1. 인공지능 역사

인공지능

- 지능적인 요소가 포함된 모든 기술 총칭
- 인공지능>머신러닝>딥러닝
- 단층 신경망 → 다층 신경망(DNN) → CNN, RNN 등

1943년
워렌 맥클록과 월터 피츠, 전기 스위치처럼 켜고 끄는 기초기능의 인공신경을 그물망 형태로 연결하면 사람의 뇌에서 동작하는 아주 간단한 기능을 흉내낼 수 있음을 증명

1950년

앨런 튜링, 기계가 인간과 얼마나 비슷하게 대화할 수 있는지를 기준으로 기계에 지능이 있는지를 판별하는 튜링 테스트 제안

1956년

다트머스 회의에서 인공지능 용어 처음 사용. "학습의 모든 면 또는 지능의 다른 모든 특성을 기계로 정밀하게 기술할 수 있고 이를 시뮬레이션할 수 있다"

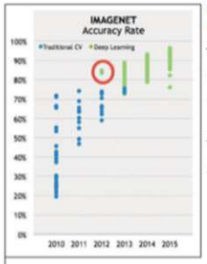
1958년
프랭크 로센블래트, 뇌신경을 모사한 인공신경 뉴런 '퍼셉트론' 제시

1970년대
AI 연구가 기대했던 결과를 보여주지 못하자 대규모 투자가 중단되며 암흑기 도래

1980년대
전문가들의 지식과 경험을 데이터베이스화해 의사결정 과정을 프로그래밍화한 '전문가 시스템' 도입. 그러나 관리의 비효율성과 유지·보수의 어려움으로 한계

1997년
IBM 딥블루, 체스 챔피언 개리 카스파로프와의 체스 대결에서 승리

2006년
제프리 힌튼 토론토대 교수, 딥러닝 알고리즘 발표

2012년


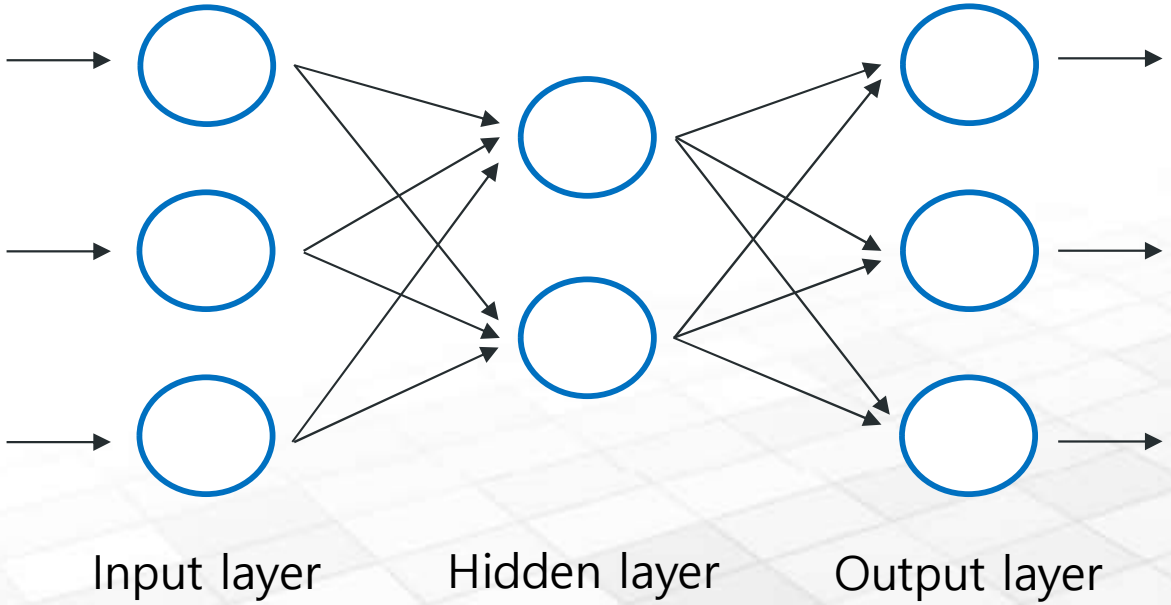
국제 이미지 인식 경진대회 '이미지넷'에서 딥러닝 활용한 팀이 우승하며 획기적 전환점

2014년
구글, 딥마인드 인수

2016년
구글 알파고, 이세돌에게 승리


1. 신경망 구조

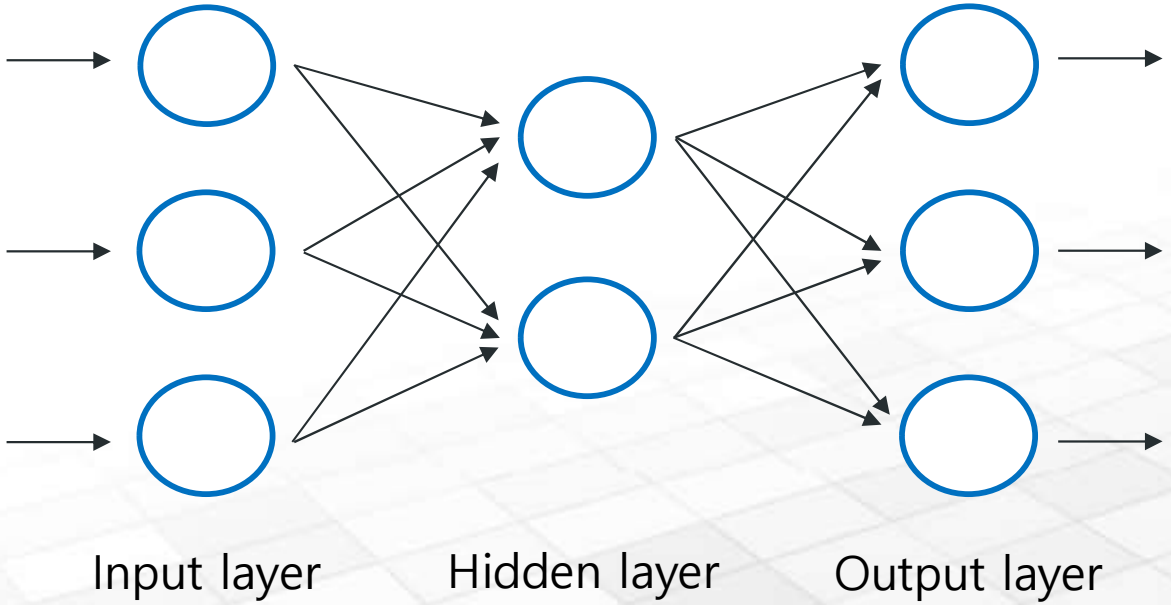
Layer	
Hidden layer 수	**신경망 이름**
0개	단층 신경망
1개	얕은 신경망
2개 이상	다층 신경망



1. 신경망 구조

분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수



1. 신경망 구조

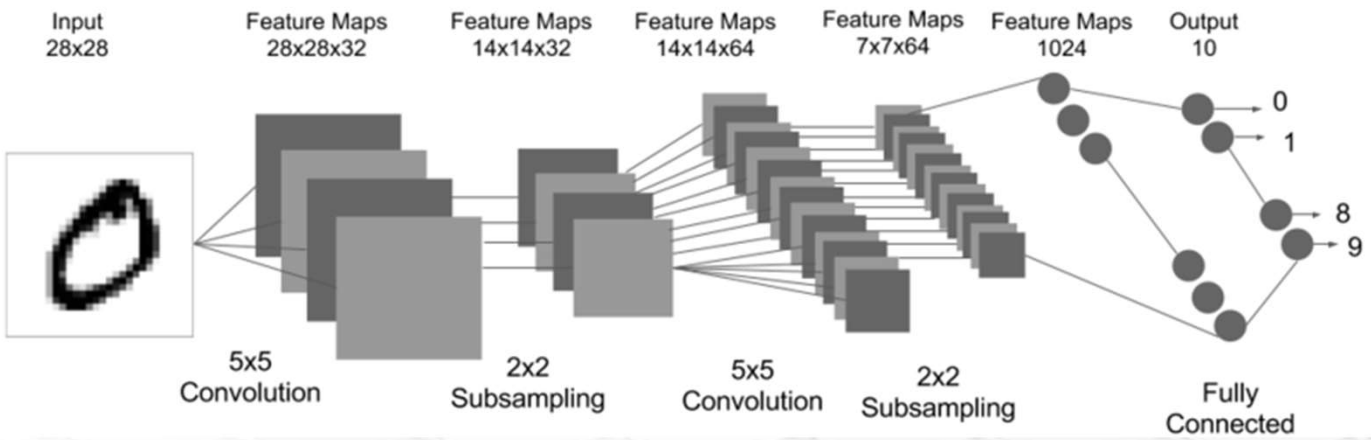
분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수
분류	이진분류	
	다중분류	
회귀		

1. 신경망 구조

분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수
분류	이진분류	
	다중분류	범주 수만큼
회귀		



1. 신경망 구조

분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수
분류	이진분류	
	다중분류	범주 수만큼
회귀		

1. 신경망 구조

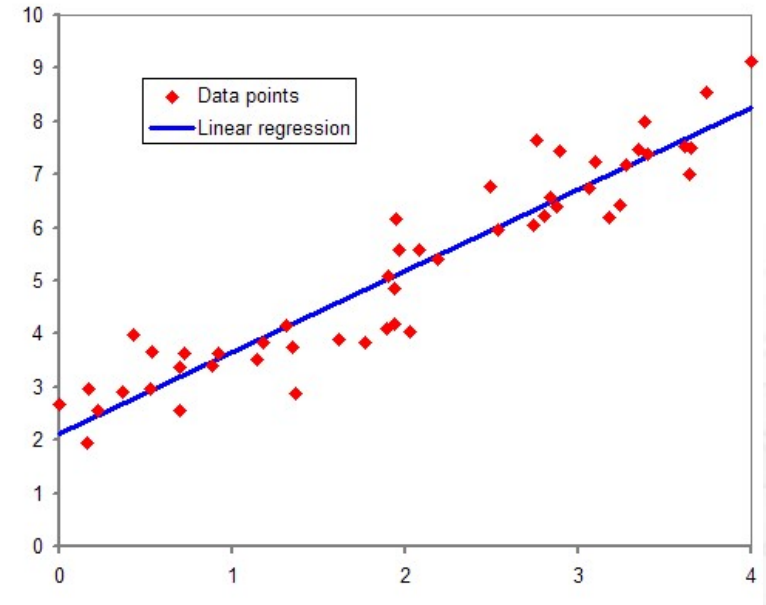
분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수
분류	이진분류	1개
	다중분류	범주 수만큼
회귀		

1. 신경망 구조

분류와 회귀에서 출력층 노드 수

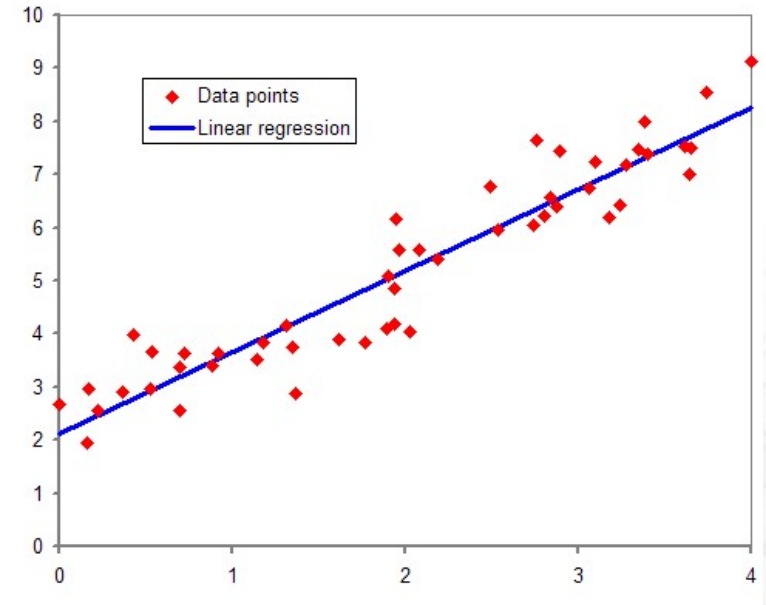
지도 학습		출력층 노드 수
분류	이진분류	1개
	다중분류	범주 수만큼
회귀		



1. 신경망 구조

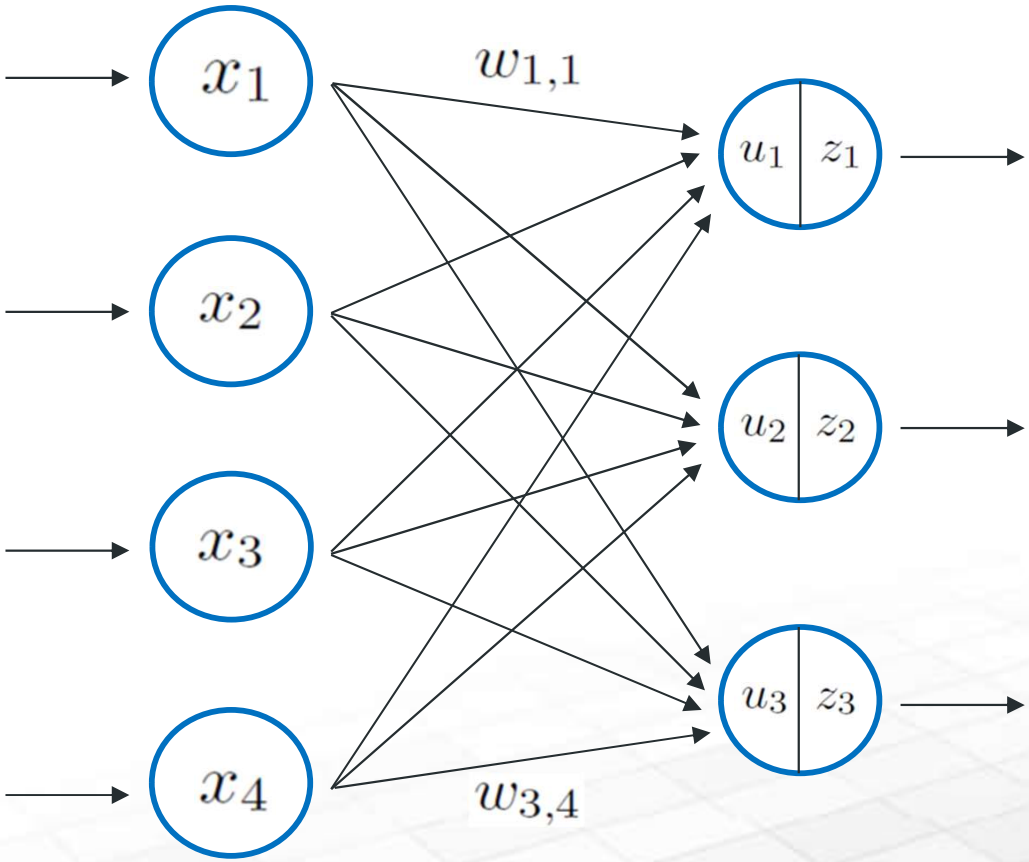
분류와 회귀에서 출력층 노드 수

지도 학습		출력층 노드 수
분류	이진분류	1개
	다중분류	범주 수만큼
회귀		1개



2. 단층 신경망

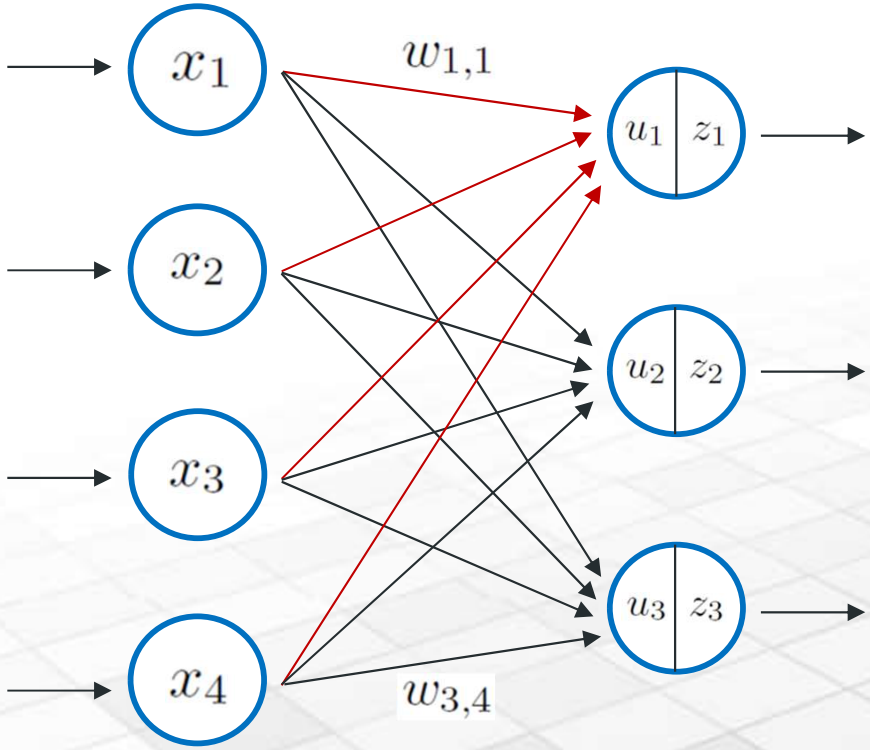
Weight



2. 단층 신경망

Weight

$$\mathbf{w} = \begin{bmatrix} \text{첫번째노드의 가중치} \\ \vdots \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ \vdots \end{bmatrix},$$



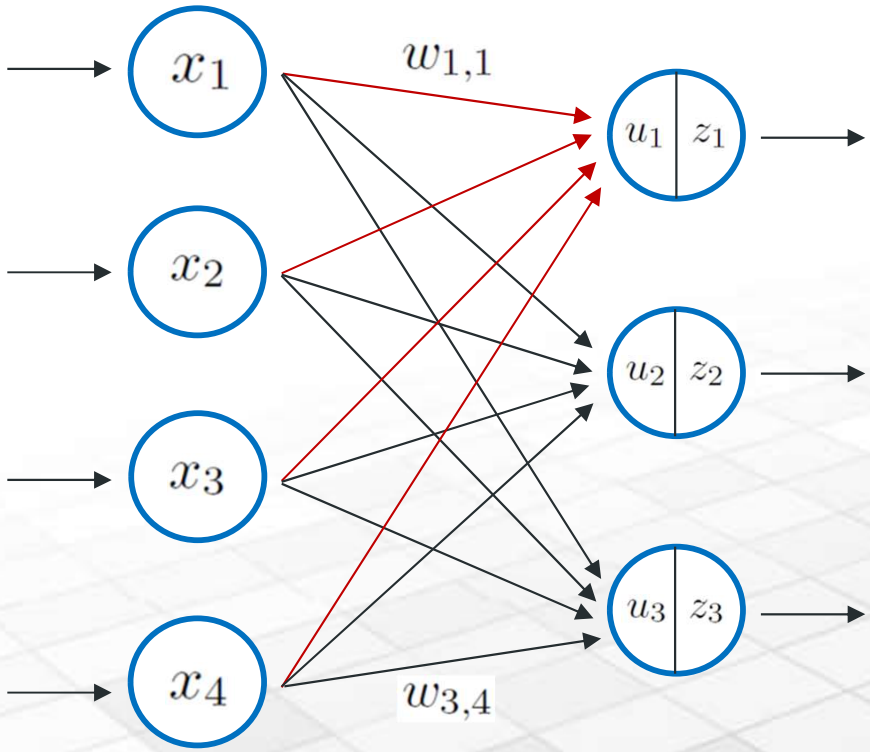
2. 단층 신경망

단층 신경망

$$\mathbf{w} = \begin{bmatrix} \text{첫번째노드의 가중치} \\ \vdots \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ \vdots \end{bmatrix},$$

$$\mathbf{u} = \mathbf{w}\mathbf{x},$$

$$\mathbf{z} = f(\mathbf{u})$$

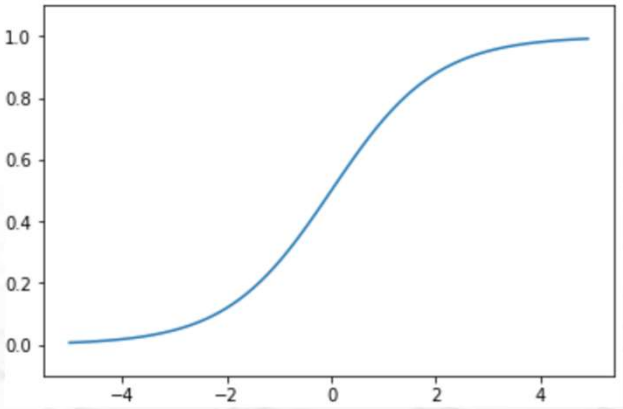


2. 단층 신경망

Activation function

문제 유형	출력층 활성함수
회귀	항등함수
이진분류	시그모이드함수
다중분류	소프트맥스함수

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



2. 단층 신경망

Activation function

문제 유형	출력층 활성함수
회귀	항등함수
이진분류	시그모이드함수
다중분류	소프트맥스함수

$$f(u_i) = \frac{e^{u_i}}{\sum_{j=1}^{J} e^{u_j}}$$

2. 단층 신경망

Cost function

문제 유형	출력층 활성함수	비용 함수
회귀	항등함수	오차제곱식
이진분류	시그모이드함수	교차엔트로피
다중분류	소프트맥스함수	교차엔드토피

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{d}_n - \mathbf{y}(\mathbf{x}_n;\mathbf{w}))^2$$

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{j=1}^{J} d_{nj}\log y_j(\mathbf{x}_n;\mathbf{w})$$

2. 단층 신경망

지도 학습		출력층 노드 수	출력층 활성함수	비용함수
분류	이진분류	1개	시그모이드함수	교차엔트로피
	다중분류	범주 수만큼	소프트맥스함수	교차엔트로피
회귀		1개	항등함수	오차제곱식

2. 단층 신경망

Optimizer

- 학습의 목표는 비용함수에 최솟값을 주는 가중치를 구하는 것
- 비용함수를 최소화하는 방법: Gradient Descent, RMSProp, Adam 등
- 경사 하강법(Gradient Descent)은 현재의 가중치를 음의 기울기 방향으로 움직이는 것을 반복

$$\nabla E_n(\mathbf{w}) = \begin{bmatrix} \frac{\partial E_n(\mathbf{w})}{\partial w_{11}} & \cdots & \frac{\partial E_n(\mathbf{w})}{\partial w_{I1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial E_n(\mathbf{w})}{\partial w_{J1}} & \cdots & \frac{\partial E_n(\mathbf{w})}{\partial w_{JI}} \end{bmatrix},$$

$$\mathbf{w} := \mathbf{w} - \eta \nabla E_n(\mathbf{w})$$

2. 단층 신경망

단층 신경망 학습 방법

- 1. 가중치를 임의로 초기화한다.
- 2. 훈련 샘플에 대해서 출력값을 계산한다.
- 3. 비용함수를 이용하여 가중치를 갱신한다.

$$\mathbf{w} := \mathbf{w} - \eta \nabla E_n(\mathbf{w})$$

- 4. 업데이트한 가중치로 다음 훈련 샘플에 대해서 출력값을 계산한다.