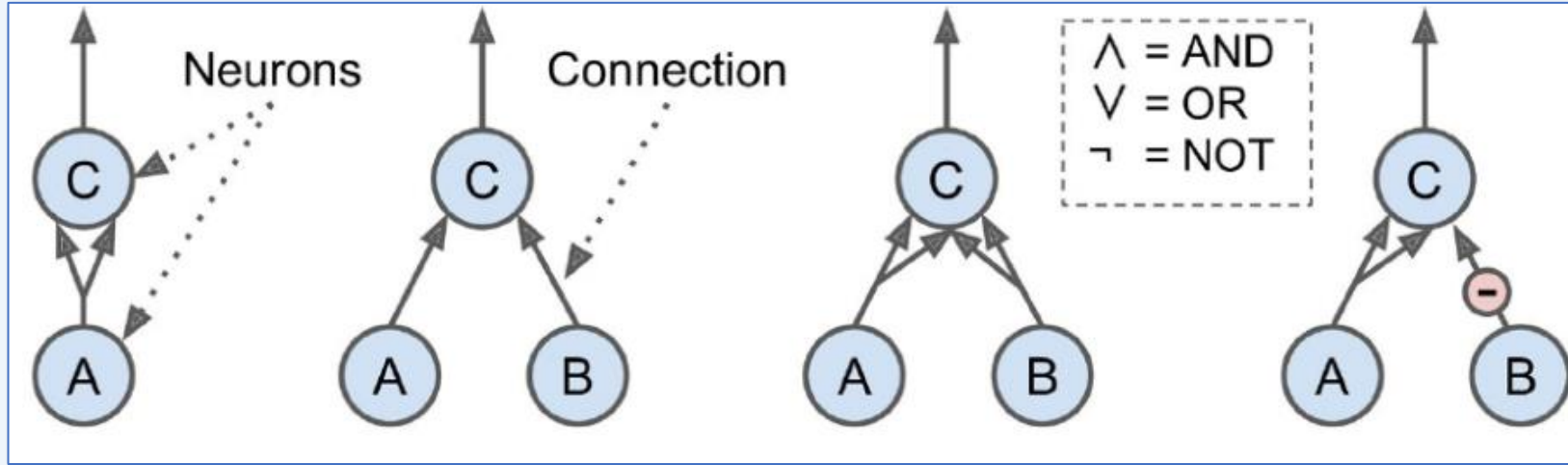
- 역사상 악명높은 사기꾼 - 프랭크 애버그네일 주니어
- 의학적인 교육은 전혀 받지 않은 채로 1960년대 후반 애틀란타에서 1년 가까이 의사로 행세함
- 아무도 모르게...
- 행위
  - 빈 진료실에 들어감 → 아무것도 모르는 환자 입실 → 환자의 증상을 들음 → 캐비닛에 들어있는 환자들의 진료 기록 검색 → 유사한 다른 환자의 기록을 꺼내어 동일한 진단 내림 → 1년 가까이 아무도 의심하지 않음

- 5가지 연구 유형에서...

- 모든 것을 만족하는 마스터 알고리즘은 아직까지 나오지 않았다
- 각 유형은 각각의 장, 단점을 가지며 어느 하나가 완벽한 모델은 없다
- 최근의 추세는 각 유형이 서로 융합, 협력하여 서로의 단점을 보완하려는 움직임
- 새로운 연구를 위해서도 서로의 장, 단점을 잘 알고 활용할 필요가 있음

- 신경세포의 간단하고 효과적인 처리 방식에 착안해 구현된 머신 러닝 알고리즘(모델)의 한 종류
- 신경세포의 형태와 동작을 극도로 단순화 시킨 뉴런 모델을 다수 연결하여 망, 즉 네트워크를 만든 모델  
( = 다량의 뉴런(Neuron)들이 층(Layer)으로 연결되어 간단한 계산과 연결 방식을 통해 복잡한 문제를 해결하는 모델 )
- 뉴런의 동작 방식은 컴퓨터 프로그램의 방식에 비해 다양한 장점을 지님

- 1943년 워런 맥컬록, 월터 피트의 최초의 신경망 모델이 시초

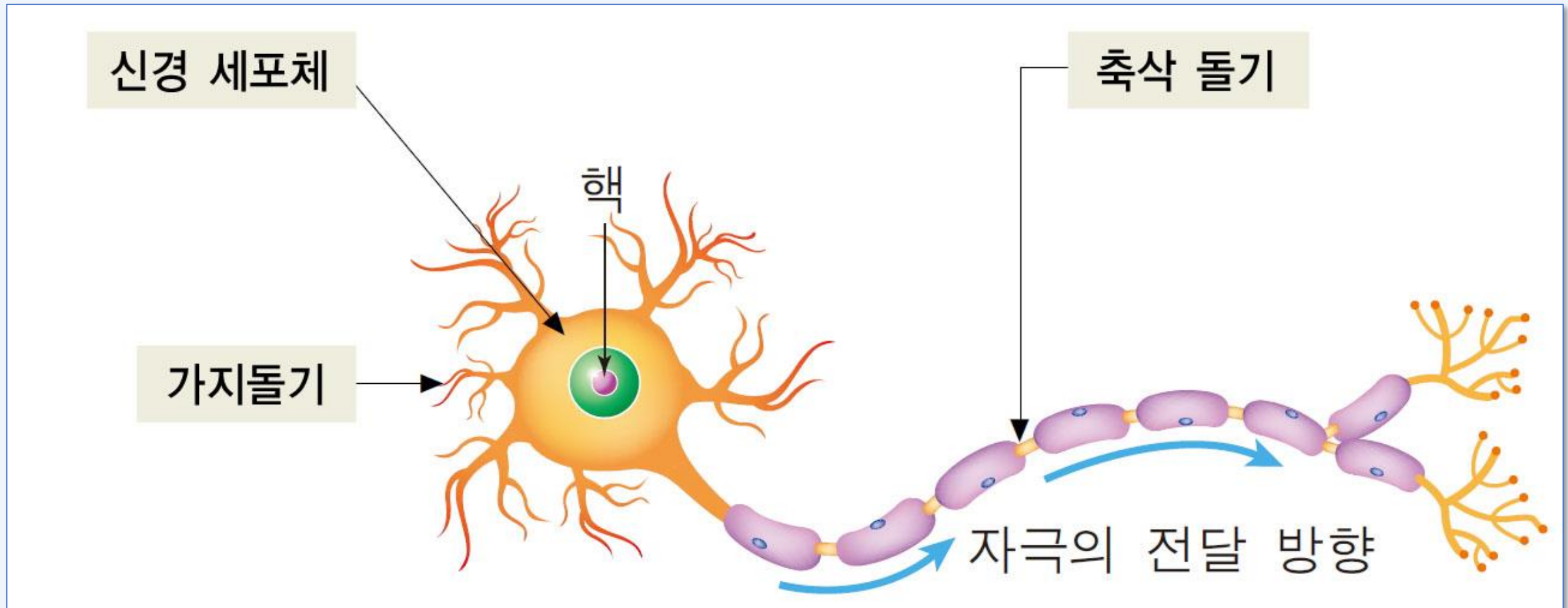


입력되는 데이터들의  
연결형태에 따라서  
결과가 결정된다

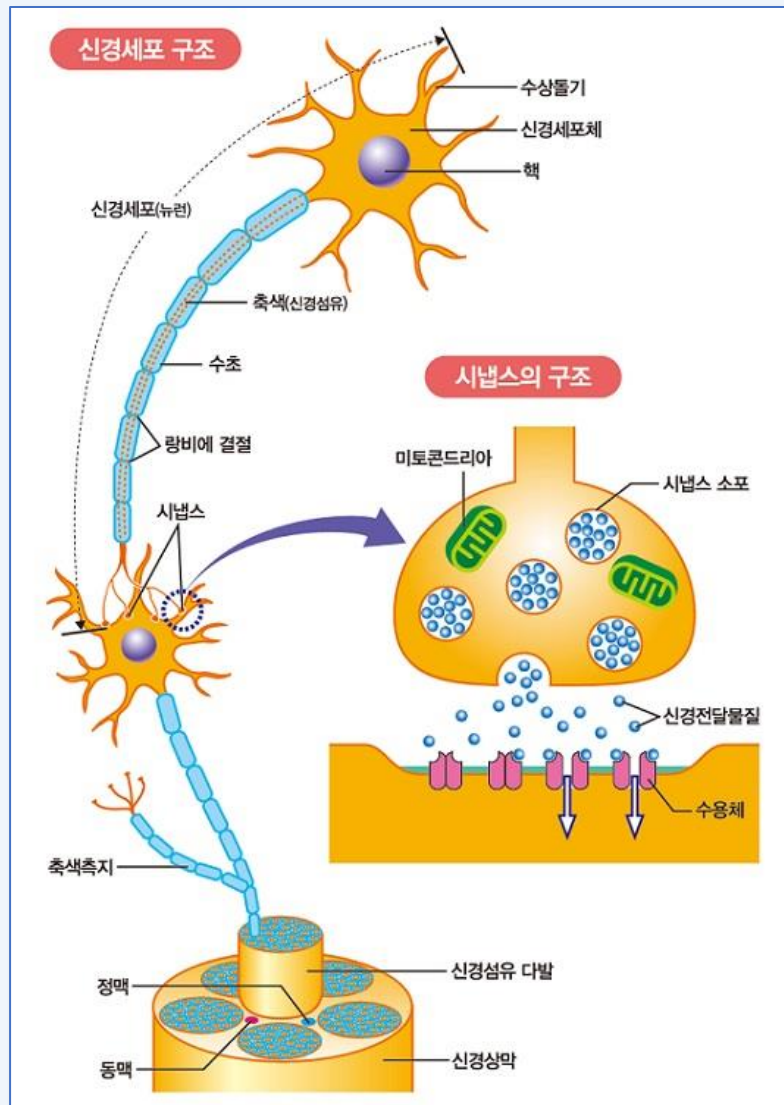
- 헵의 규칙이 신경망 모델의 동작을 정의하는 기반이 됨
  - 시냅스의 앞과 뒤에서 동시에 신경세포가 흥분할 때, 해당 시냅스의 효율이 강화된다.



# 신경세포의 구조



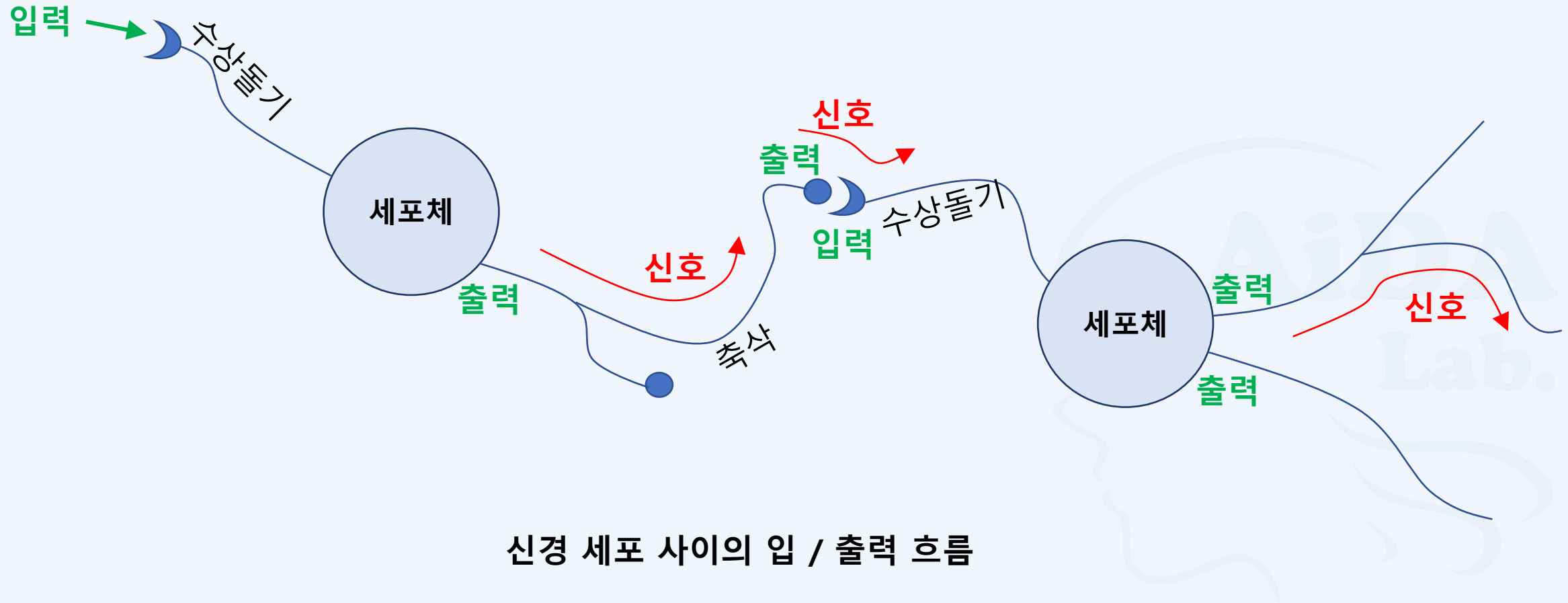
# 신경세포의 연결 형태



- 신호의 전달은 전기로 이루어짐
- 신경세포의 말단에는 시냅스가 존재
- 시냅스 사이의 신호 전달은 신경전달물질이라는 화학물질을 통해 전달됨
- 신경전달물질
  - 세로토닌
  - 도파민
  - 엔돌핀
  - 아드레날린 등..

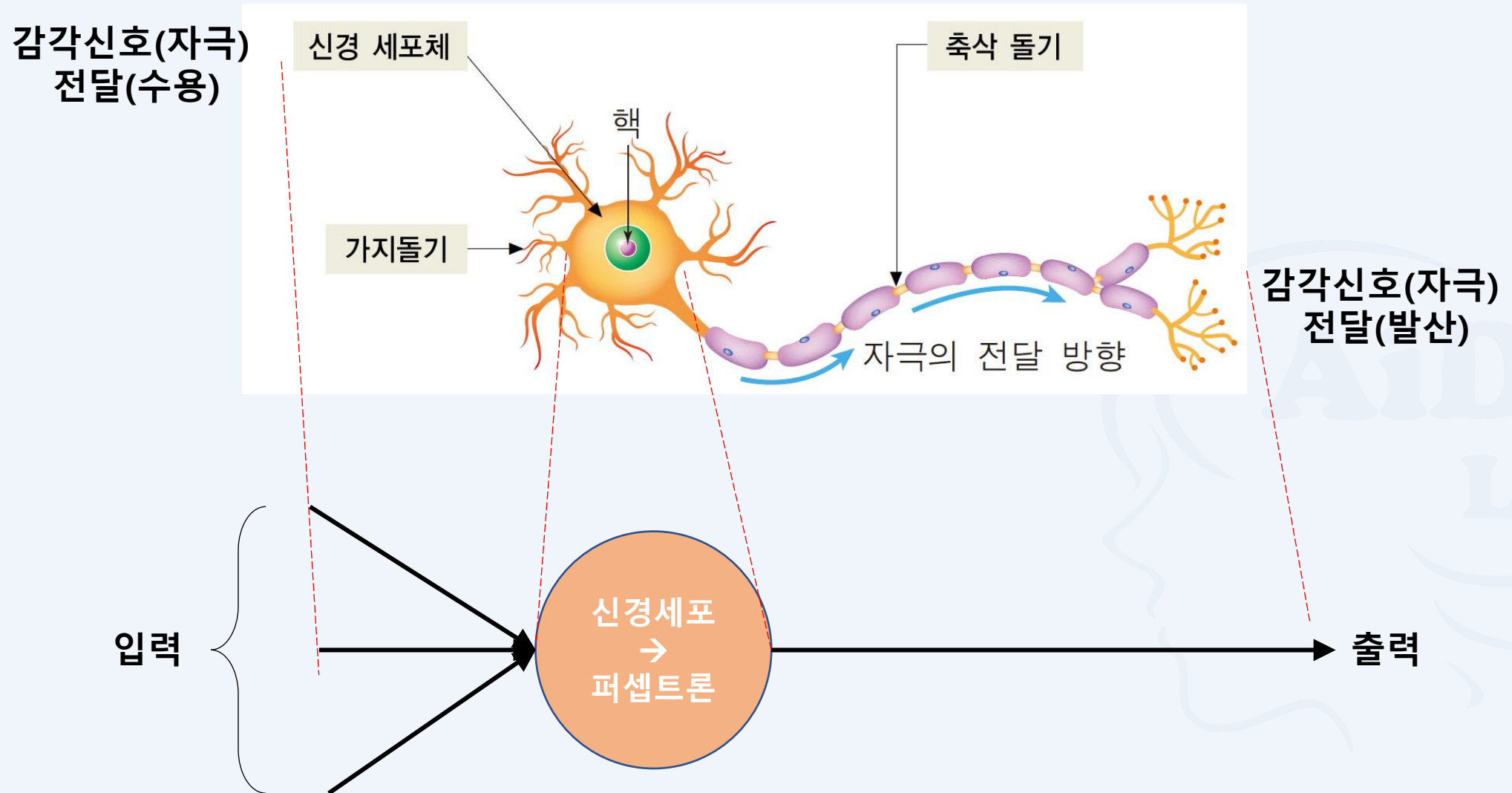
- 신경체계를 구성하는 수많은 신경세포들
- 다양한 감각기관을 통하여 (전기)신호를 발생, 전달
- 각 신경세포는 수많은 시냅스를 통해 신호를 전달 받음
- 전달 받은 신호는 대체로 무시하지만.. 동시에 전달된 신호의 합이 임계값을 넘으면 활성화(발산, 흥분한다 라고 표현함)
- 활성화 된 신경세포는 활성화 패턴에 따라 신경전달물질 분비
- 이웃 신경세포는 신경전달물질을 수용하면서 이온화 작용, 화학작용을 통하여 전기 신호 발생
- 처리 단계 반복

# 신경세포에서의 데이터 흐름

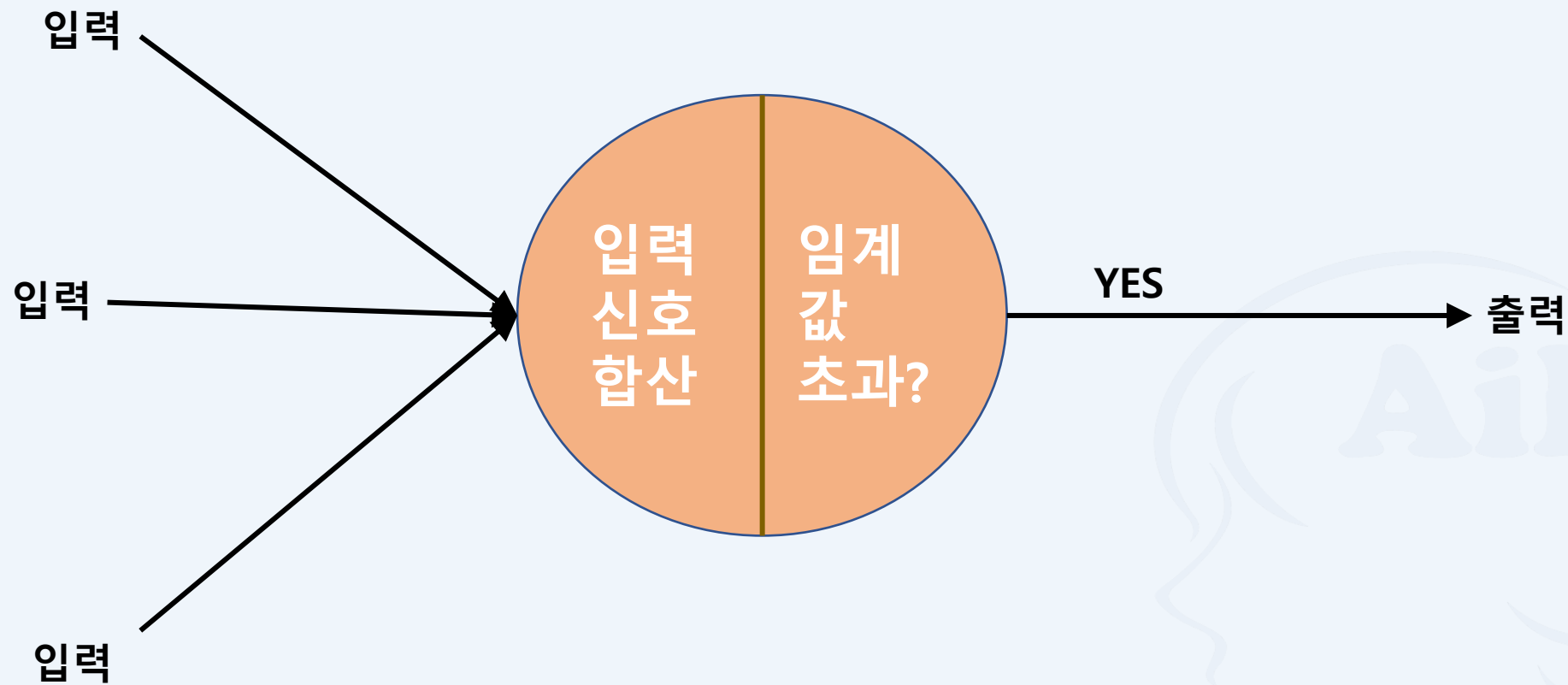


신경 세포 사이의 입 / 출력 흐름

# 신경세포, 신경망을 어떻게 단순화하였나?



# 신경세포를 어떻게 단순화하였나?

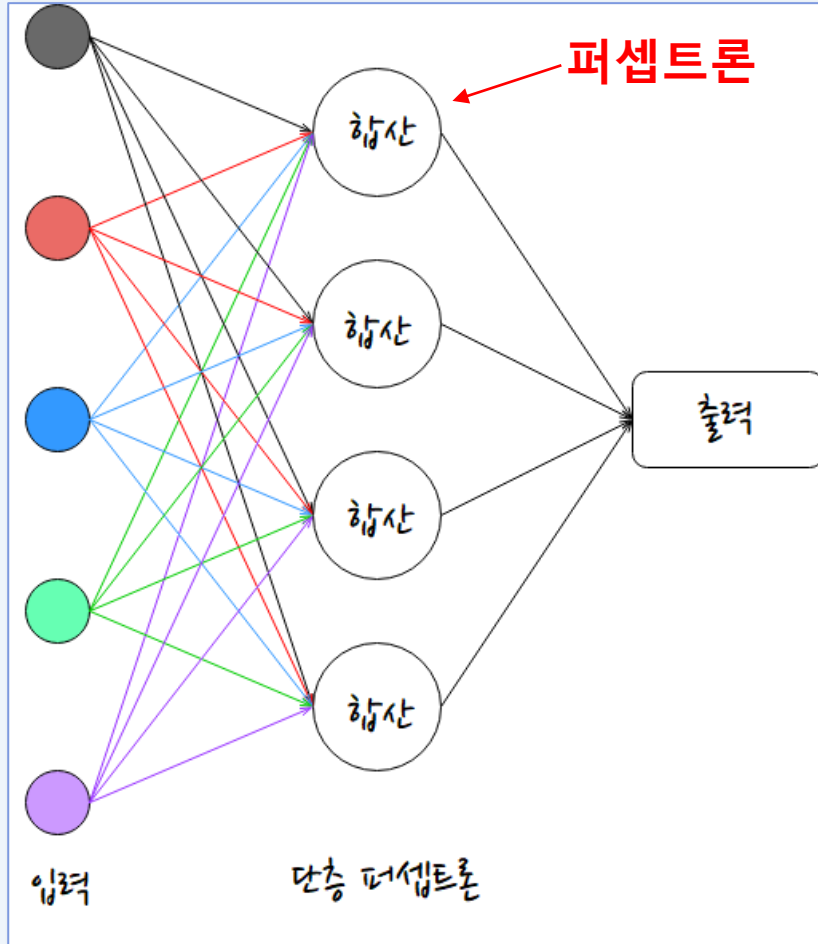


퍼셉트론 모델의 형태

# 퍼셉트론 모델의 문제점? 한계?

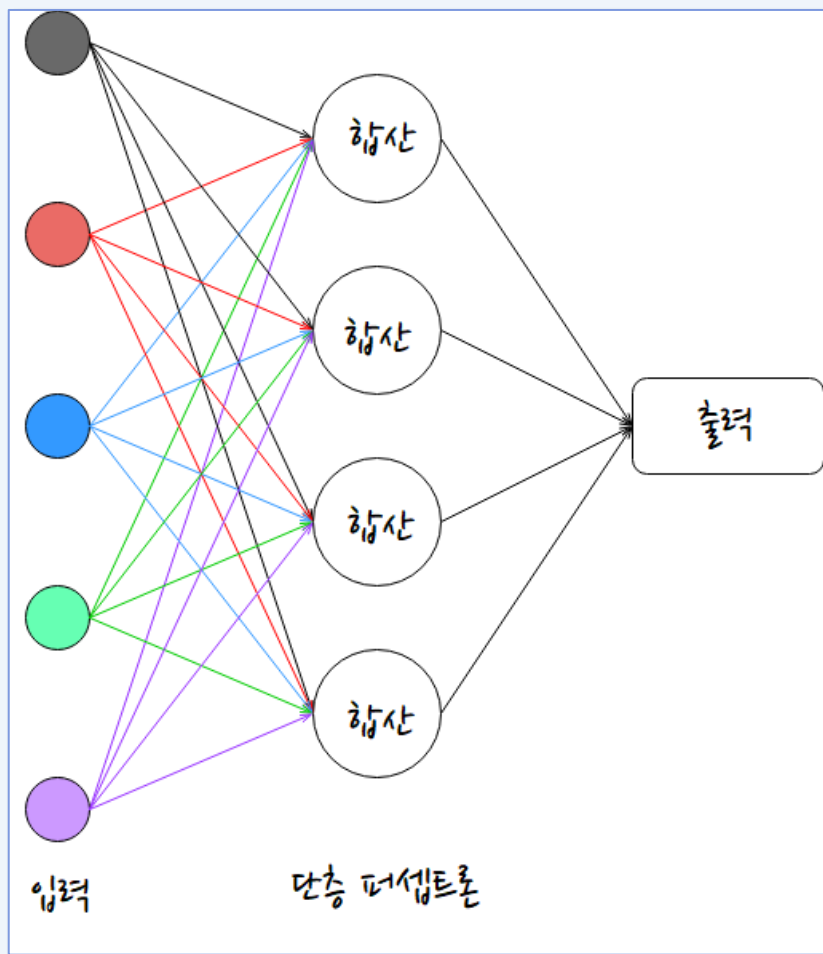
- 퍼셉트론 모델은 실제 신경세포의 형태를 극도로 단순화한 모델
- 전기신호의 전달, 합산, 발산을 모델링 함
- 그러나... 신경세포는 전기신호만 전달하는 것이 아님
- 신경세포의 활성화는 전기신호의 합산으로 이루어지더라도 해당 신호를 전달할 때는 다양한 신경전달물질이 활동함
- 동일하게 전기신호는 전달되더라도 신경전달물질의 종류에 따라 사람의 반응은 달라짐  
→ 이런 부분은 전혀 반영되지 않은 모델
- 그러나 기능의 모방을 목표로 하는 약 인공지능의 시점에서는 충분히 활용가치가 높은 모델

## • 단층 퍼셉트론(SLP, Single Layer Perceptron)



- 다수의 퍼셉트론이 하나의 층을 이루고 있는 형태
- 센서 데이터 등 다양한 데이터를 각 퍼셉트론의 입력으로 전달
- 각 퍼셉트론은 입력된 데이터를 모아서 합산
- 합산 결과가 임계 값을 넘으면 1, 넘지 않으면 0 출력
- 입력층에서 각 퍼셉트론으로 진행되는 통로에는 가중치 적용 (가중치는 모든 통로가 각각 다르게 적용될 수 있음)
- 왼쪽 그림에서 4개의 퍼셉트론이 각각 1, 0, 0, 1 이라는 결과를 낸다면 최종 출력은 1001 이라는 2진수 값이 나오는 형태

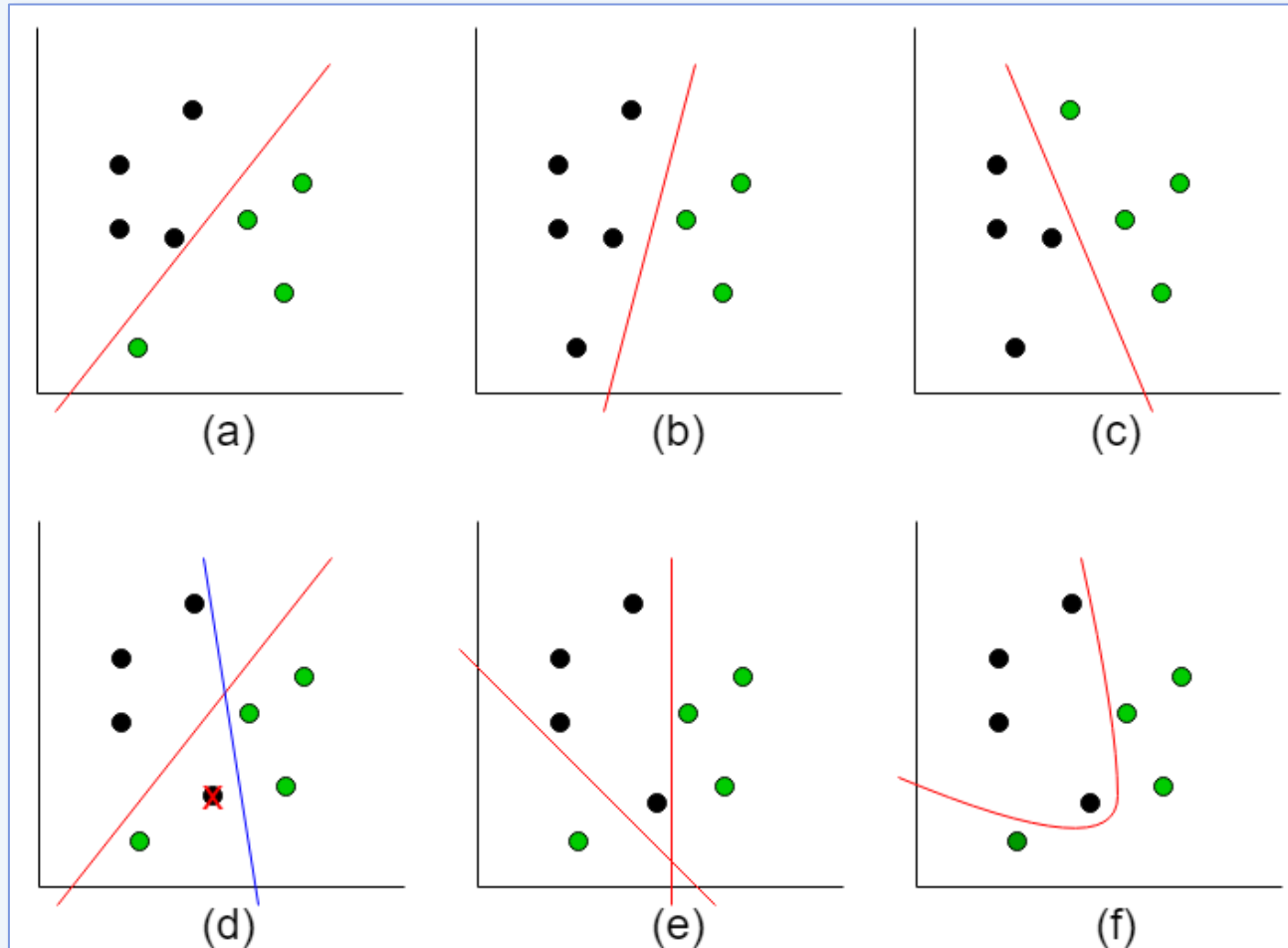




## • 단층 퍼셉트론의 구조를 보면...

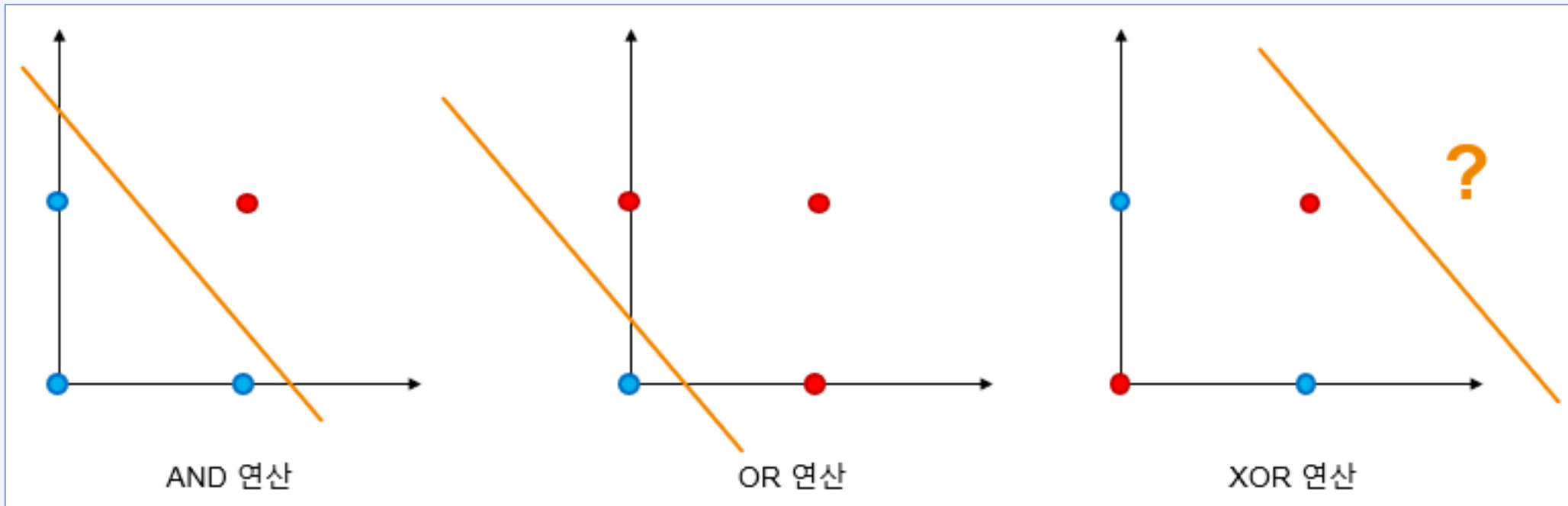
- 가중치를 변경할 수 있는 방법이 없다  
→ 학습이라는 개념이 없다
- 한 번 생성된 후에는 아무런 변형이 없는 단순한 분류 알고리즘에 불과함
- 한 층의 변경가능한 퍼셉트론만 존재  
→ 1개의 선을 그어 분리 가능한 패턴만 분류 가능

- 직선으로 데이터 분류하기

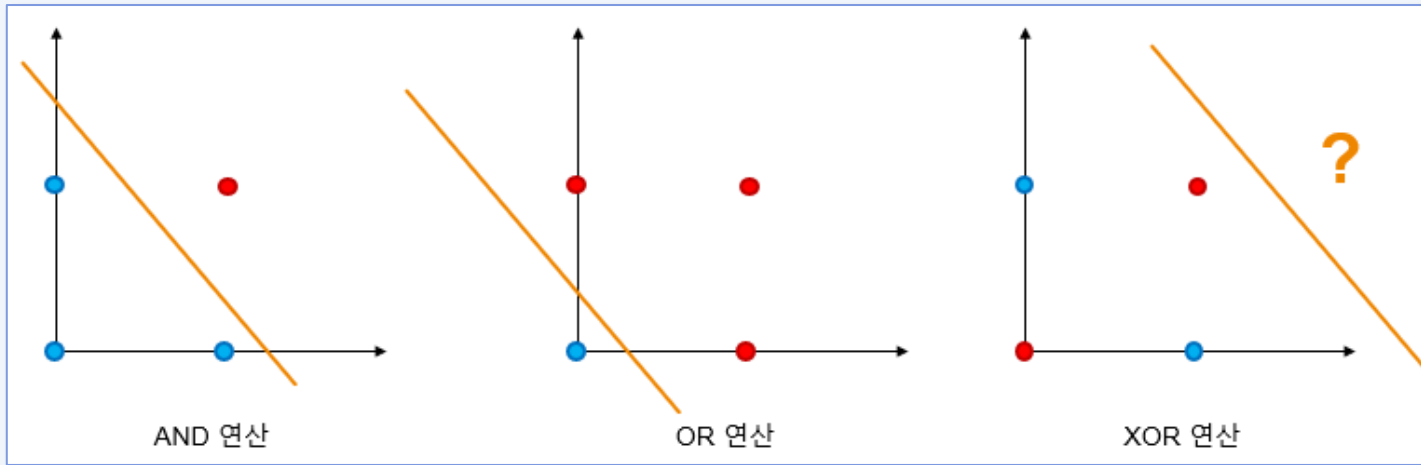


XOR 불가능 문제도  
여기에서 확인됨

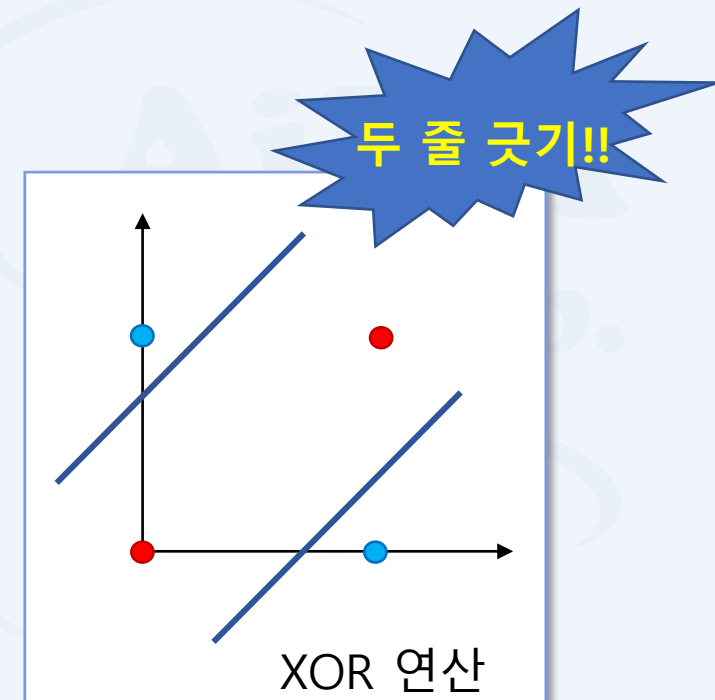
- XOR 문제



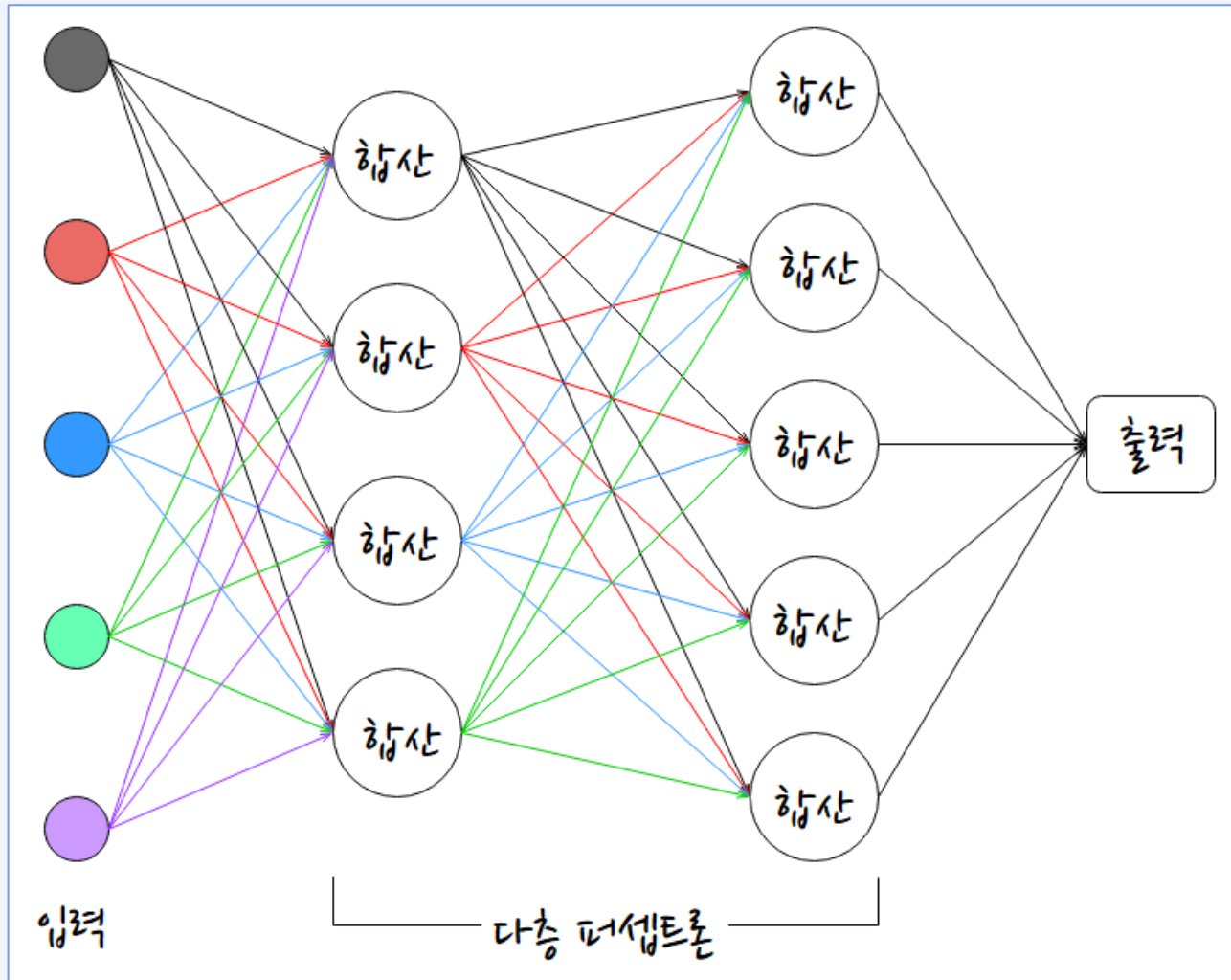
- XOR 문제



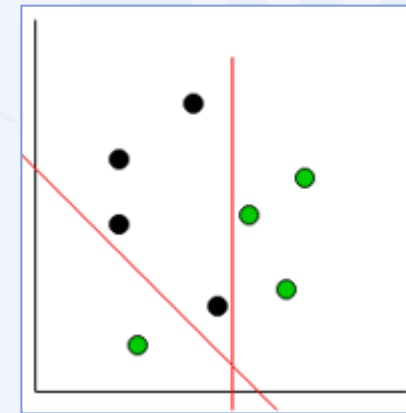
그런데 이렇게 하면?



# 다층 퍼셉트론(Multi Layer Perceptron)



단층 퍼셉트론의  
가장 큰 문제 해결



# 다층 퍼셉트론(Multi Layer Perceptron)



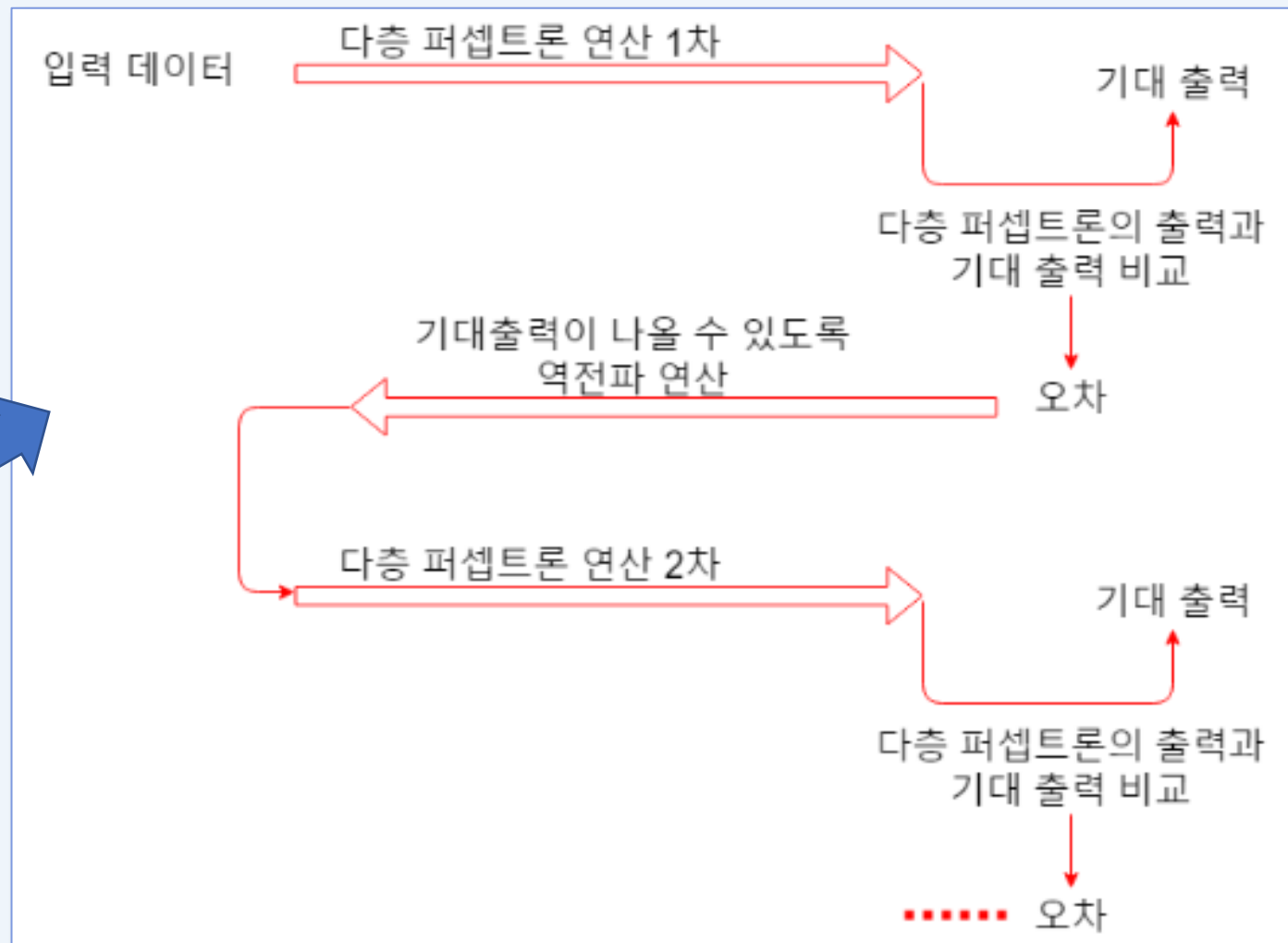
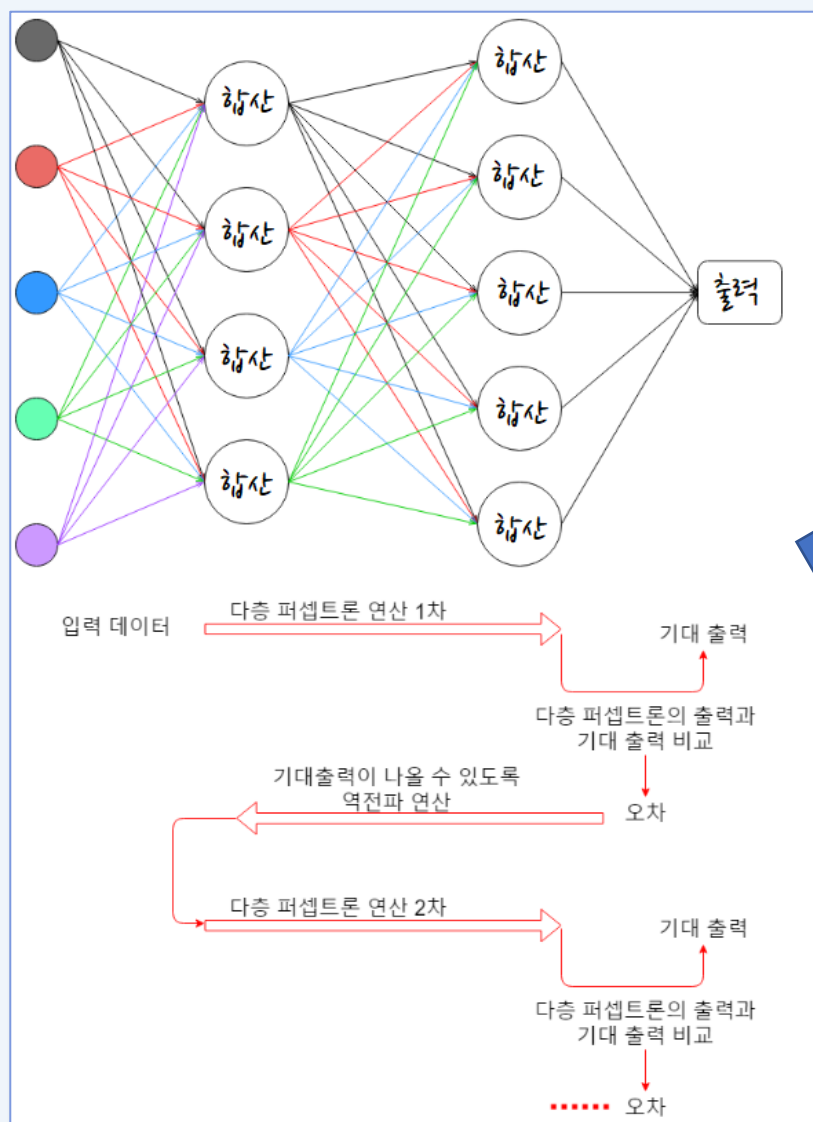
- 또 문제점

- 단층 퍼셉트론과 마찬가지로 각 통로의 가중치를 변경할 방법이 없다
- 한 번 생성되면 변경 불가능한 분류 알고리즘
- 역시 학습의 개념이 없다     인공지능이 아닌 단순한 분류 알고리즘

- 해결책은?

- 수행할 때마다 예전 데이터를 들고 와서 가중치를 수정해 주면 어떨까?
- 그럼 아예 앞뒤로 왔다 갔다 반복하면서 가중치를 바꾸어 주면 어떨까?
- Back Propagation (역전파) 알고리즘 제안

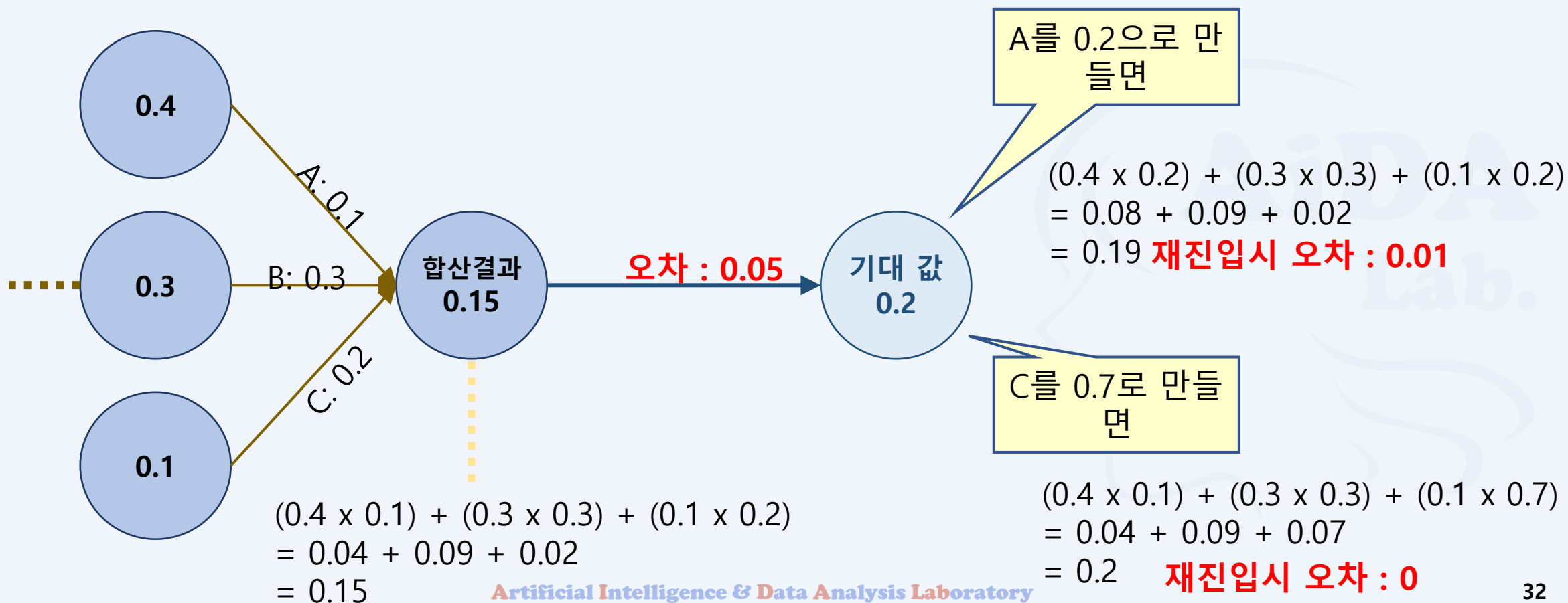
# Back Propagation (역전파) 알고리즘



# Back Propagation (역전파) 알고리즘

## • 역전파 시 어떻게 가중치를 조절하는가?

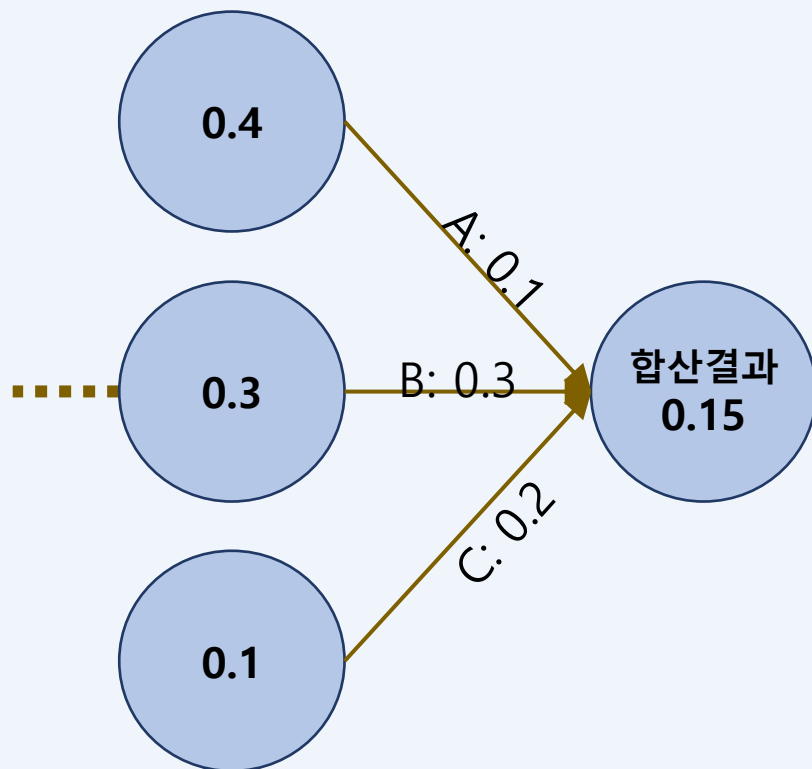
- 조절 내용: **은닉층을 거친 결과값**과 **기대한 결과값**의 오차를 줄이는 방향으로 수정





- 역전파 시 어떻게 가중치를 조절하는가?

- 조절 방법

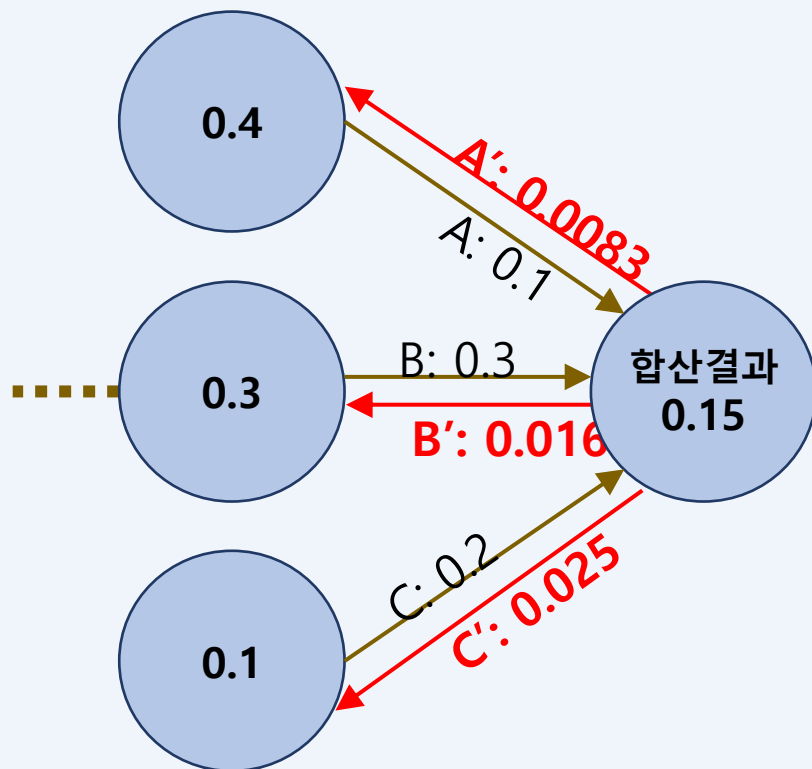


오차 값 0.05를 줄이기 위해서

- 가중치의 비율은 1 : 2 : 3
- 오차 0.05를 1 : 2 : 3 으로 나눈다
- $A' = 0.0083$
- $B' = 0.0167$
- $C' = 0.025$
- $A' + B' + C' = 0.05$

- 역전파 시 어떻게 가중치를 조절하는가?

- 조절 방법

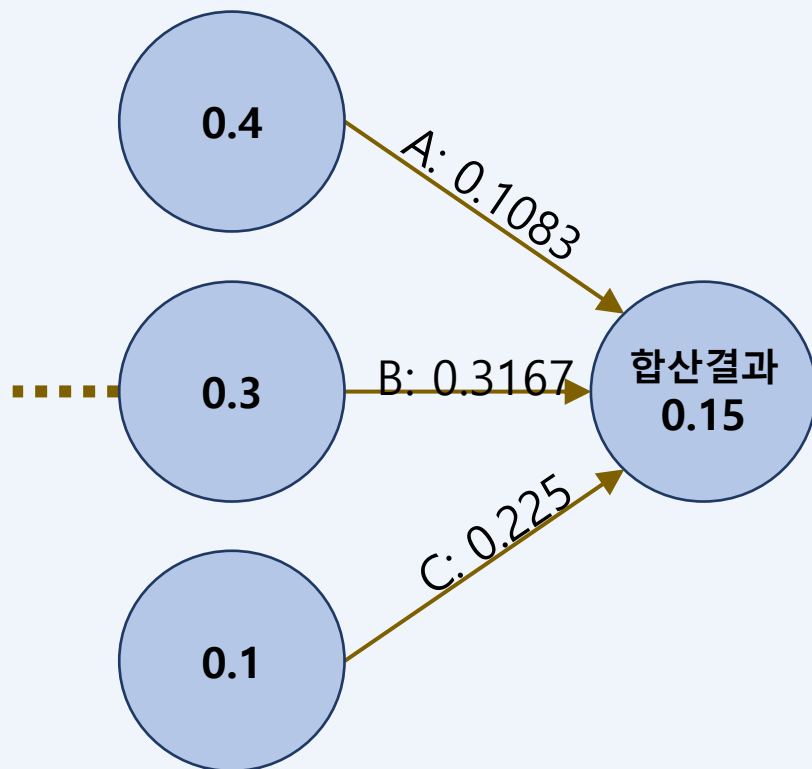


되돌려준 후 반영한다

- $A = 0.1 + 0.0083 = 0.1083$
- $B = 0.3 + 0.0167 = 0.3167$
- $C = 0.2 + 0.025 = 0.225$

- 역전파 시 어떻게 가중치를 조절하는가?

- 조절 방법



## 반영 후 다시 계산

- $A = 0.1 + 0.0083 = 0.1083$
- $B = 0.3 + 0.0167 = 0.3167$
- $C = 0.2 + 0.025 = 0.225$
- $0.4 \times A + 0.3 \times B + 0.1 \times C$   
 $= 0.04332 + 0.09501 + 0.0225 = 0.16083$   
→ 기대 값 0.2와 비교하여 → 오차 0.03917  
→ 기존의 오차 0.05보다 줄어듦

# Back Propagation (역전파) 알고리즘



- Back Propagation (역전파) 알고리즘을 구현한 다층 신경망에서
- 그 층을 훨씬 많이 만들어서
- 수 많은 분류 작업을 수행할 수 있게 한다면?
- 다층 신경망(Multilayer Neural Network)
  - 심층 신경망(Deep Neural Network) 으로 진화
- 심층 신경망을 이용한 학습 모델 = 딥 러닝(Deep Learning)

- 그럼 여기서... 각 퍼셉트론에서 임계 값을 넘으면 활성화는 어떻게 하나?
- 활성화 함수(Activation Function)
  - 신경망을 구성하는 각 퍼셉트론에서 임계 값을 넘었을 때 출력을 처리하는 함수
  - 함수의 정의
    - 입력: 이전 층의 디바이스 또는 퍼셉트론들로부터 전달되는 데이터
    - 함수의 동작: 입력 값의 합산 + 합산결과와 임계 값의 비교 + 출력 결정  
(활성화 조건)
    - 출력: 퍼셉트론 층의 연산 결과값. 다음 층의 뉴런에 대한 입력 또는 최종 층의 출력

- 그런데 활성화 함수는 왜 필요한가?
  - 피부, 눈과 같은 감각기관이 어떤 자극을 받아 신호를 발생시키면
  - 그 신호는 축삭을 통해서 이동하고
  - 축삭의 말단에 있는 시냅스를 거쳐
  - 다음 뉴런으로 전달

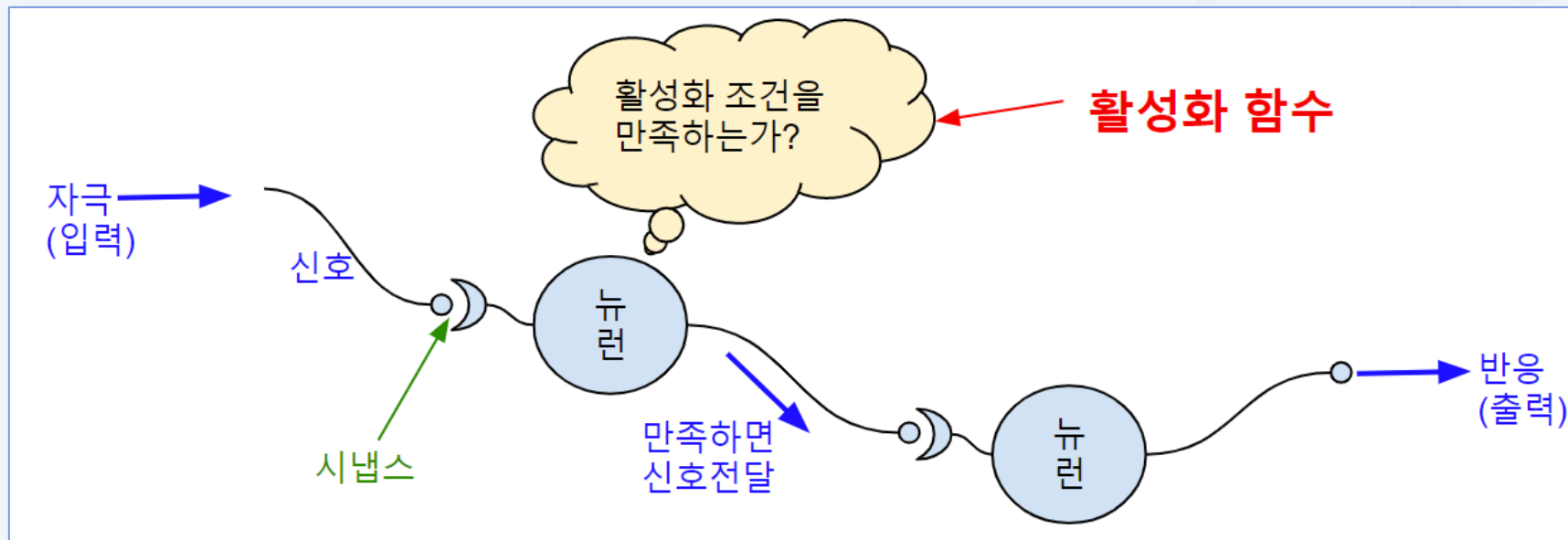


- 그런데!!!

- 전달되는 모든 신호(아주 미세한 신호부터 강한 신호까지)를 모두 다음 뉴런으로 전달한다면?
  - 생활 자체가 어려워짐
  - 매우 비 효율적
- 우리 몸에서 반응할 필요가 있는 수준까지만 신호를 전달하고 나머지의 신호는 무시한다!!
  - 진화의 결과
  - 이 기준을 모델에 반영한 것이 활성화 함수

- 즉, 활성화 함수란

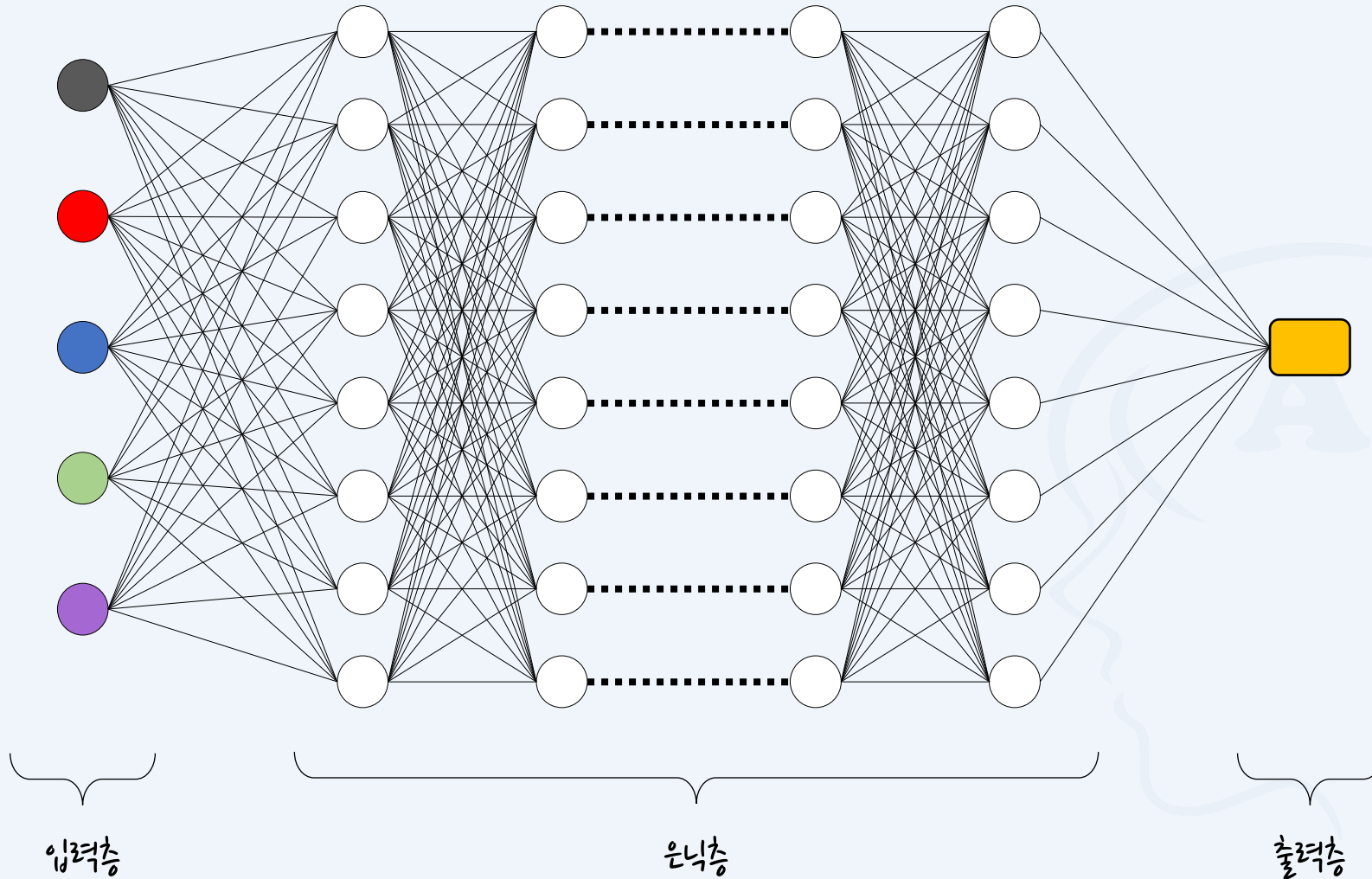
- 뉴런의 신호 흐름을 모델링 할 때 각각의 뉴런에 제한을 걸어 둔 것
- 활성화 함수에서 적용한 기준에 따라 조건을 만족하는 경우에만 다음 뉴런으로 신호 전달





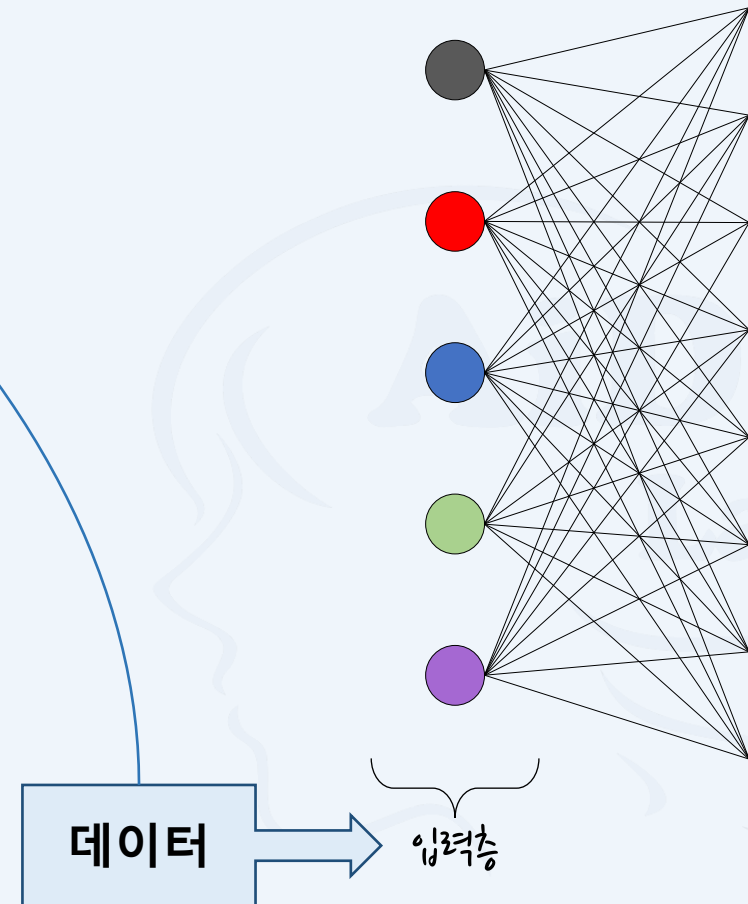
- 그럼 활성화 함수는 왜 그렇게 많은 형태가 존재하는가?
  - 우리는 아직 우리의 뇌와 신경들이 어떻게 동작하고 서로 어우러지는지 정확하게 알지 못함
  - 따라서 우리가 AI로 해결하고자 하는 문제에 맞게 가장 효율적이고 적절한 활성화 함수를 계속 연구, 개발하여 활용하는 것

# 가장 기본적인 딥러닝 모델: 전체 구조



# 가장 기본적인 딥러닝 모델: 입력층

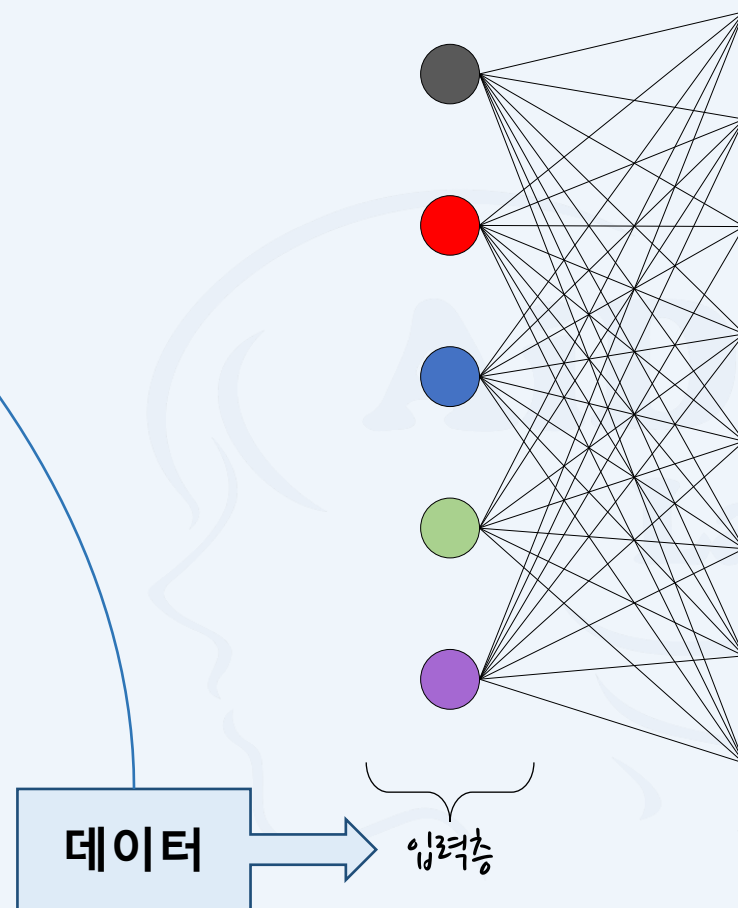
어떤 데이터들이 입력되는가?



# 가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...
  
- SNS 데이터
- Web Scraping 데이터
- 등...



# 가장 기본적인 딥러닝 모델: 입력층

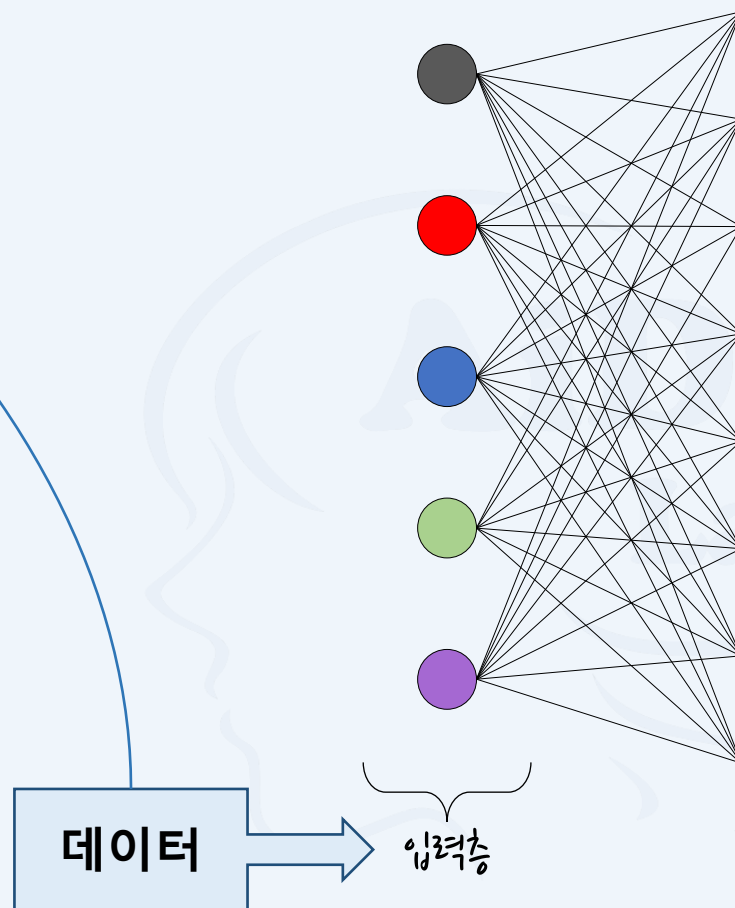
어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...

수치 데이터

- SNS 데이터
- Web Scraping 데이터
- 등...

문자(열) 데이터 → 수치화



# 가장 기본적인 딥러닝 모델: 입력층

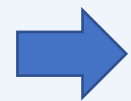
어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...

수치 데이터

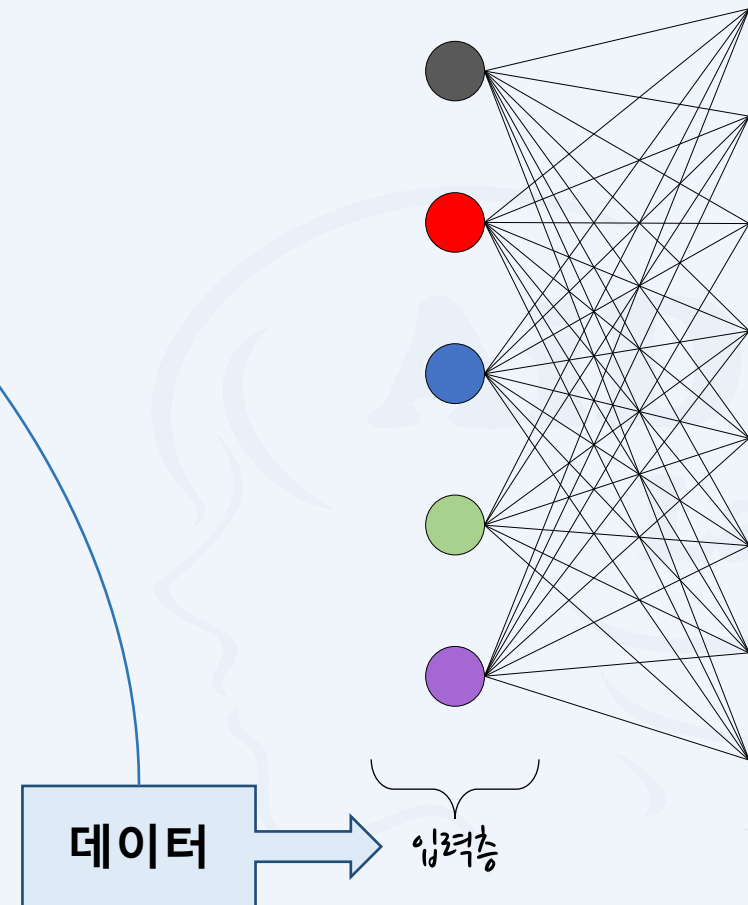
- SNS 데이터
- Web Scraping 데이터
- 등...

문자(열) 데이터 → 수치화



딥러닝에서 사용되는 데이터는

기본적으로 수치 데이터



# 가장 기본적인 딥러닝 모델: 입력층

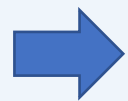
어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...

수치 데이터

- SNS 데이터
- Web Scraping 데이터
- 등...

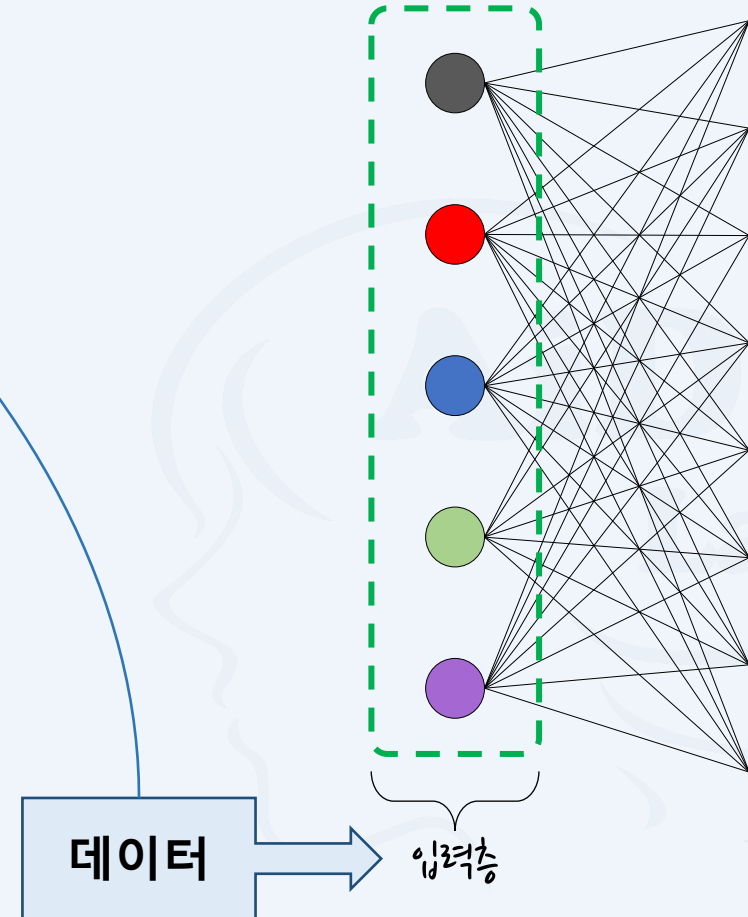
문자(열) 데이터 → 수치화



딥러닝에서 사용되는 데이터는

기본적으로 수치 데이터

각 노드에 하나씩 입력  
→ 한 줄로 이어진 데이터



# 가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...

수치 데이터

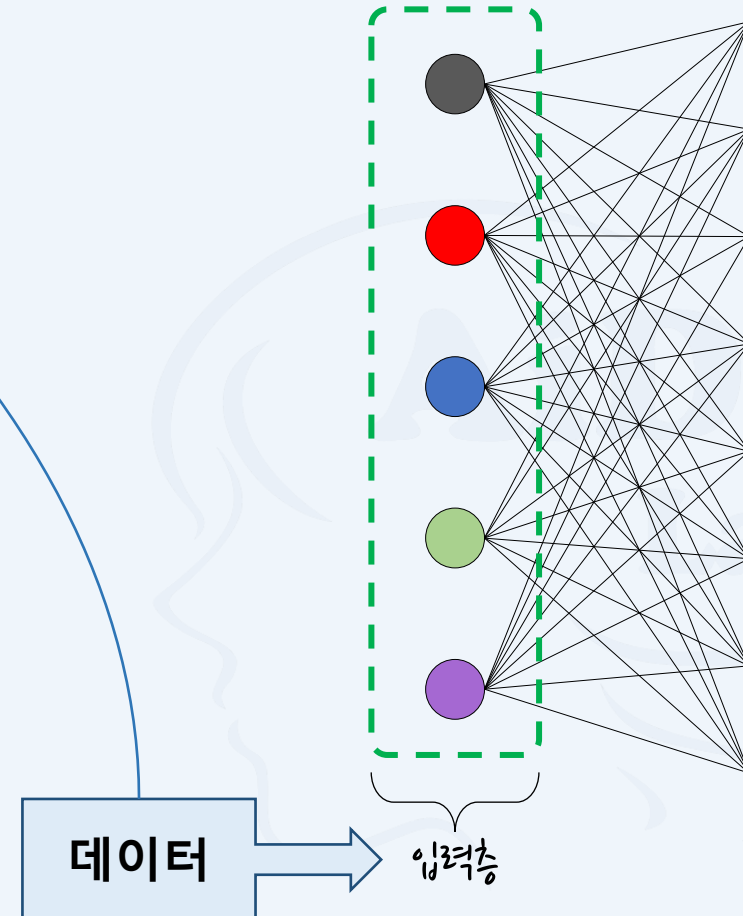
- SNS 데이터
- Web Scraping 데이터
- 등...

문자(열) 데이터 → 수치화

➡ 딥러닝에서 사용되는 데이터는  
기본적으로 수치 데이터

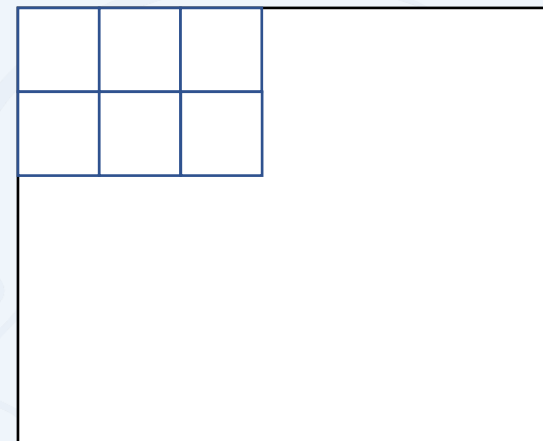
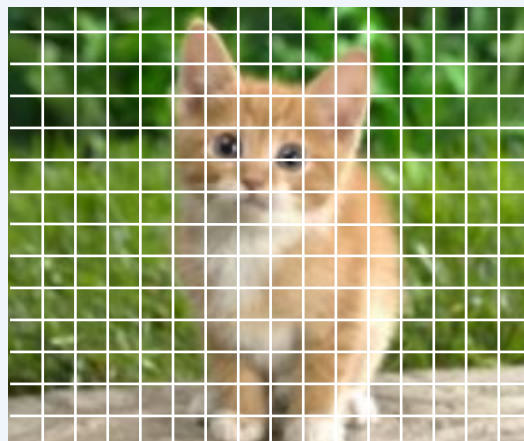
➡ 1차원 배열로 이루어진 수치 데이터

각 노드에 하나씩 입력  
→ 한 줄로 이어진 데이터



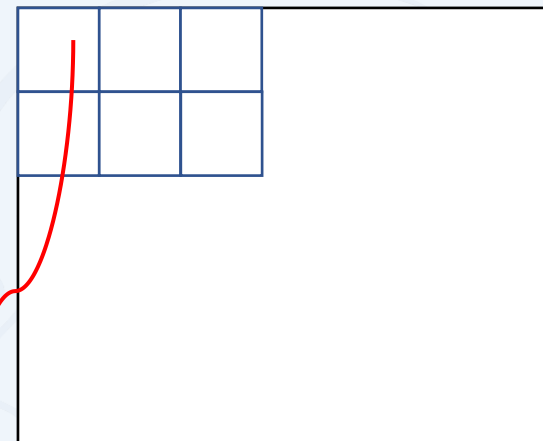
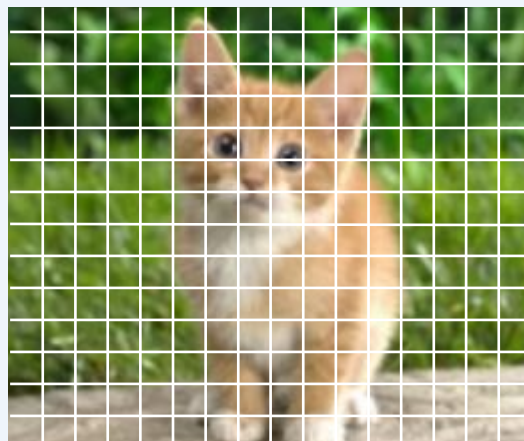


- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



# 가장 기본적인 딥러닝 모델: 입력층

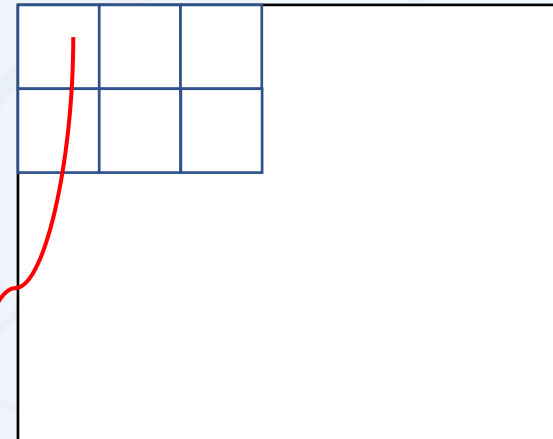
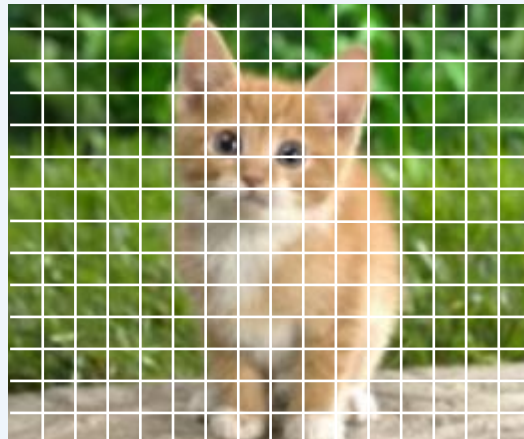
- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



32832 = #008040 = #00 #80 #40 = R G B =

# 가장 기본적인 딥러닝 모델: 입력층

- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



➡ 배열 데이터를 1차원으로 변환하여 입력 데이터로 사용  $32832 = \#008040 = \#00 \#80 \#40 =$  R G B

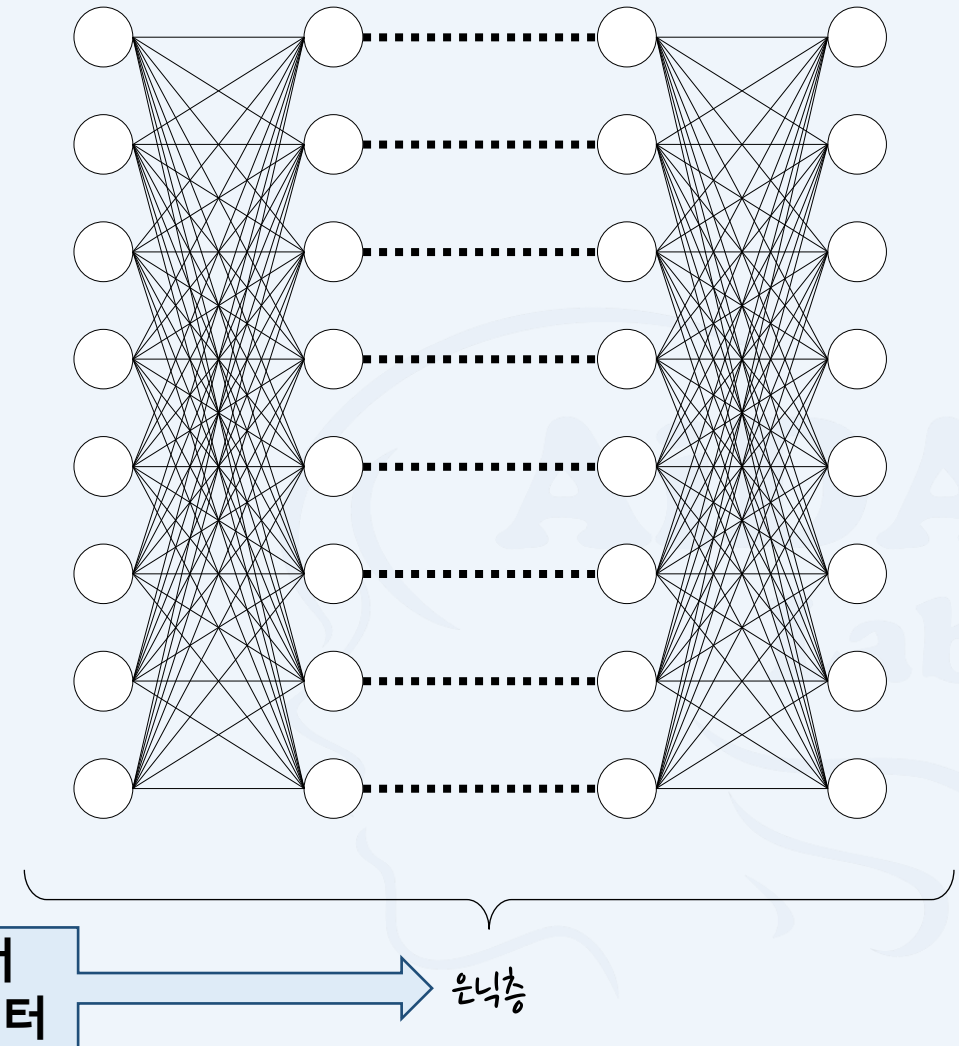
➡ 동영상 데이터는 다수의 이미지가 순서대로 연결된 것

# 가장 기본적인 딥러닝 모델: 은닉층

데이터는 어떻게 전달되는가?

- 입력층에서 입력된 데이터는
- 각 데이터가 첫번째 층의 모든 뉴런에
- 동일하게 전달됨
- 각 층의 모든 뉴런이 가진 데이터는
- 다음 층의 모든 뉴런에
- 동일하게 전달됨

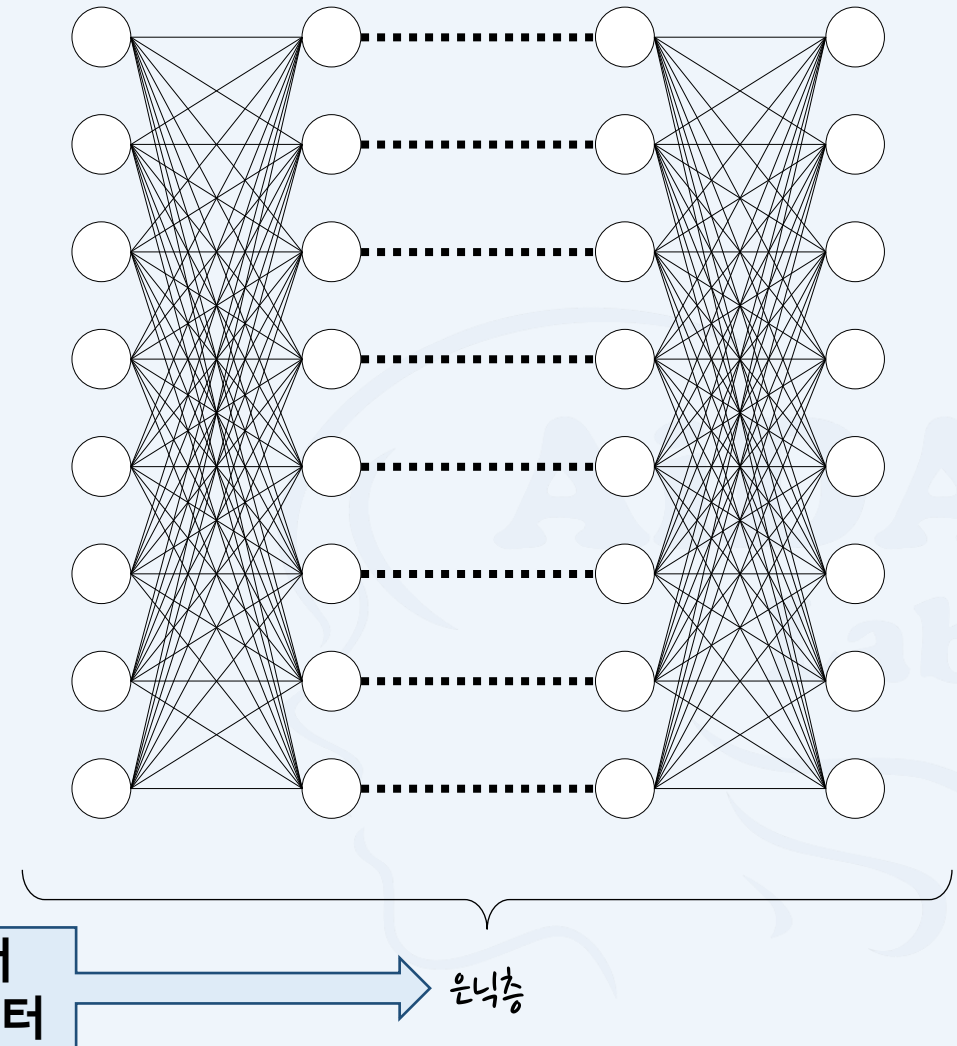
**실제의 신경망에서는 신경전달물질이 퍼져 나가는 범위 안의 신경세포에만 신호가 전달됨 (모델링의 한계)**



# 가장 기본적인 딥러닝 모델: 은닉층

각 뉴런에 전달된 데이터는 어떻게 가공되는가?

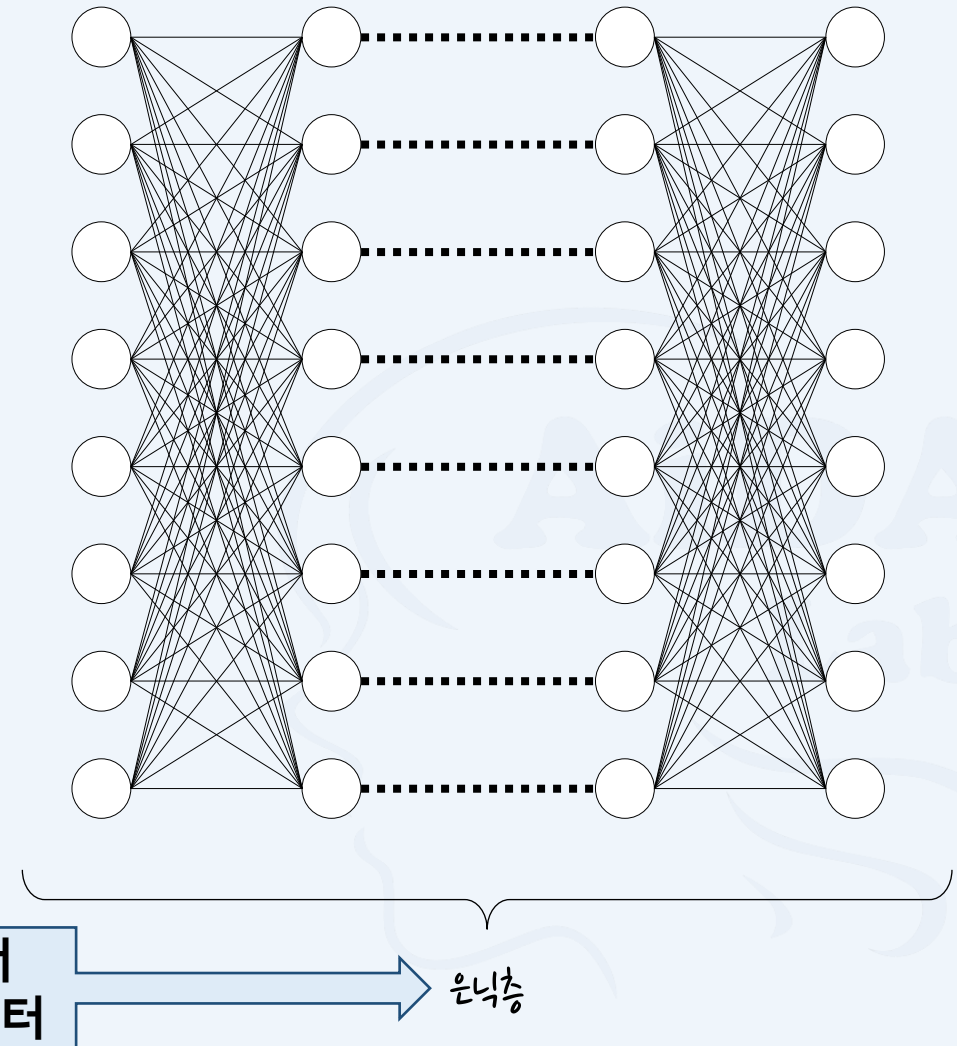
- 각 층의 뉴런은
- 입력되는 모든 데이터를
- 다 더한 후(합산)
- 합산 결과를 활성화 함수에 적용하고
- 활성화 함수(F1) 적용 결과를
- 다음 층의 뉴런으로 전달



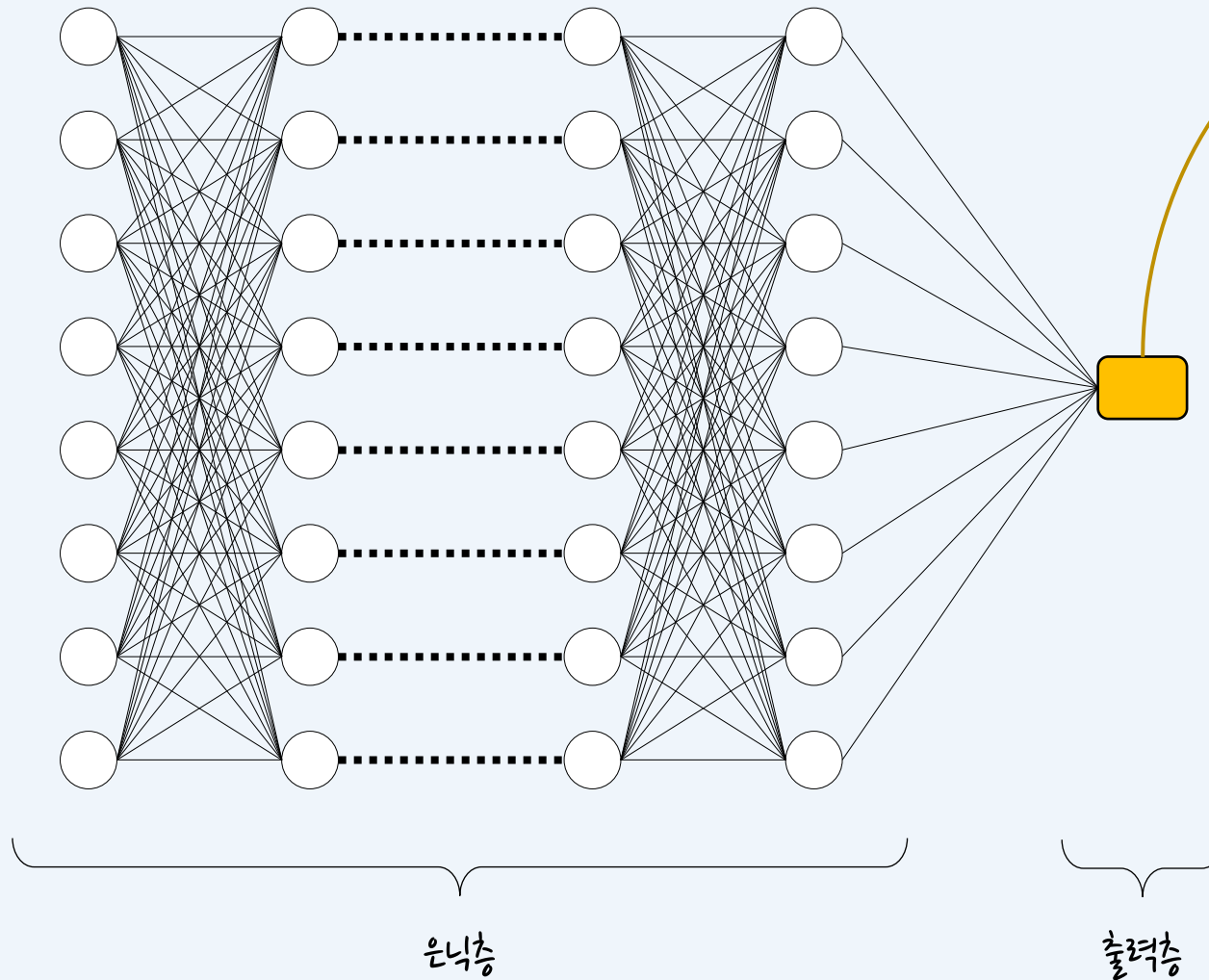
# 가장 기본적인 딥러닝 모델: 은닉층

은닉층은 데이터에 어떤 작업을 하는가?

- 각 층에 입력된 데이터는
  - 다음 층의 뉴런으로 가는 통로에 설정된 가중치를
  - 현재의 뉴런이 가진 데이터에 곱하여
  - 그 결과를 다음 층의 뉴런으로 전달
- 
- 모든 층의 데이터 이동에서
  - 가중치의 곱셈과 입력 값의 합산은
  - 동일하게 적용됨



# 가장 기본적인 딥러닝 모델: 은닉층

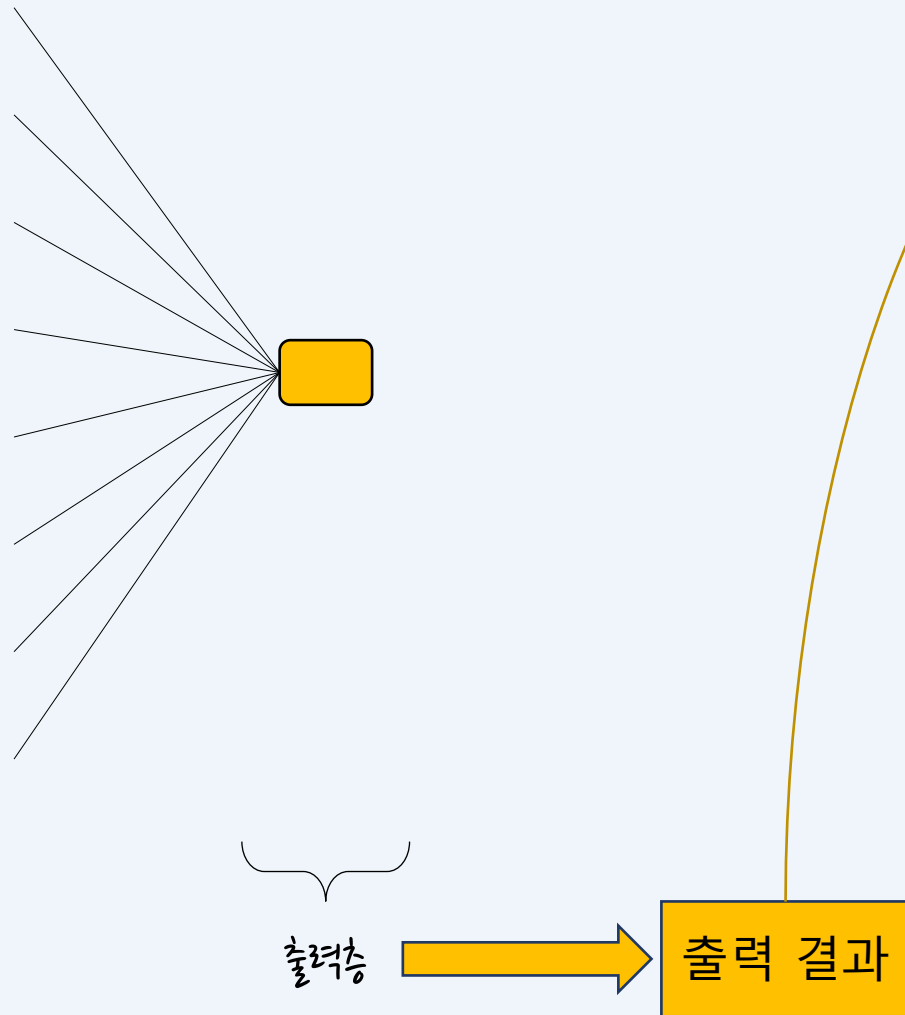


은닉층과 출력층은 함께 어떤 작업을 하는가?

- 은닉층의 마지막 층의 뉴런들은
  - 자기에게 입력된 데이터를
  - 합산 후
  - 활성화 함수(F1)를 적용하고
  - 적용 결과를
  - 출력층의 출력 뉴런으로 전달
- 
- 출력 뉴런은
  - 전달된 모든 데이터를
  - 순서대로 나열하여(벡터화)
  - 활성화 함수(F2)를 적용하고
  - 적용 결과(벡터값)를
  - 결과로서 출력함



# 가장 기본적인 딥러닝 모델: 출력층

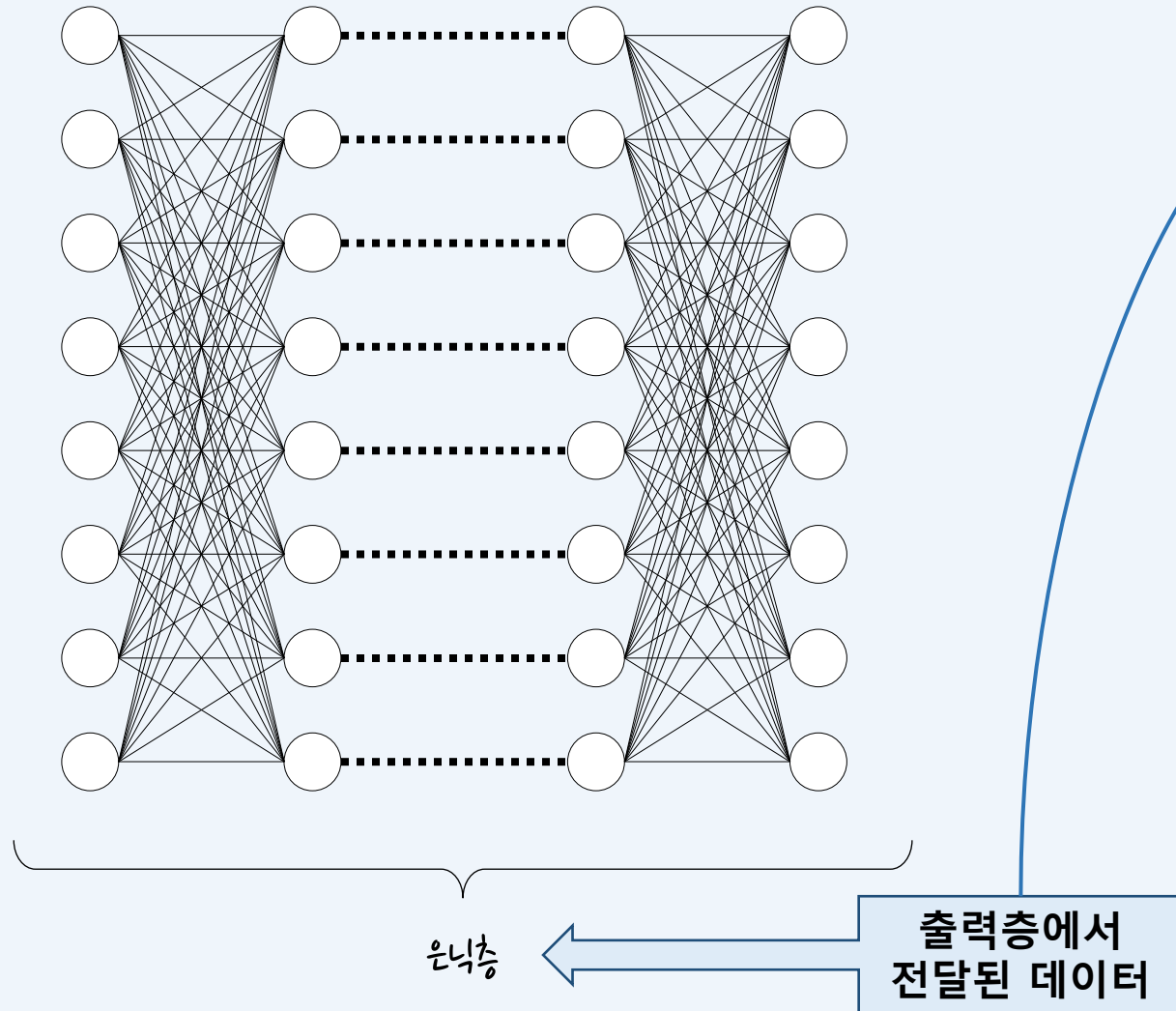


출력 층에서는 어떤 결과가 나오는가?

- 출력 뉴런은
- 전달된 모든 데이터를
- 순서대로 나열하여(벡터화)
- 활성화 함수(F2)를 적용하고
- 적용 결과(벡터값)를
- 결과로서 출력함
- 출력층의 결과는
- 미리 입력된 기대 결과값과 비교하여(지도학습)
- 일치하면 → 학습 완료
- **일치하지 않으면 → 진행의 반대 방향으로 다시 전달**



# 가장 기본적인 딥러닝 모델: 은닉층



결과층을 거치고 다시 돌아온 데이터는 은닉층에서 또 어떤 작업을 하는가?

- 오류 갱신(보정) 작업 수행
  - 많이 사용하는 알고리즘
    - 경사 하강법 (Gradient Descent)  
( =확률론적 경사 하강법 (Stochastic GD) )
    - 모멘텀 (Momentum)
    - 경사 하강법 + 모멘텀
    - 배치 방식 (Batch)
    - 미니 배치 방식: SGD + Batch
    - AdaGrad
    - RMSprop
    - Adam (RMSprop + Momentum)
- 등...

# 신경망은 어떻게 발전해야 하는가?

- 현재의 모델로도 충분히 효과적임
- 만약 충분한 컴퓨팅 파워, 데이터가 존재한다면
- 극도로 단순화 된 현재의 신경망 모델을 실제 생물학적 신경망과 더욱 유사하게 모델링 할 수 있을까?
- 더욱 유사한 모델은 현재보다 복잡할 것
- 복잡해진 만큼 효율성이나 가치의 상승을 기대할 수 있는가?
- 현재의 모델로는 강 인공지능으로 진행하기는 매우 어려워 보임
- 해결책은 무엇일까? 등등..