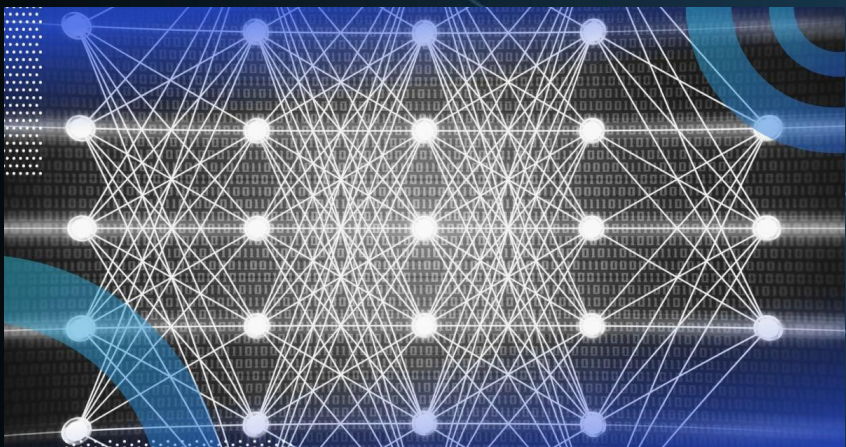




Neural Network



Neural Network

딥러닝 ⊂ 머신러닝 ⊂ 인공지능

인공지능 | Artificial Intelligence

사람의 지적 능력을 컴퓨터를 통해 구현하는 기술

머신러닝 | Machine Learning

사람이 정한 모델과 특징 추출 방법을 이용하여
데이터를 기반으로 학습해서 추론할 수 있게 하는 기술

딥러닝 | Deep Learning

인공신경망 방법을 이용해 만든 머신러닝 기술로,
빅데이터 학습에 적합한 기술

인공지능(AI)

컴퓨터에서 음성 및 작성된 언어를 보고 이해하고 번역하고
데이터를 분석하고 추천하는 기능을 포함하여
다양한 고급 기능을 수행할 수 있는 일련의 기술

기계학습

데이터를 제공하여 명시적으로 프로그래밍하지 않고
신경망과 딥 러닝을 사용하여 시스템이 자율적으로
학습하고 개선하는 과정

딥 러닝

여러 '비선형 변환기법'의 조합을 통해 높은 수준의 추상화를
시도하는 기계 학습 알고리즘의 집합
인공신경망 + 여러 층의 히든레이어

Neural Network

전통적인 ML 알고리즘 vs 딥러닝

▪ 딥러닝

- > 신경망 (neural networks) based
- > 데이터에서 자동으로 복잡한 특징을 학습
- > 복잡한 데이터 구조와 패턴 학습 가능. **비선형성**
- > 일반적으로 성능이 더 좋다고 알려져 있음
 - 데이터의 크기와 특성 등에 따라 다름
 - 보통 비정형 데이터에 대한 성능 우수

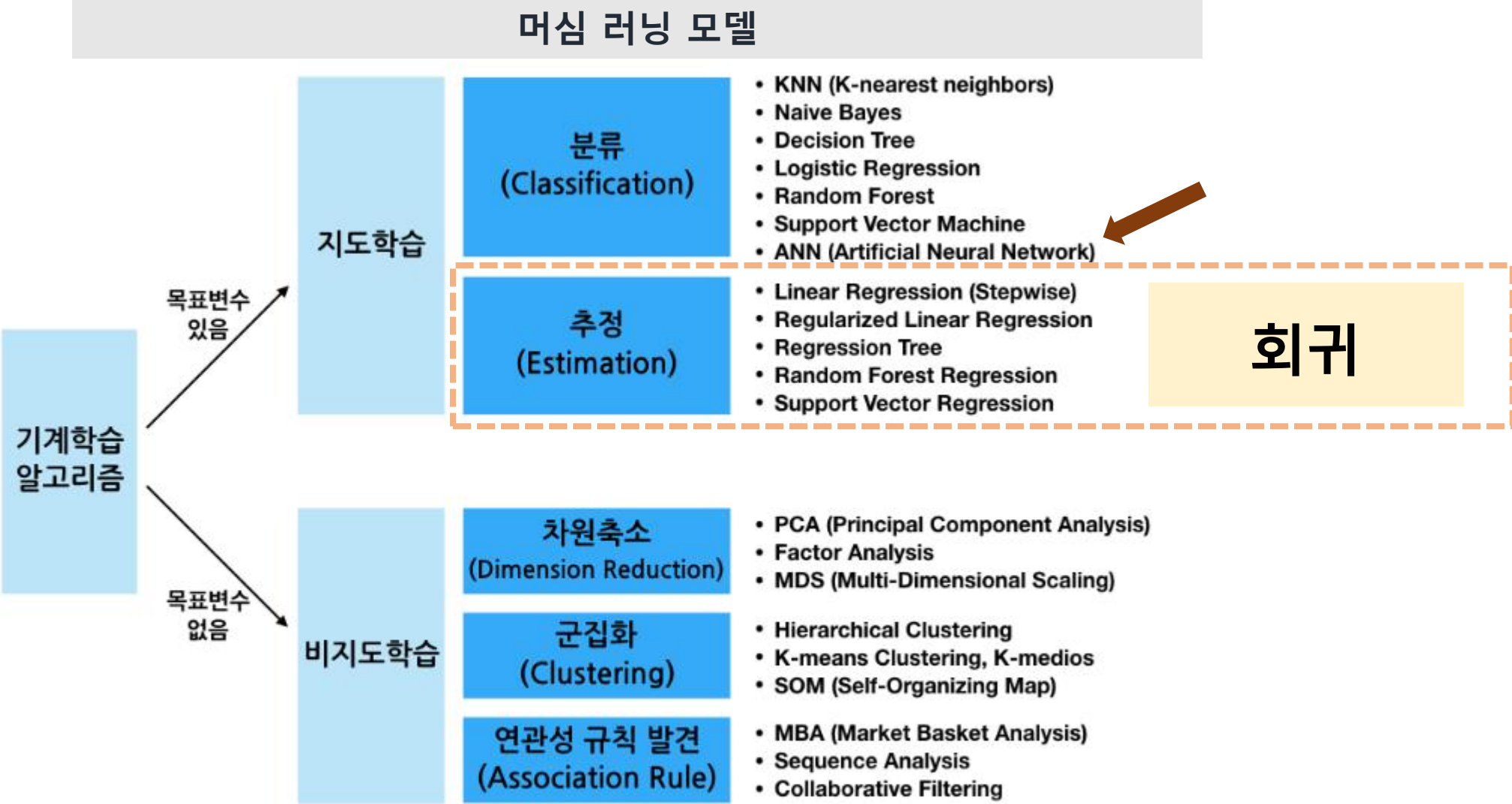
▪ 전통적인 ML 알고리즘

- > 상대적으로 간단한 데이터 구조에서 잘 작동하며, 데이터의 **선형성**, 분포 등에 대한 가정을 기반
- > 주로 구조화된 데이터에 적용
- > 특징 추출(feature extraction) 과정에서 도메인 지식이 중요.
- > 모델의 성능은 선택된 특징의 질에 크게 의존

선형 *vs* 비선형

deeplearning.linear.nonlinear

Neural Network



Neural Network

인공 신경망

- 하나의 독립된 모델
- 머신러닝 및 인공지능 기술의 범주에 속하는 방법론과 기술

유형	모델/기법	설명	학습 유형	모델 유형
신경망	인공 신경망 (Neural Networks)	입력 데이터로부터 복잡한 패턴을 학습하여 예측 이나 분류를 수행할 수 있는 모델. 다양한 층 (layer)과 노드(node)로 구성되며, 각 연결마다 가중치(weight)가 존재	지도 학습 /비지도 학습 /강화 학습	예측/묘사
	컨볼루션 신경망 (Convolutional Neural Networks, CNN)	이미지 인식, 비디오 분석, 이미지 분류 등 시각적 데이터를 처리하는데 특화된 신경망 구조. 로컬 패 턴을 효과적으로 학습	지도 학습	예측
	순환 신경망(Recurrent Neural Networks, RNN)	시계열 데이터, 자연어 처리 등 순차적 데이터를 처리하는데 적합한 신경망 구조. 과거 정보를 현재 의 결정에 반영		
	자기 조직화 맵(Self- Organizing Maps, SOM)	고차원 데이터를 저차원(보통 2차원)으로 매핑하 여 시각화하고, 데이터의 패턴을 발견하는데 사용	비지도 학습	묘사
	생성적 적대 신경망 (Generative Adversarial Networks, GANs)	두 개의 신경망(생성자와 판별자)이 서로 경쟁하면 서 학습하는 구조로, 새로운 데이터를 생성하는 데 사용		예측/묘사

Neural Network

인공 신경망

- 하나의 독립된 모델
- 머신러닝 및 인공지능 기술의 범주에 속하는 방법론과 기술

1. 머신러닝 모델(Machine Learning Model):

데이터로부터 학습할 수 있는 능력을 가진 머신러닝 모델의 한 종류
지도 학습, 비지도 학습, 강화 학습 등 다양한 학습 방식 지원

2. 딥러닝 아키텍처(Deep Learning Architecture):

다층 퍼셉트론(Multilayer Perceptrons, MLP)이나 심층 신경망(Deep Neural Networks, DNN)과 같이
여러 층으로 구성된 신경망

3. 인공지능(AI) 기술:

인공지능 연구와 응용의 핵심 요소 중 하나로, 인공지능을 구현하는 기술

4. 계산 모델(Computational Model):

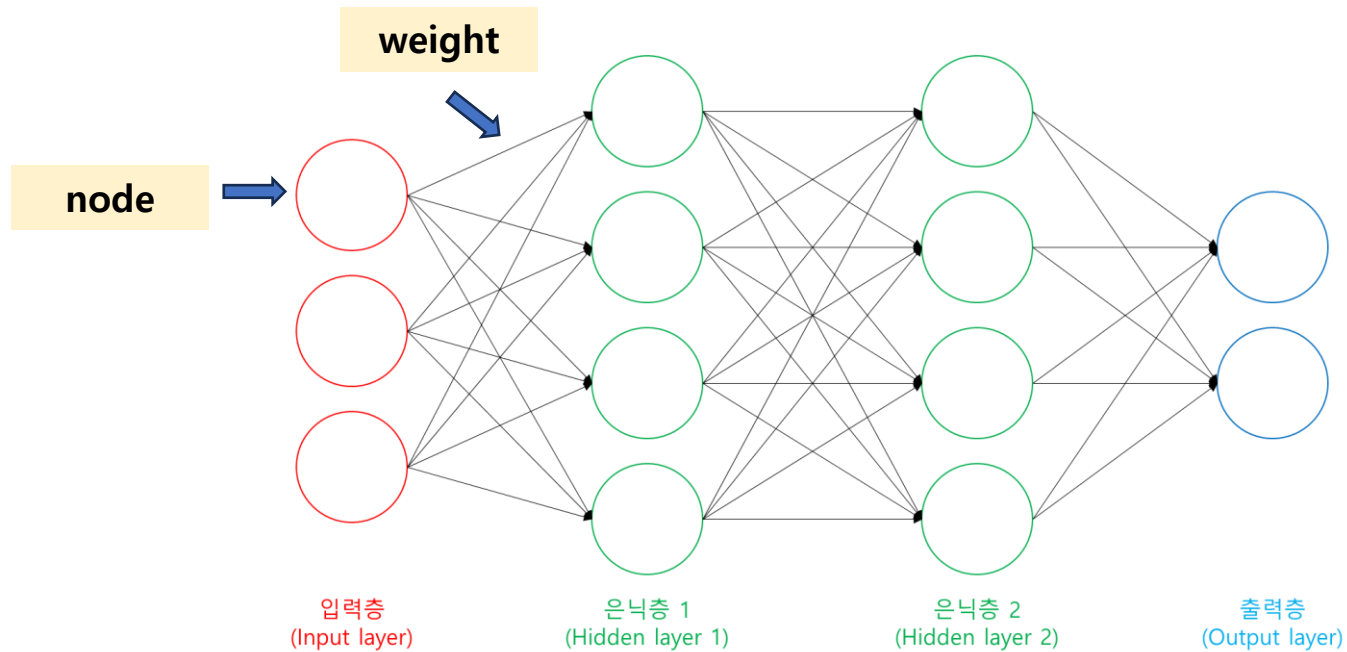
입력에서 출력으로의 매핑을 학습하는 계산 모델
데이터의 복잡한 관계의 모델링 및 예측에 사용

Neural Network

NN 구조

- 신경망 기본 구조

: Input layer, Hidden layer, Output layer



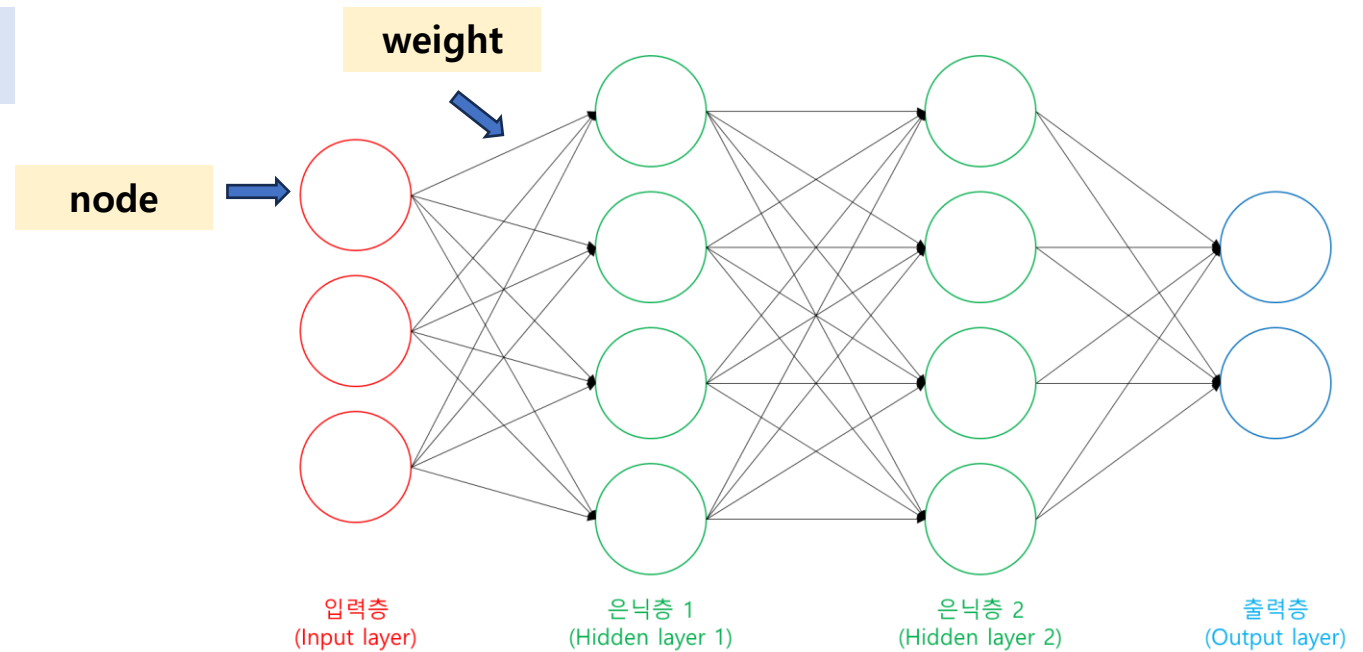
Neural Network

NN 구조

- 신경망 기본 구조

: Input layer, Hidden layer, Output layer

파라미터(Parameter) ???

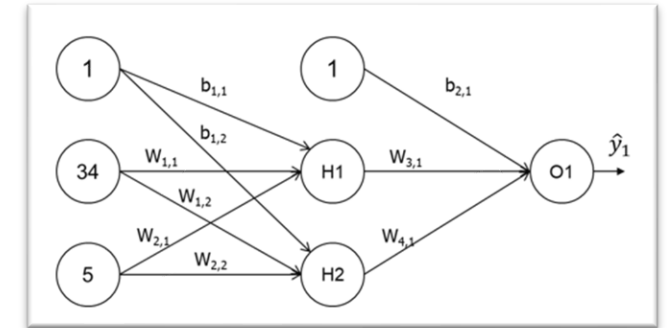


Neural Network

NN 구조

■ 신경망 기본 구조

- 입력층, 출력층 : 1개
- 은닉층의 수 ? -> 분석가의 설계(결정)의 영역
- 은닉층의 수가 여러 개 = 다층 신경망 (Deep NN) cf) shallow NN



■ 각 layer 역할

- 입력층: 독립변수의 값 (=features 정보)을 입력 받아 다음 층으로 전달
- 은닉층: 입력받은 데이터에서 종속변수의 값을 추론하는데 필요한 중요 특성 추출
- 출력층: 종속변수의 예측치 출력

Neural Network

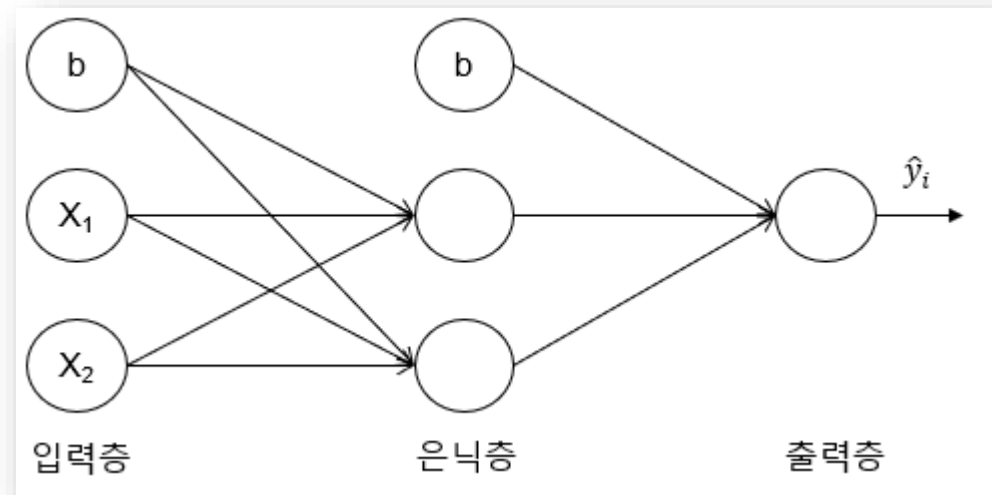
NN 구조

■ 각 층의 수

- 입력층 수 = ?
- 출력층 수 = ?
- 은닉층의 수 = ?

■ 각 층의 노드 수

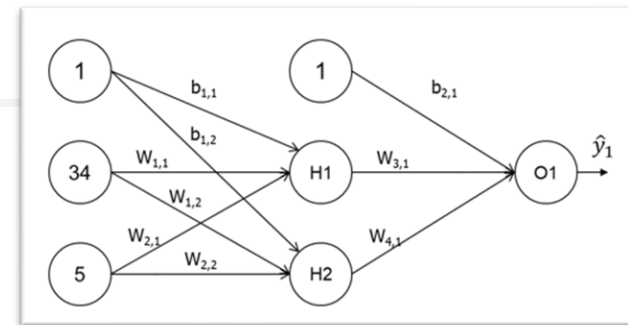
- 입력 노드의 수 = ?
- 출력 노드의 수 = ?
 - ✓ 회귀 : 출력 노드의 수 = ?
 - ✓ 분류 : 출력 노드의 수 = ?
- 은닉 노드의 수 = ?



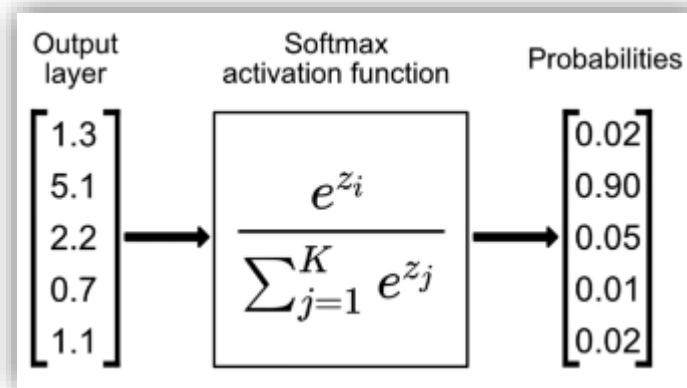
Neural Network

NN 구조

노드



- 각 노드는 이전 층으로 전달받은 값들을 입력 받은 후 다음 층의 노드로 전달 (출력)
 - 입력노드 : 입력된 (독립변수 or 피처)값을 그대로 출력
 - 은닉노드 : 입력받은 값을 변환하여 출력, 활성화 함수 사용
 - 출력노드
 - 회귀 문제: (보통) 활성화 함수 없음
 - 분류 문제: (보통) 소프트맥스 함수 사용



Neural Network

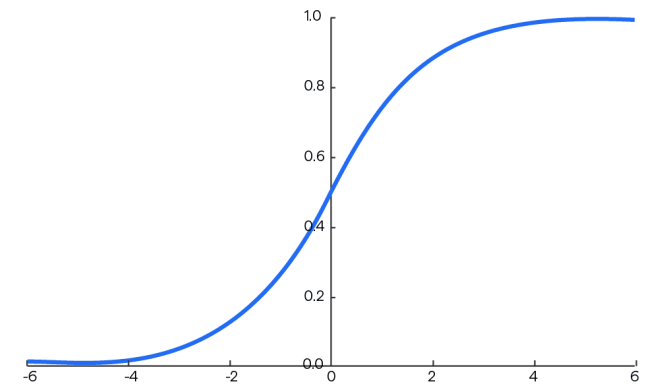
NN 구조

Softmax 함수

The softmax function takes as input a vector \mathbf{z} of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.

- 입력받은 값을 출력으로 0~1사이의 값으로 정규화
- 출력 값들의 총합은 항상 1.
- 분류하고 싶은 클래스수의 수 만큼 출력으로 구성.
- 가장 큰 출력 값을 부여받은 클래스가 확률이 가장 높은 것으로 이용
 - 가장 큰 값을 강조하고 최대값보다 낮은 값을 억제
 - 모델이 예측에 확신을 가져야 하는 다중 클래스 분류 문제에서 유용

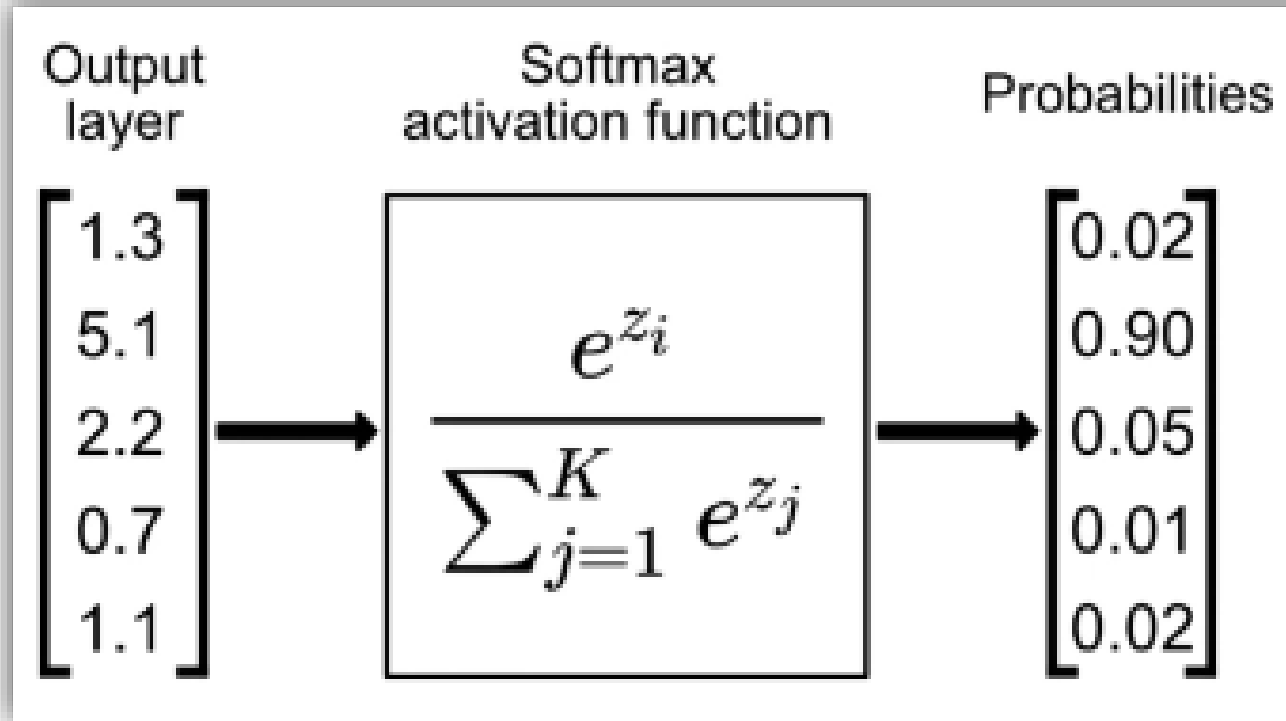
Softmax Function



Neural Network

NN 구조

Softmax 활성화 함수

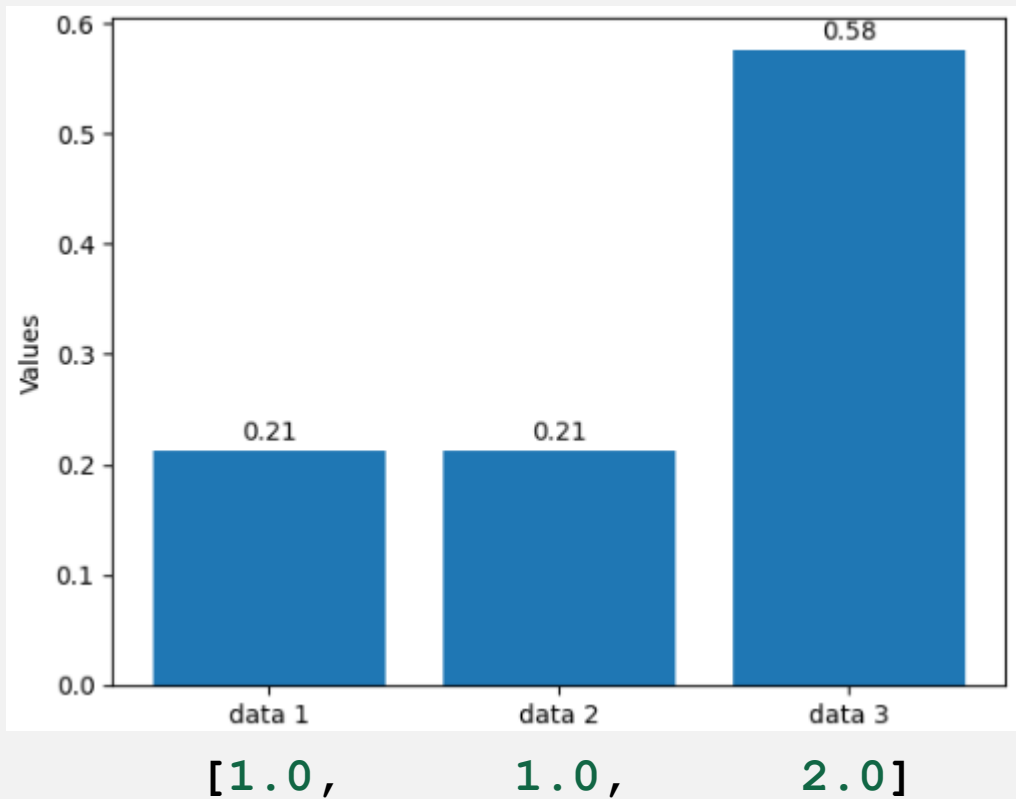


Neural Network

NN 구조

Softmax 활성화 함수

- 실습 - 비용함수
4.03.Softmax.ipynb



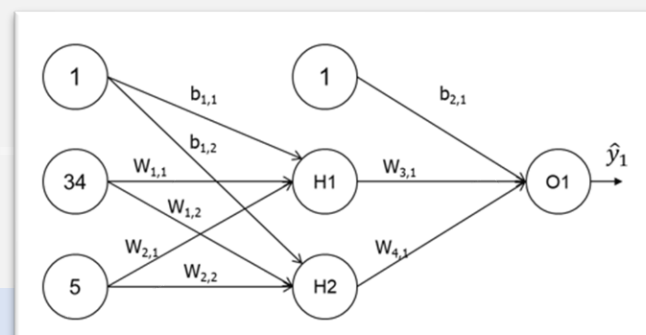
Neural Network

NN 구조

Softmax 활성화 함수

출력값 -> 확률 변환 Process

1. 각 클래스에 대한 로짓(logit) 값 계산
= 신경망 마지막 레이어에서 계산된 값들
2. **Softmax 함수** : 각 로짓의 지수를 계산하여 양수로 변환, 각 출력값이 차지하는 비율(확률) 계산
-> 각 클래스에 속할 확률이 도출 (확률의 합 : 1)



ex:

- 신경망의 출력 : 세 개의 클래스에 대한 로짓 값 [2.0, 1.0, 0.1] 반환
- Softmax 함수 : 확률 값으로 변환

1. 각 로짓의 지수 계산:

- $e^{2.0}$ ($e^{2.0}$) = (엑셀 수식 EXP(2.0))
- $e^{1.0}$ ($e^{1.0}$) =
- $e^{0.1}$ ($e^{0.1}$) =

2. 값들의 합:

$$a + b + c = d$$

3. 각 클래스 확률 계산:

1. 첫 번째 클래스의 확률: a/d
2. 번째 클래스의 확률: b/d
3. 세 번째 클래스의 확률: c/d

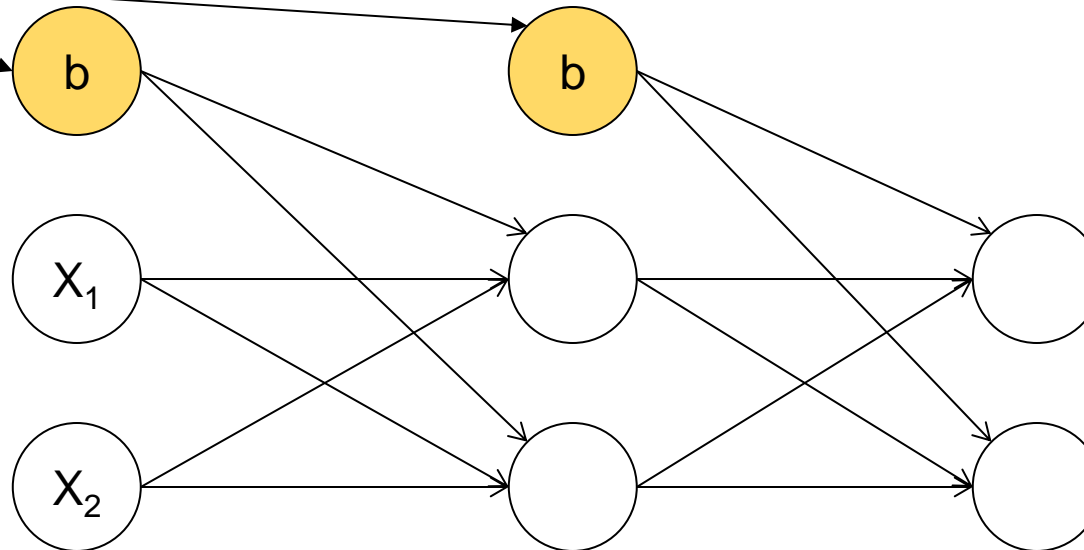
Neural Network

NN 구조

bias node (편향 노드) = 선형 회귀 모델에서의 절편(intercept)

- 입력층과 은닉층에 존재
- 선형 회귀 모델의 intercept(절편) 와 비슷한 역할 (출력값의 기준점(절편) 조절)

bias node



편향을 통해
모델은 입력 데이터가 없거나 0에 가까울 때도 활성화
-> 모델의 학습 능력과 일반화 능력 향상

Neural Network

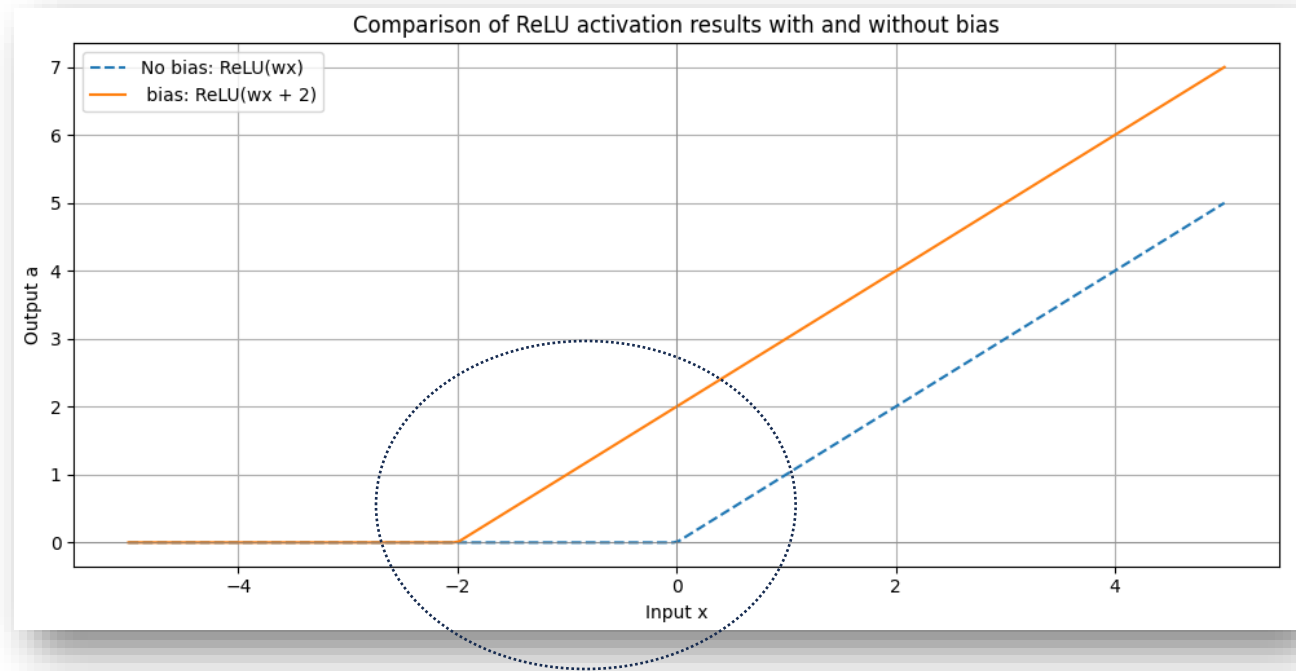
NN 구조

bias node (편향 노드) = 선형 회귀 모델에서의 절편(intercept)

- 입력층과 은닉층에 존재
- 선형 회귀 모델의 intercept(절편) 와 비슷한 역할 (출력값의 기준점(절편) 조절)

• 실습 – bias

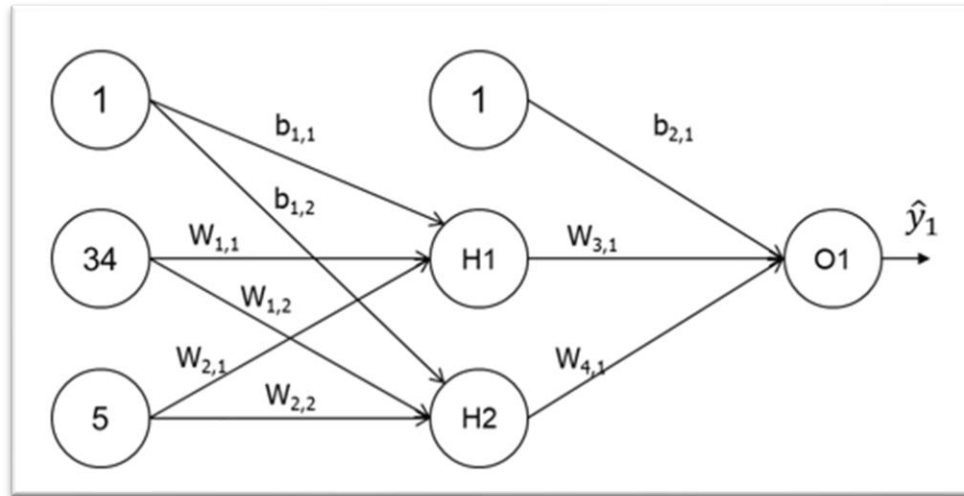
4.03.NN.bias.ipynb



Neural Network

NN 구조

- 출력 노드의 수 - 회귀
 - 출력 노드 1개
 - 출력노드가 출력하는 값 = 종속변수의 예측치(\hat{y}_i)
- ex, 아파트 가격 예측

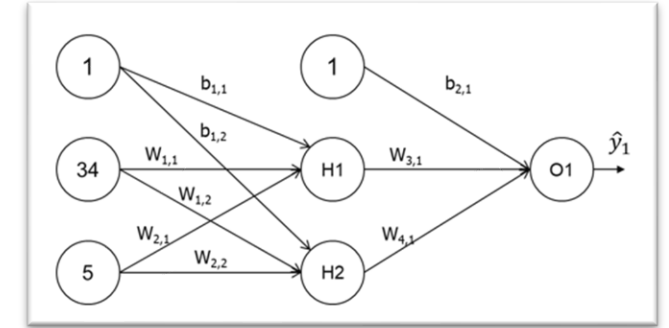


Neural Network

NN 구조

- 출력 노드의 수 - 분류
= 종속변수가 취할 수 있는 값의 수
ex, 감염 여부, then $y_i \in \{0, 1\}$
출력 노드의 수 = 2

- 각 출력 노드가 출력하는 값은?
 - ✓ 종속변수가 각 값을 취할 확률
 - ✓ 첫번째 출력 노드의 출력값 = $P(y_i = 0)$
 - ✓ 두번째 출력 노드의 출력값 = $P(y_i = 1)$



이진 분류 모델의 출력 노드의 수?
1 or 2?

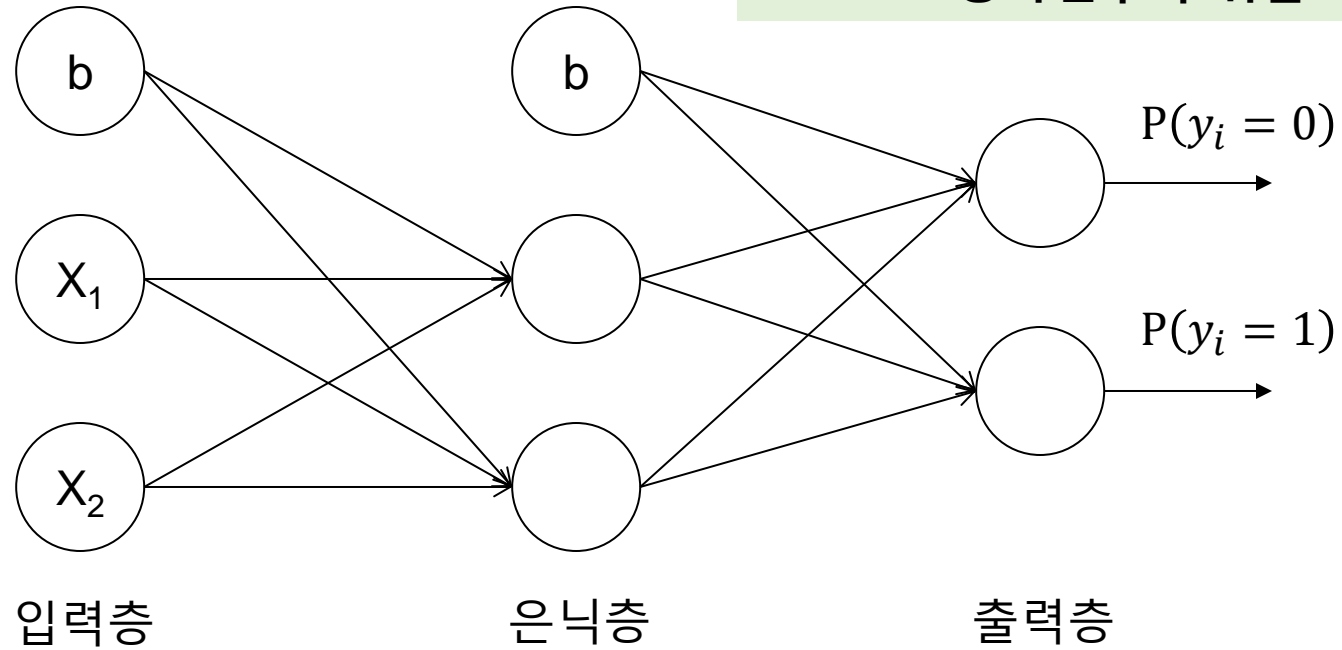
Neural Network

NN 구조

- 분류문제

출력 노드의 수-분류

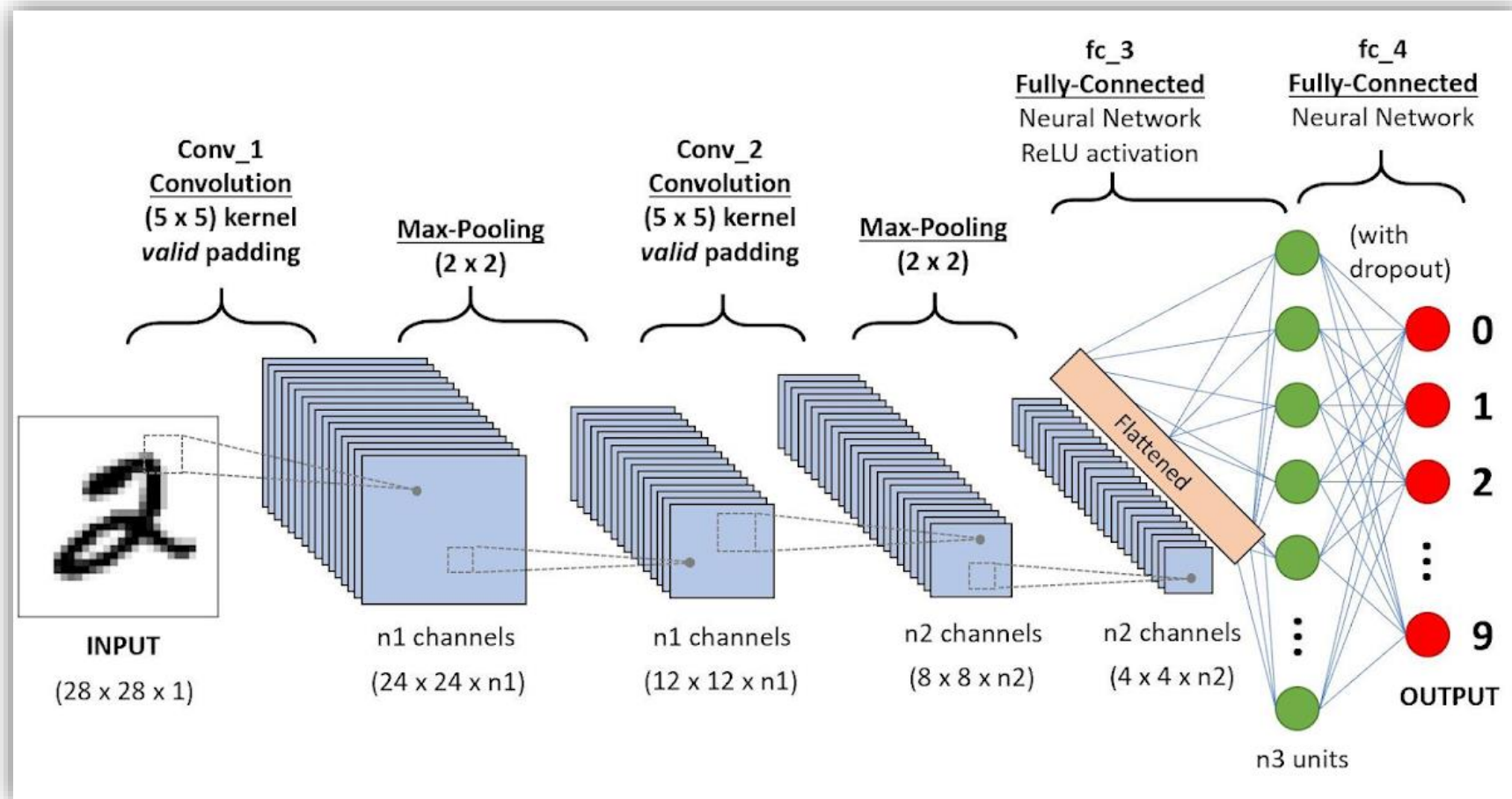
= 종속변수가 취할 수 있는 값의 수



Neural Network

NN 구조

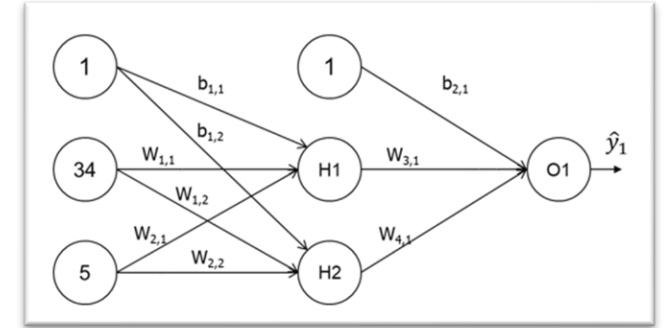
- 분류문제



Neural Network

NN 구조

- 사용하고자 하는 신경망 모델 구조 결정(모델 설계)
 - ✓ 입력노드의 수 = ?
 - ✓ 출력노드의 수 = ?
 - ✓ 은닉층의 수와 은닉노드의 수는 설계자가 결정
ex, 은닉층의 수 = 1, 은닉 노드의 수 = 2 인 모델



Neural Network

NN 구조

```
from keras.models import Sequential
from keras.layers import Dense

# 모델 구축
model = Sequential()
# model.add(Dense(8, input_dim=X_train.shape[1], activation='relu')) # 입력 층
model.add(Dense(8, input_dim=3, activation='relu')) # 입력 층
model.add(Dense(4, activation='relu')) # 은닉 층
model.add(Dense(1, activation='sigmoid')) # 출력 층

# 모델 컴파일
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# 모델 요약 출력
model.summary()
```

• TASK – 모델 구조 스케치

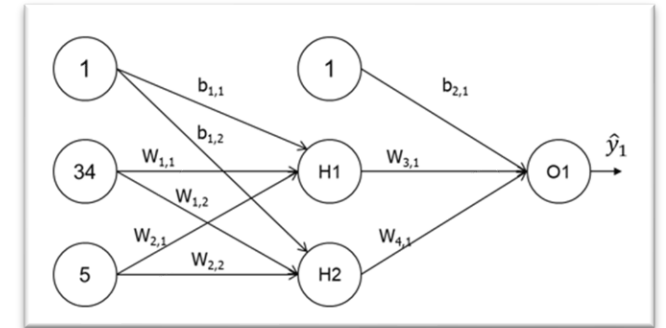
Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 8)	32
dense_16 (Dense)	(None, 4)	36
dense_17 (Dense)	(None, 1)	5

Total params: 73 (292.00 B)
Trainable params: 73 (292.00 B)
Non-trainable params: 0 (0.00 B)

Neural Network

NN 학습

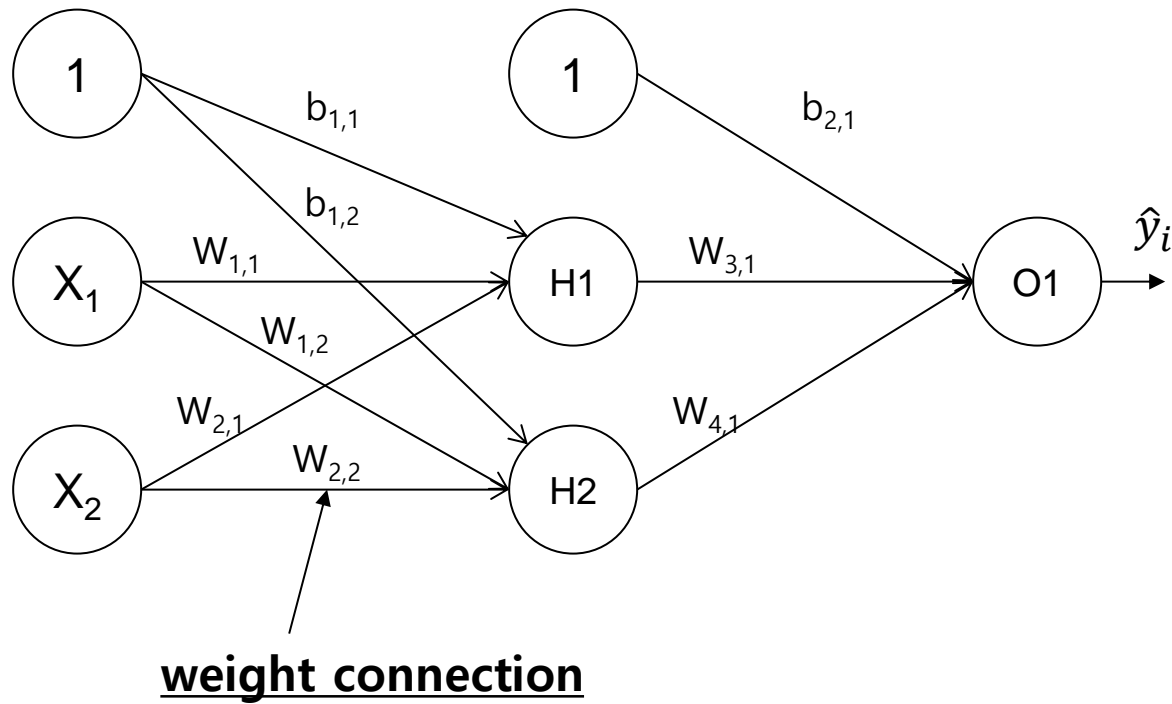
- 회귀문제
 - 종속변수 (y, 연속변수)
ex, 아파트 가격 예측
 - 2개의 독립변수 (X1, X2)
- 학습
 - 비용함수를 최소화하는 모델의 **파라미터값**을 탐색
 - 비용함수 $\rightarrow \frac{1}{N} \sum_i (y_i - \hat{y}_i)^2$
 - 각 관측치의 예측치 (\hat{y}_i)는 (신경망) 모델을 통해 도출
- 사용하고자 하는 신경망 모델
 - ex, 은닉층의 수 = 1, 은닉 노드의 수 = 2 인 모델



Neural Network

NN 학습

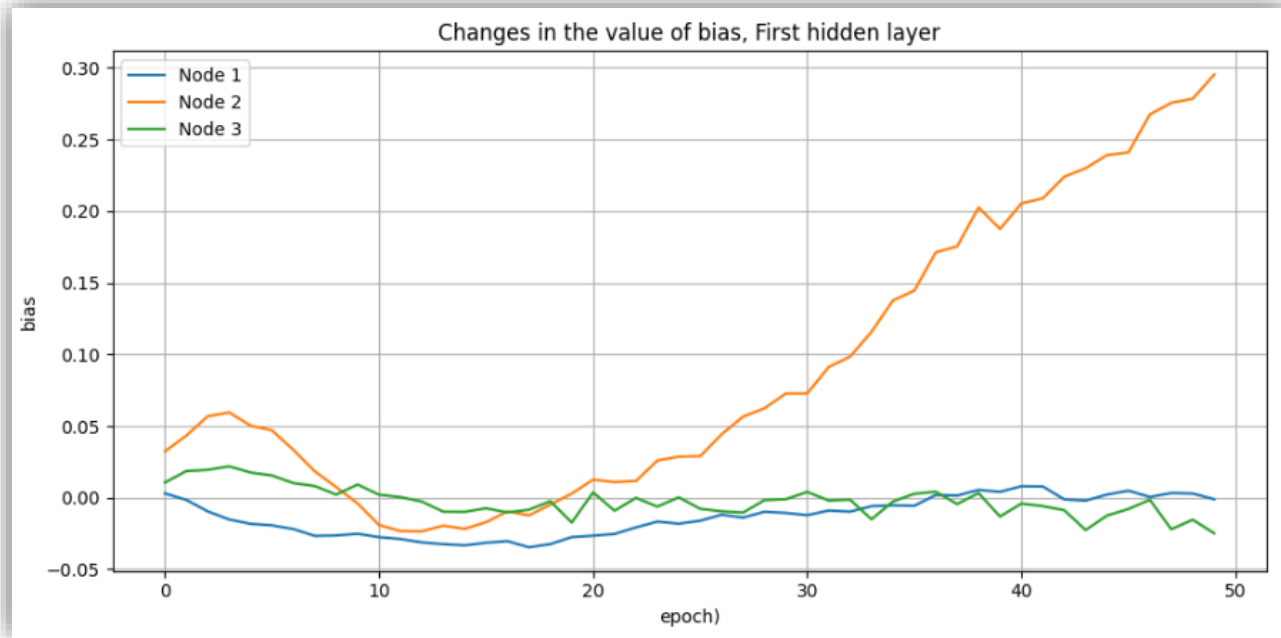
신경망의 파라미터 : 가중치 (weight)와 편향으로 구성



Neural Network

NN 학습

신경망의 파라미터 : 가중치 (weight)와 편향으로 구성



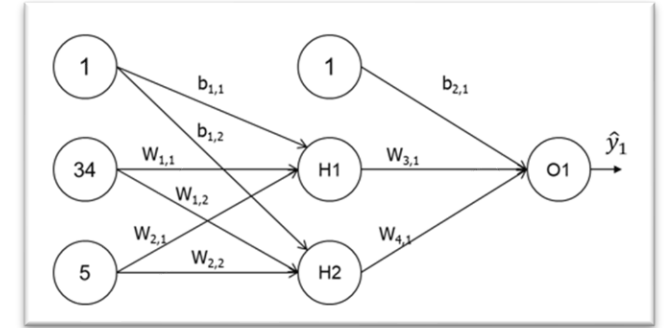
```
biases = self.model.layers[0].get_weights()[1]
self.bias_history.append(biases.copy())
```

Neural Network

NN 학습 - 예제

ex

- 문제: 아파트의 가격 예측
- 사용 독립변수: 아파트의 크기 (평형)와 연식
- training data (10개의 data points)



ID	평수 (X1)	연식 (X2)	가격 (y)
1	34	5	5
2	25	5	2.5
3	30	2	4
4	38	20	3
5	44	12	3.3
6	48	18	4.2
7	52	22	4.6
8	60	19	6
9	34	18	3
10	34	22	2.9

Neural Network

NN 학습 - 예제

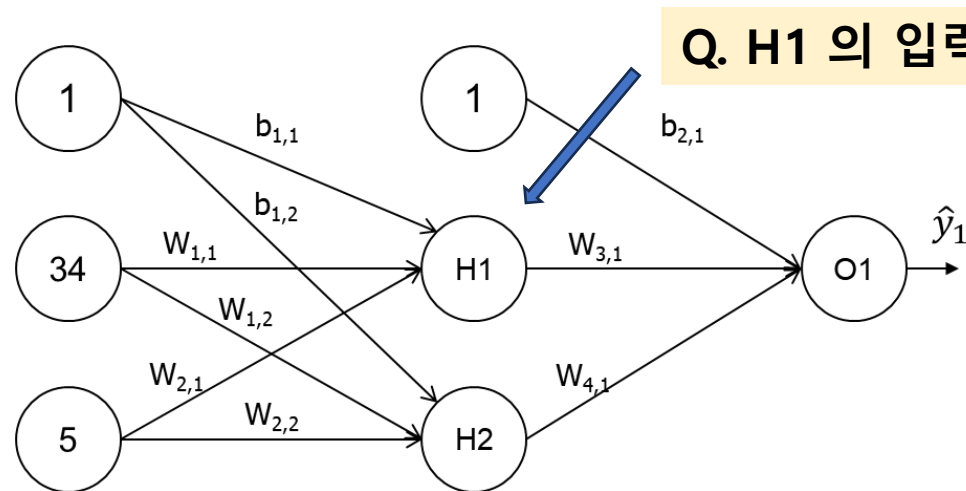
ex

- 문제: 아파트의 가격 예측
- 사용 독립변수: 아파트의 크기 (평형)와 연식
- training data (10개의 data points)

ID	평수 (X1)	연식 (X2)	가격 (y)
1	34	5	5
2	25	5	2.5
3	30	2	4
4	38	20	3
5	44	12	3.3
6	48	18	4.2
7	52	22	4.6
8	60	19	6
9	34	18	3
10	34	22	2.9

- 첫번째 관측치 : $X1 = 34, X2 = 5$

첫번째 입력 노드 : $X1$ 의 값을 입력, 두번째 입력 노드 : $X2$ 의 값을 입력 받음



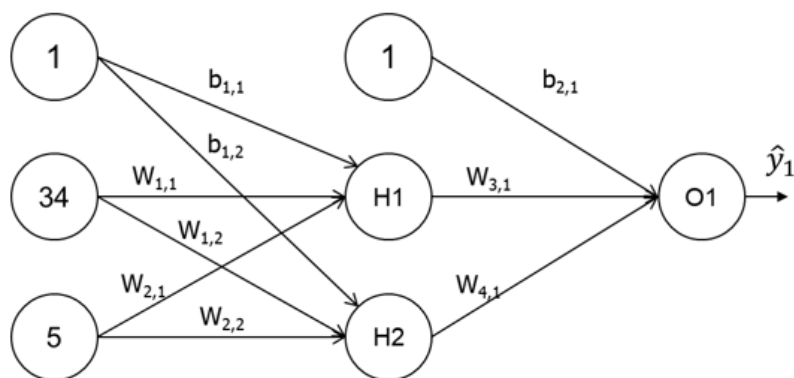
Q. H1의 입력 값은?

각 관측치에 대해 종속변수의 예측치 계산

Neural Network

NN 학습 - 예제

신경망의 가중치(weight) 값의 해석?



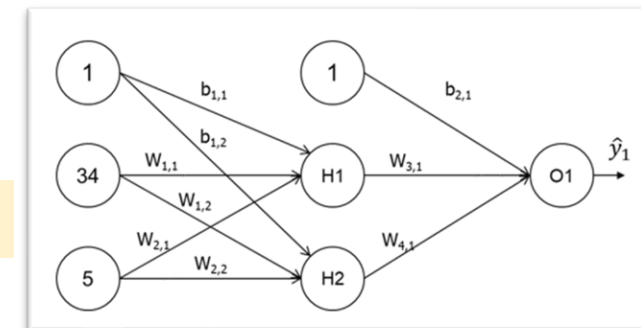
ID	평수 (X1)	연식 (X2)	가격 (y)
1	34	5	5
2	25	5	2.5
3	30	2	4
4	38	20	3
5	44	12	3.3
6	48	18	4.2
7	52	22	4.6
8	60	19	6
9	34	18	3
10	34	22	2.9

첫 번째 아파트의 예측 가격: 4.2520, 실제 가격: 5.0000
모델의 가중치: Parameter containing:
tensor([[-0.0673, 0.5276]], requires_grad=True)
모델의 편향: Parameter containing:
tensor([0.9644], requires_grad=True)

Neural Network

NN 학습 - 입력 데이터

Q. 입력층과 출력층의 구조? - 타이타닉

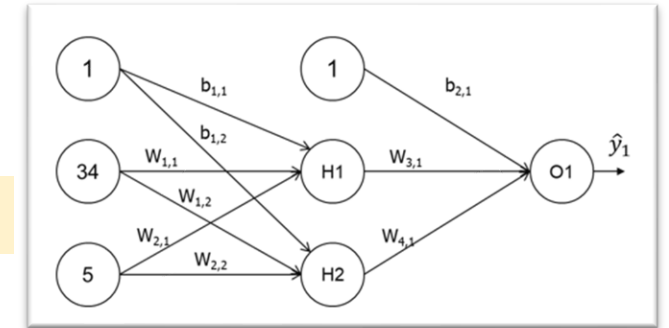


	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton
...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown

Neural Network

NN 학습 - 입력 데이터

Q. 입력층과 출력층의 구조? - 아이리스



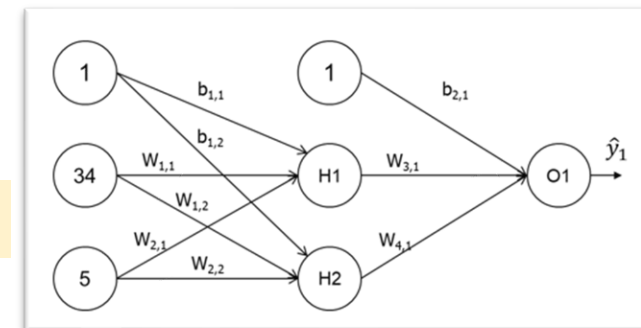
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	4.9	3.0	1.4	0.2	setosa
1	4.7	3.2	1.3	0.2	setosa
2	4.6	3.1	1.5	0.2	setosa
3	6.4	3.2	4.5	1.5	versicolor
4	6.9	3.1	4.9	1.5	versicolor
5	5.5	2.3	4.0	1.3	versicolor
6	7.1	3.0	5.9	2.1	virginica
7	6.3	2.9	5.6	1.8	virginica
8	7.6	3.0	6.6	2.1	virginica



Neural Network

NN 학습 - 입력 데이터

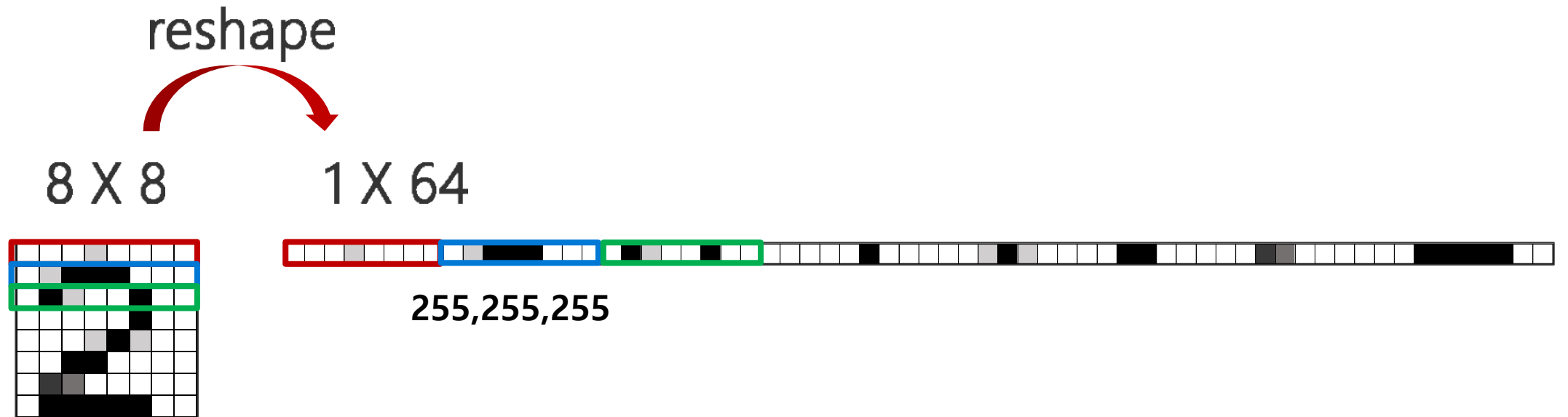
Q. 입력층과 출력층의 구조? - hr



	birthday	entry_year	departmer	marital_status	performan	job_satisfa	working_h	salary	last_year_s	num_companies_worked	attrition
0	1980-07-20	2013	sales	single	high	very high	8.33	9431500	8923739	8	yes
1	1972-11-08	2011	rnd	married	very high	medium	6.93	5170672	4617495		no
2	1984-05-07	2014	rnd	single	high	high	9	9898200	9176045	6	yes
3	1988-10-19	2013	rnd	married	high	high	8.33	5673500	5362476	1	no
4	1994-07-11	2015	rnd	married	high	medium	7.2	3484080	3284389	9	no
5	1989-11-24	2013	rnd	single	high	very high	6.93	6592560	6185550	0	no
6	1962-05-04	2009	rnd	married	very high	low	8.33	8281000	7526129	4	no
7	1991-08-24	2020	rnd		very high	high	7.8	6281496	5637167	1	no
8	1983-03-22	2011	rnd	single	very high	high	8.67	4741880	4279675		no
9	1985-05-02	2004	rnd	married	high	high	8.33	9180250	8591717	6	no
10	1986-08-12	2015	rnd	married	high	medium	8.33	8266500	7761243	0	no
11	1992-06-02	2011	rnd	single	high	high	8.33	4959500	4657682	0	no

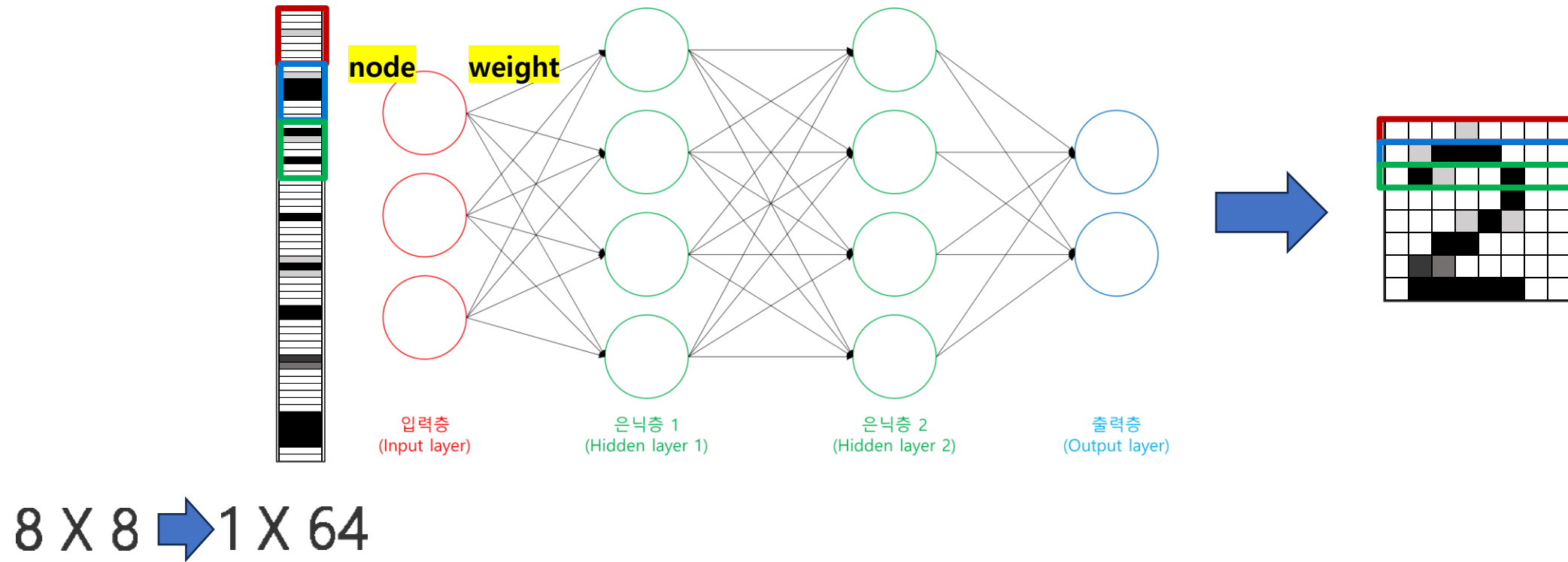
Neural Network

NN 학습 - 입력 데이터



Neural Network

NN 학습 - 입력 데이터



Neural Network

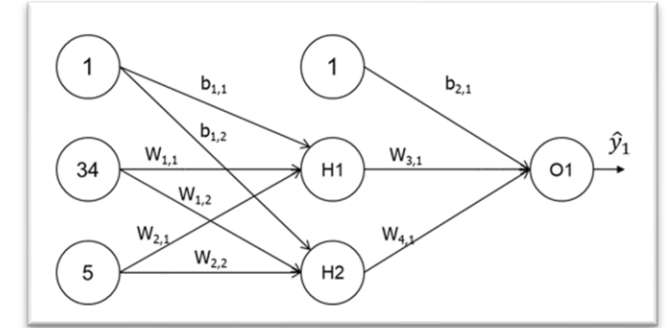
NN 학습 – 입력 데이터

Artificial Neural Network Modeling

If we can **train a model to map X to Y** based on a labelled dataset then it can be used to predict

Neural Network

NN 학습 - Process



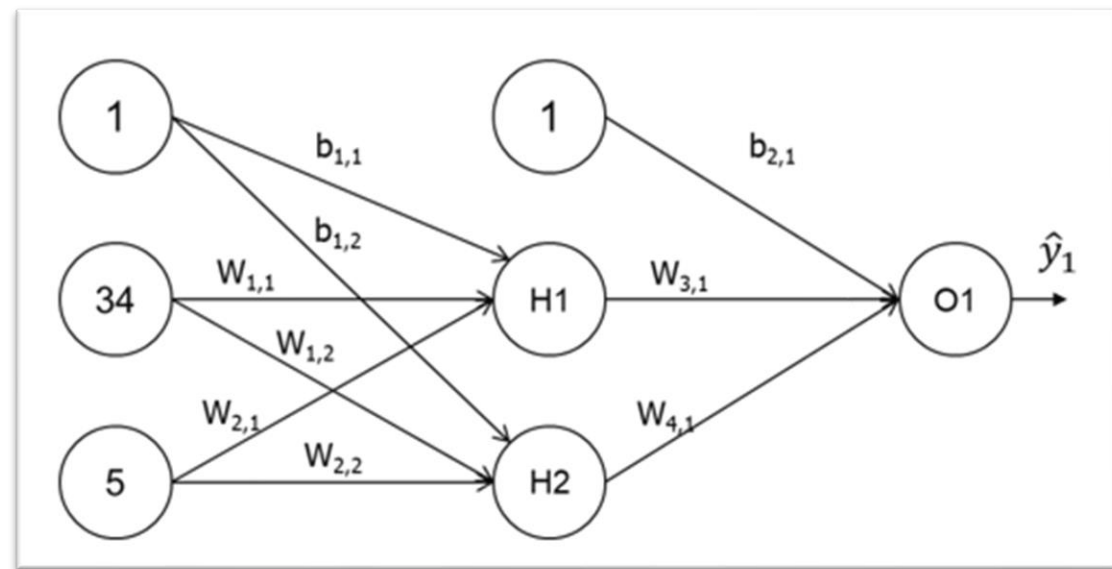
- ① 데이터 세팅 (with Lable)
- ② 데이터를 학습 데이터와 평가 데이터로 분리
- ③ 신경망 (딥러닝 모델)을 통해 독립변수와 종속변수의 관계 파악
- ④ 비용함수를 최소화하는 파라미터의 (최적)값을 탐색
- ⑤ 모델 성능 평가, 훈련된 모델을 사용, 새로운 데이터에 대해 종속변수 값 예측



Neural Network

NN 학습 - Process

- 각 은닉노드 (즉, H1과 H2)에 입력되는 값은?
 - 입력 노드가 출력하는 값과 각 가중치의 곱, 그리고 이들의 합 + 편향
 - H1 입력 값 (z_1)
 - $z_1 = b_{1,1} + w_{1,1} \cdot 34 + w_{2,1} \cdot 5$
 - H2 입력 값?
 - $z_2 = ?$
- 은닉 노드의 출력값
 - 입력된 값을 특정한 형태로 변환
=> **활성화 함수 (activation function)** 사용

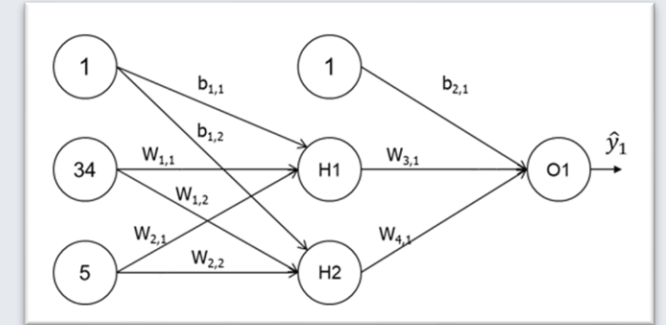


Neural Network

NN 학습 - Process

활성화 함수(Activation Function)

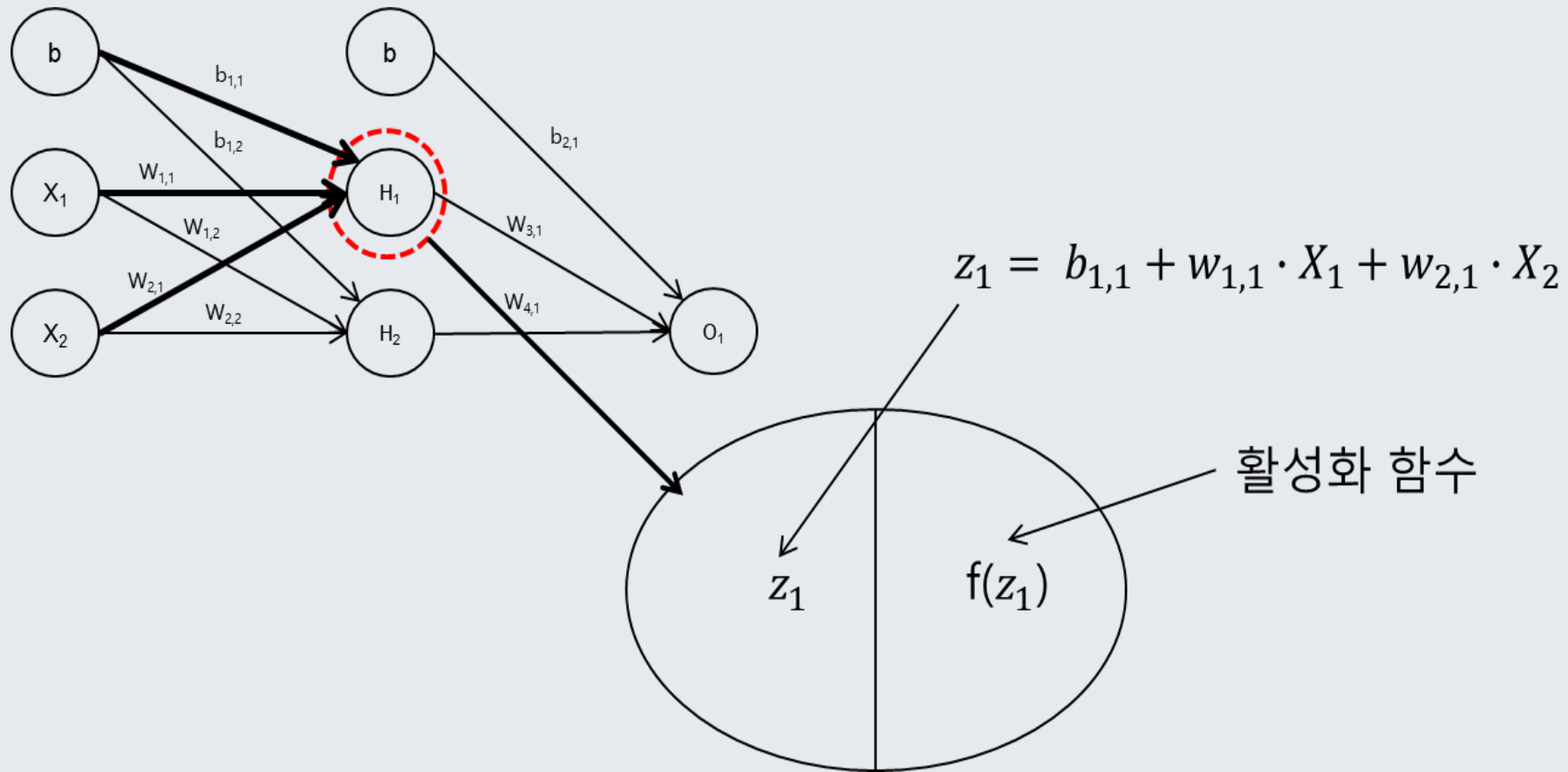
- (통상) 은닉노드와 출력 노드에 존재,
해당 노드에 입력된 값을 변환하여 출력
- 출력되는 값은 해당 노드가 정답을 맞히는데 기여하는 정도를 반영
- $f(z)$
 - z : 해당 노드에 입력되는 값
 - z 를 입력 받아 $f(z)$ 출력
- f : 비선형 함수 => 독립변수와 종속변수 간에 존재할 수 있는 비선형 관계 파악
 - 선형함수를 여러개 사용? -> 선형 관계
- H1과 H2 노드
 - H1: z_1 을 입력받고 $f(z_1)$ 출력
 - H2: z_2 를 입력받고 $f(z_2)$ 출력



Neural Network

NN 학습 - Process

활성화 함수(Activation Function)

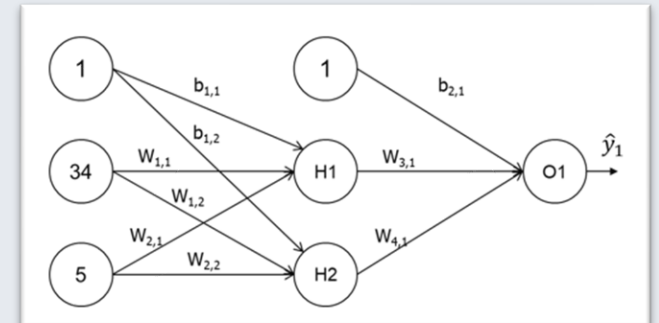


Neural Network

NN 학습 - Process

활성화 함수(Activation Function)

- 출력노드 (O1)에 입력되는 값
 - $z_3 = b_{2,1} + w_{3,1} \cdot f(z_1) + b_{2,1} + w_{4,1} \cdot f(z_2)$
- 출력노드의 활성화 함수
 - 수행하는 Task 에 따라
 - 회귀: 없음 (또는 항등함수 즉, $z_3 = f(z_3)$)
 - 분류
 - (보통) 소프트맥스 함수 사용
 - 확률값 리턴
- 아파트 가격 예측 (회귀)
 - $\hat{y}_i = b_{2,1} + w_{3,1} \cdot f(z_1) + w_{4,1} \cdot f(z_2) = z_3$



활성화 함수

deeplearning.ActivationFunction.share.pdf

Neural Network

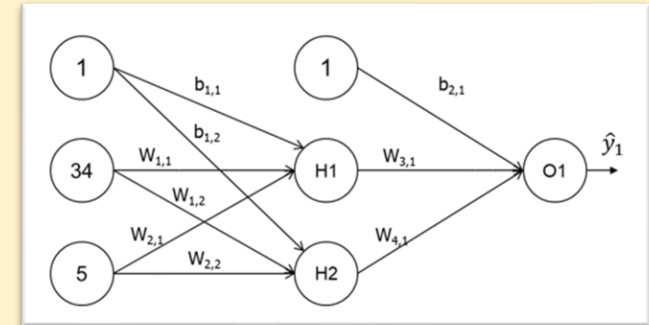
NN 학습 - 비용함수

비용 함수 (cost function)

- 신경망의 예측이 실제 데이터와 얼마나 잘 일치하는지를 측정하는 데 사용
- 모델의 예측이 실제 값에 가까워지도록 모델의 파라미터(가중치와 편향)를 조정
- 위 과정을 통해 신경망이 주어진 데이터에 대해 최적의 예측을 수행하도록 학습

비용 함수의 역할

- **모델 평가:** 비용 함수는 모델의 예측이 실제 값과의 차이를 수치적으로 표현
(비용 함수 값이 낮을수록 모델의 예측이 실제 데이터에 더 잘 부합함을 의미)
- **모델 최적화:** 학습 과정은 비용 함수의 값을 최소화하기 위해 가중치와 편향을 조정하는 과정



Neural Network

NN 학습 - 비용함수

- 분류 문제의 비용함수: 교차 엔트로피

- $E = -\left[\sum_{i=1}^N y_i \ln p(y_i = 1) + (1 - y_i) \ln p(y_i = 0)\right]$
- $y_i \rightarrow$ 각 관측치의 실제 종속변수 값
- $y_i \in \{0,1\}$
- $p(y_i = 1), p(y_i = 0) \rightarrow$ 모델을 통해서 예측되는 값

- 회귀 문제의 비용함수: 평균 제곱 오차 (Mean Squared Error, MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

N : 데이터 포인트의 수,

y_i : i번째 실제 값,

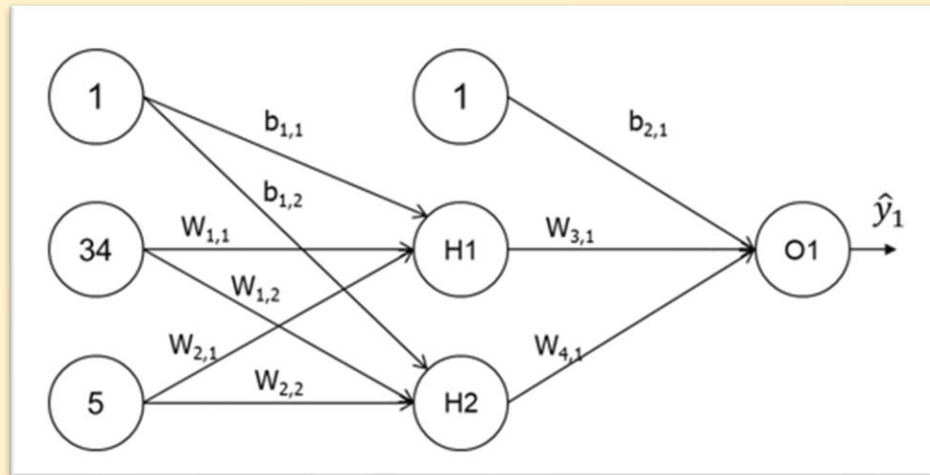
\hat{y}_i : i번째 예측 값

```
1. 11억 . 12억 => |1억|
2. 10억   9억  => |-1억|
3. ....
-----
sum (a, b, c) / 3
```

Neural Network

NN 학습 - 비용함수

- 회귀 : MSE 등
 - $\frac{1}{N} \sum_i (y_i - \hat{y}_i)^2$
 - ex, 아파트 가격 예측
 - y_i : i 번째 관측치의 실제 y값
 - \hat{y}_i : i 번째 관측치에 대한 예측치
 - $\hat{y}_i = b_{2,1} + w_{3,1} \cdot f(z_{i,1}) + b_{2,1} + w_{4,1} \cdot f(z_{i,2})$



Neural Network

NN 학습 - 비용함수

비용함수의 종류 - 회귀

1. 평균 제곱 오차 (Mean Squared Error, MSE)

- 예측값과 실제값 사이의 차이의 제곱의 평균

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- y_i : 실제값, \hat{y}_i 예측값, N 은 데이터 포인트의 총 개수
- MSE는 가장 일반적으로 사용, 큰 오차에 더 많은 가중치를 둠

2. 평균 절대 오차 (Mean Absolute Error, MAE)

- 예측값과 실제값 사이의 차이의 절대값의 평균

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

- 이상치(outlier)에 덜 민감

Neural Network

NN 학습 - 비용함수

비용함수의 종류 - 회귀

3. 허브(Huber) 손실

- MSE와 MAE의 조합. 오차가 작을 때는 MSE, 큰 오차에 대해서는 MAE
- 이상치에 대해 더 강인

- $L_{\delta}(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$ for $|y - \hat{y}| \leq \delta$
- $L_{\delta}(y, \hat{y}) = \delta|y - \hat{y}| - \frac{1}{2}\delta^2$ otherwise

δ : 임계값 (MSE와 MAE 사이의 전환점)

4. 로그 코사인 유사도 (Log-Cosh Loss)

- 오차의 하이퍼볼릭 코사인의 로그

$$\text{Log-Cosh} = \frac{1}{N} \sum_{i=1}^N \log(\cosh(\hat{y}_i - y_i))$$

- 이상치에 매우 강인, 최적화에 장점

Neural Network

NN 학습 - 비용함수

- 분류

- 예: 폐질환 여부
 - 폐질환 음성 $y = 0$, 양성 $y = 1$
- 데이터 (예시)

ID	나이 (X1)	흡연여부 (X2)	도시거주여부 (X3)	폐암 여부 (y)
1	34	1	0	1
2	60	0	1	1
3	55	0	0	0

Neural Network

NN 학습 - 비용함수

- 분류

- 비용함수: 교차 엔트로피

- $E = -\left[\sum_{i=1}^N y_i \ln p(y_i = 1) + (1 - y_i) \ln p(y_i = 0)\right]$
 - $y_i \rightarrow$ 각 관측치의 실제 종속변수 값
 - $p(y_i = 1)$ & $p(y_i = 0) \rightarrow$ 모델을 통해서 예측되는 값

비용함수의 종류 - 분류

모델이 얼마나 잘 분류를 수행하는지 측정
주로 클래스 정답(레이블)과 예측 확률 사이의 오차를 계산

1.교차 엔트로피 손실 (Cross-Entropy Loss):

- 이진 분류 문제에서 사용
(다중 클래스 분류 문제에서는 범주형 교차 엔트로피 손실(Categorical Cross-Entropy Loss) 또는 소프트맥스 손실(Softmax Loss) 사용)
- 모델의 예측이 실제 레이블로부터 얼마나 벗어났는지를 나타냄
- 예측 확률이 실제 정답에 가까울수록 손실은 낮아짐.

2.힌지 손실 (Hinge Loss):

- 주로 서포트 벡터 머신(Support Vector Machine, SVM)에서 사용.
- 주로 이진 분류 문제에 사용, 마진 오류를 최소화하는 것이 목표

- 실습 – 비용함수

4.03.CostFunction.ipynb

```
criterion = nn.CrossEntropyLoss()  
loss='sparse_categorical_crossentropy'
```


- TASK – NN 모델 하이퍼파라미터 설정

4.03.NN.ModelTest.ipynb

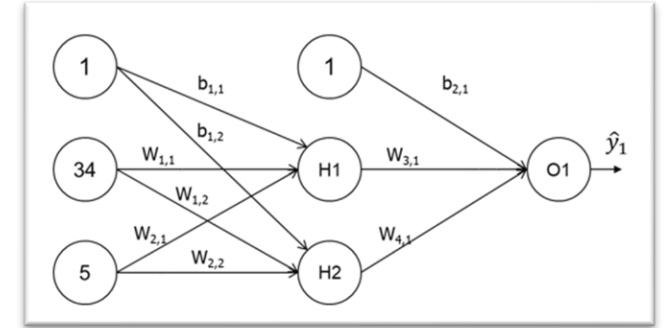
다양한 모델 하이퍼파라미터를 설정, 변경 하여 모델 성능에 미치는 영향 확인

- 모델 설정 및 성능 평가 내용 정리(ex 엑셀 표)

Neural Network

NN -요약

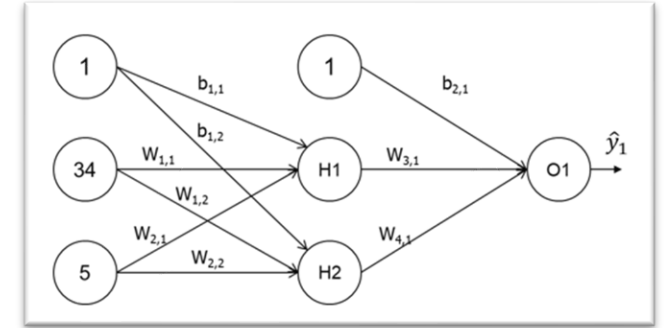
- 딥러닝 : 신경망 기반 모델
- 신경망의 구조
 - 입력층, 은닉층, 출력층
 - **은닉층** : 다른 기계학습 알고리즘과의 가장 큰 차이
- 은닉 노드
 - **활성화 함수**
 - 같은 층에 있는 노드들은 같은 활성화함수 사용
 - 활성화함수는 (보통) **비선형 함수**를 사용
 - 각 은닉노드가 종속변수를 예측하는데 얼마만큼의 기여를 하는지를 반영



Neural Network

NN -요약

- 가중치 = 파라미터
- 비용함수
 - 문제의 종류에 따라 구분
 - 실제의 종속변수 값과 모델을 통한 예측치로 구성
 - 회귀문제
 - 종속변수 값의 예측치 출력
 - 분류문제
 - 각 노드에서 출력되는 값 \rightarrow 종속변수가 특정한 값을 갖을 확률 (출력노드에 입력된 값이 확률 값으로 변환되어 출력)



THANK YOU