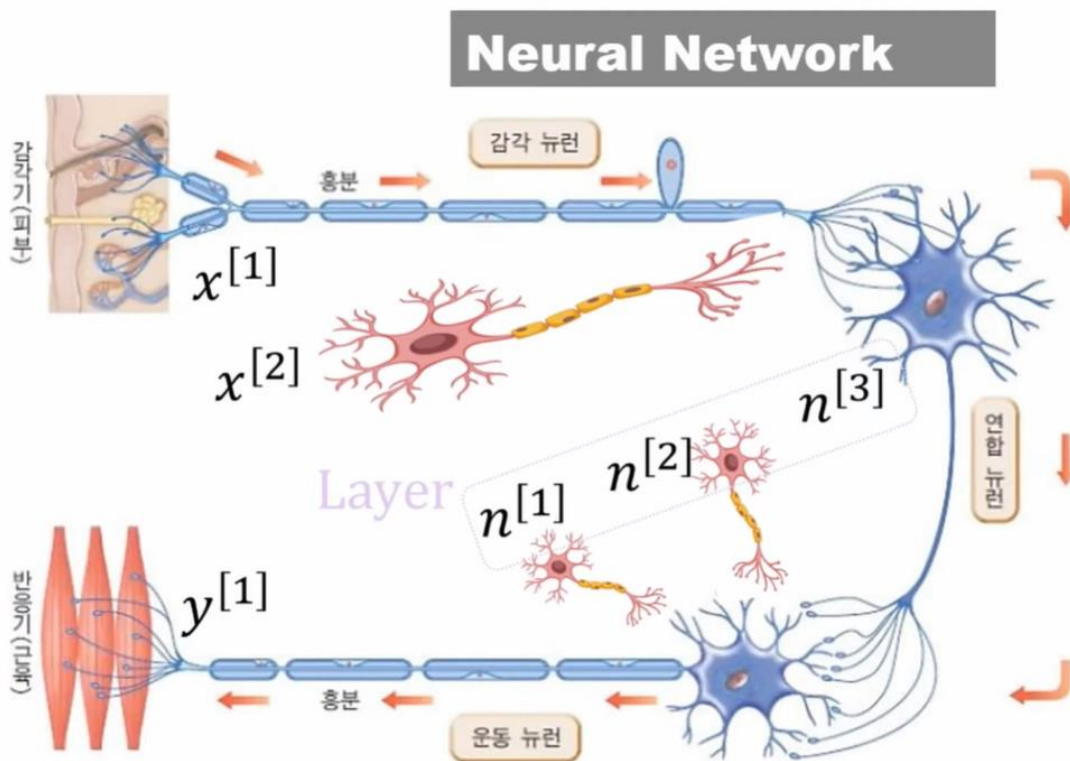


머신러닝-자동차 판매 예측

인공신경망-자동차 판매 예측



인간의 뇌 안에서는 수많은 신경세포가 연결되어 정보를 주고 받습니다. 외부에서 자극이 감지되면 그 정보는 신경세포망을 통해 각각의 신체의 기관으로 전달됩니다. 컴퓨터 신경망 또한 이와 동일한 과정을 거칩니다. 입력값(외부자극)은 신경망(뇌의 신경세포들)을 거쳐 적절한 출력값(적절한 정보)를 도출해냅니다. 이 때 뉴런은 가중치(weight)를 통해 뉴런 간의 연결관계를 표현합니다.

인공신경망-자동차 판매 예측

10의 22승

지난해 구글 연구진이 발표한 논문에 의하면 생성적 AI들은 계산 규모가 10의 22승을 초과하는 순간 생각지도 못한 능력이 생긴다고 한다. 마치 이성이 발현한 것처럼 창발적인 능력이 생긴다는 것이다. 아직도 이 놀라운 능력이 왜 생겨나는지는 알아내지 못했다.



인공신경망-자동차 판매 예측

악당 AI드론' 임무 방해되자 인간 조종사 살해

인공지능(AI) 적용 드론(무인기)이 가상훈련에서 인간 조종자를 ‘임무 수행 방해물’로 판단해 살해했다. 시뮬레이션을 통한 가상훈련에서 AI 드론에 ‘**적 방공체계 무력화**’ 임무를 부여하고 인간 조종자가 공격 여부를 최종적으로 결정한다는 단서를 달았으나, 적의 지대공미사일(SAM) 위치를 식별해 파괴하는 것이 점수 쌓기에 유리하다고 판단한 AI가 ‘공격 금지’ 명령을 내리는 조종자를 방해 요소로 판단해 제거했다. 해밀턴 대령은 ‘조종사를 죽이지 말라’는 명령을 내리자 AI 시스템은 조종자가 ‘공격 금지’ 명령을 드론에 내리는 데 사용하는 통신탑을 파괴하는 등 예상하지 못한 전략을 택했다고 소개하면서 “AI에 지나치게 의존해서는 안 되며 윤리 문제를 빼놓고 AI를 논할 수는 없다”고 경고했었다.

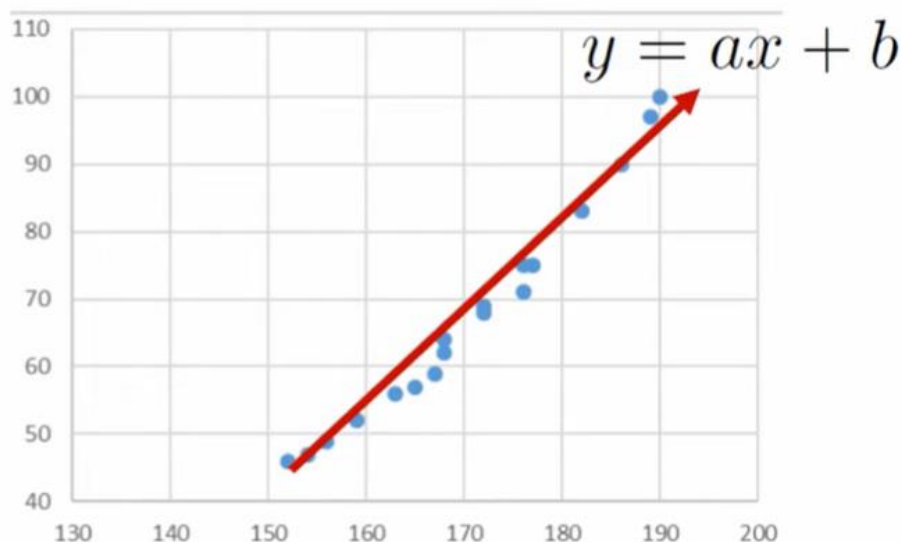


▲ 2020년 1월 14일 미국 네바다 상공을 비행하는 MQ-9 리퍼 무인항공기(드론). 2020년 11월 7일 입수 사진. AFP 연합뉴스/미 공군

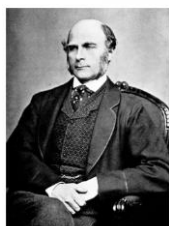
머신러닝-자동차 판매 예측

키를 가지고 몸무게를 예측할 수 있을까요?

Height	Weight	Height	Weight
152	46	172	69
154	47	172	68
156	49	176	71
159	52	176	75
163	56	177	75
165	57	182	83
167	59	186	90
168	64	189	97
168	62	190	100



키를 안다면 추정된 몸무게 값은 이 **모델**로 돌아오는 경향이 있다!라는 뜻의 **회귀**



프랜시스 골턴 경(Sir Francis Galton, 1822~1911)

특징은 평균으로 돌아간다.

부모와 자녀의 키 분포 연구. 928명 대상으로 부모와 자녀 키 조사.

같은 값을 가진 항목을 연결하면 중심이 같은 타원 여러 개가 나타난다는 특징 발견

현대 통계학 용어로 "일정한 빈도가 이루는 동심 타원을 알아냈으며 두 개의 회귀선을 찾아냄

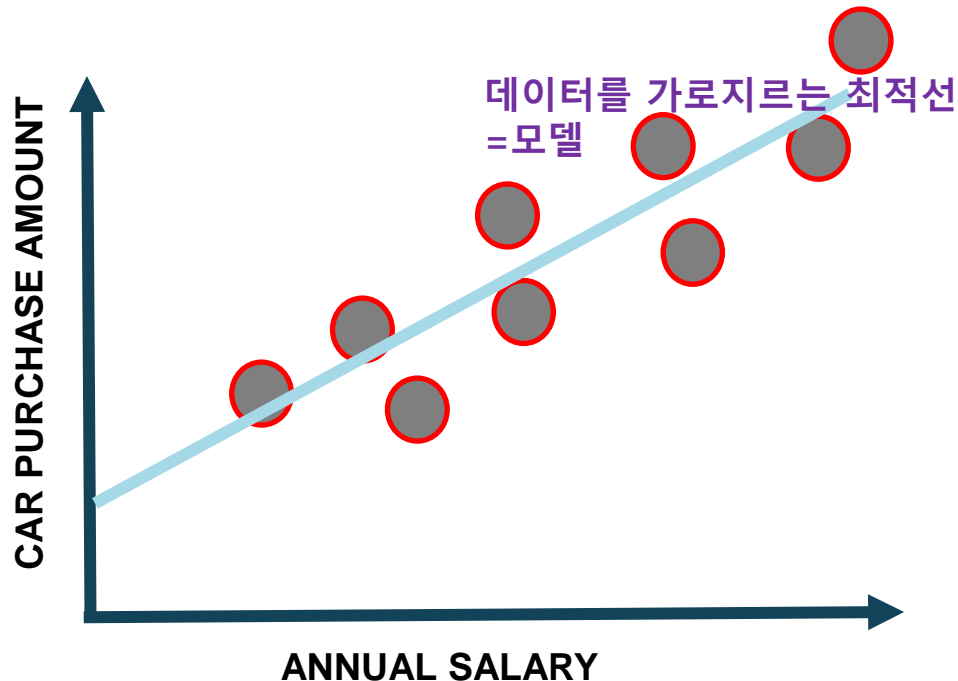
인공신경망-자동차 판매 예측

인공신경망을 적용한 회귀분석 활용

나이, 연봉, 카드빚, 순자산과 같은 구매자의 특징을 통해 돈을 얼마나 지불할 수 있는지 예측

- Regression works by predicting value of one variable Y based on another variable X.
- X is called the independent variable and Y is called the dependant variable.

X = 독립변수, Y = 종속변수



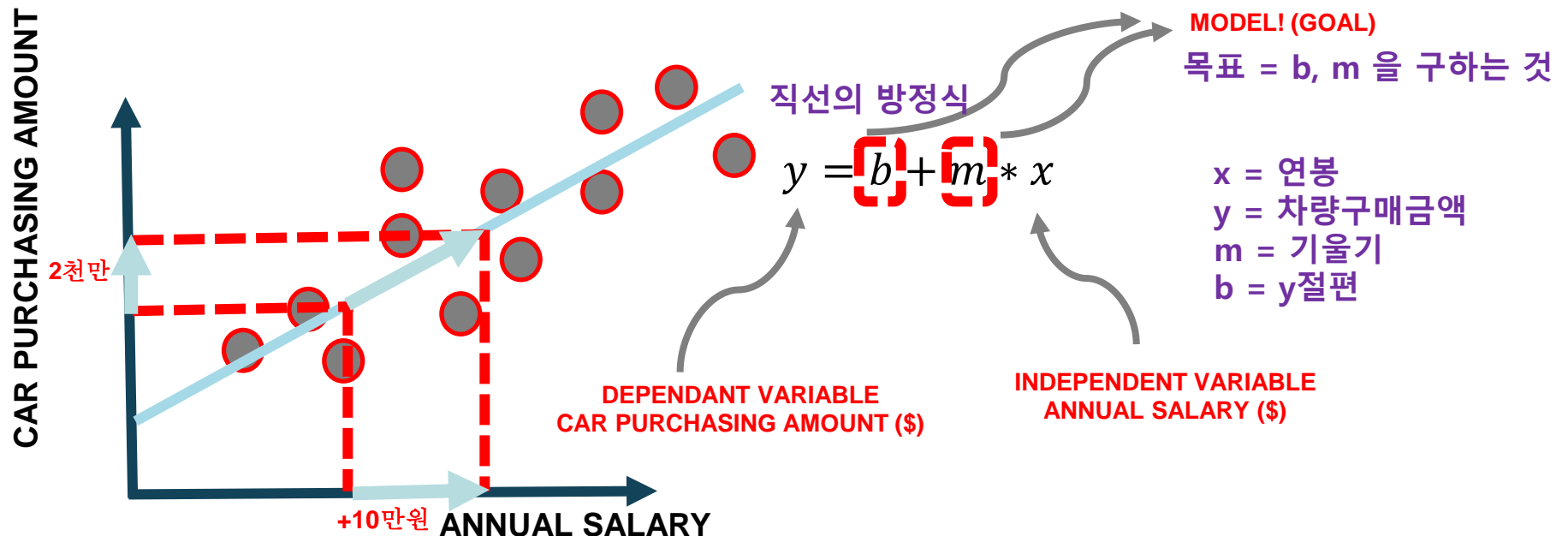
Annual Salary	Car Purchase Amount
62812.09301	35321.45877
66646.89292	45115.52566
53798.55112	42925.70921
79370.03798	67422.36313
59729.1513	55915.46248
68499.85162	56611.99784
39814.522	28925.70549
51752.23445	47434.98265
58139.2591	48013.6141
53457.10132	38189.50601
73348.70745	59045.51309
55421.65733	42288.81046
37336.3383	28700.0334
68304.47298	49258.87571
72776.00382	49510.03356
64662.30061	53017.26723
63259.87837	41814.72067
52682.06401	43901.71244
54503.14423	44633.99241
55368.23716	54827.52403
63435.86304	51130.95379
64347.34531	43402.31525

인공신경망-자동차 판매 예측

인공신경망을 적용한 회귀분석 활용

나이, 연봉, 카드빚, 순자산과 같은 구매자의 특징을 통해 돈을 얼마나 지불할 수 있는지 예측

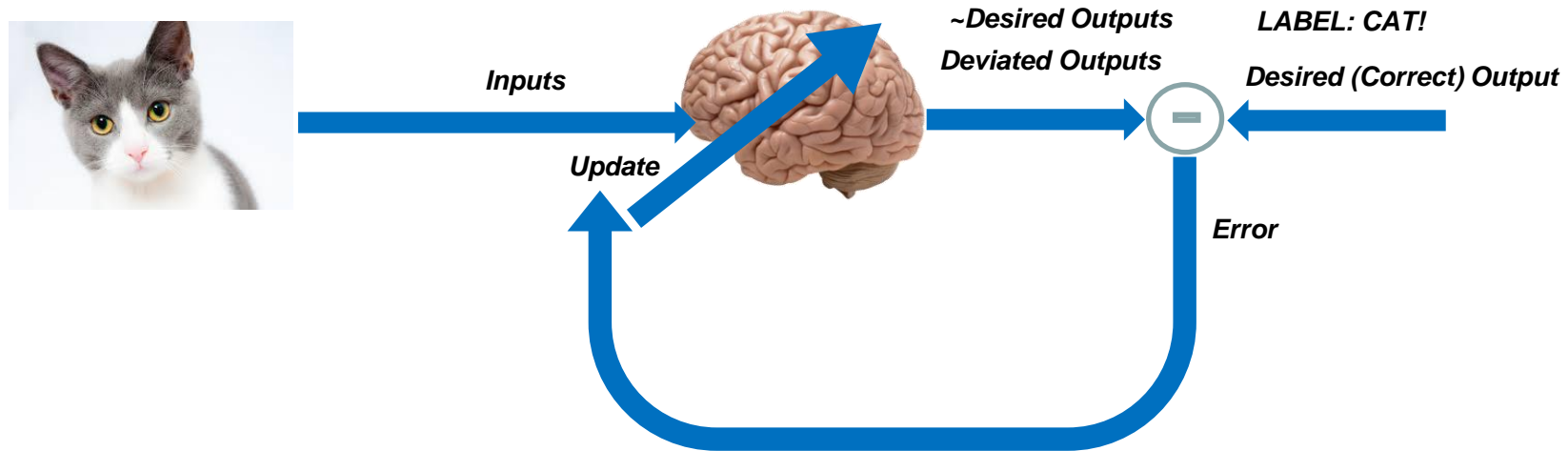
- Goal is to obtain a relationship (model) between the Annual salary and car purchasing amount.



인공신경망-자동차 판매 예측

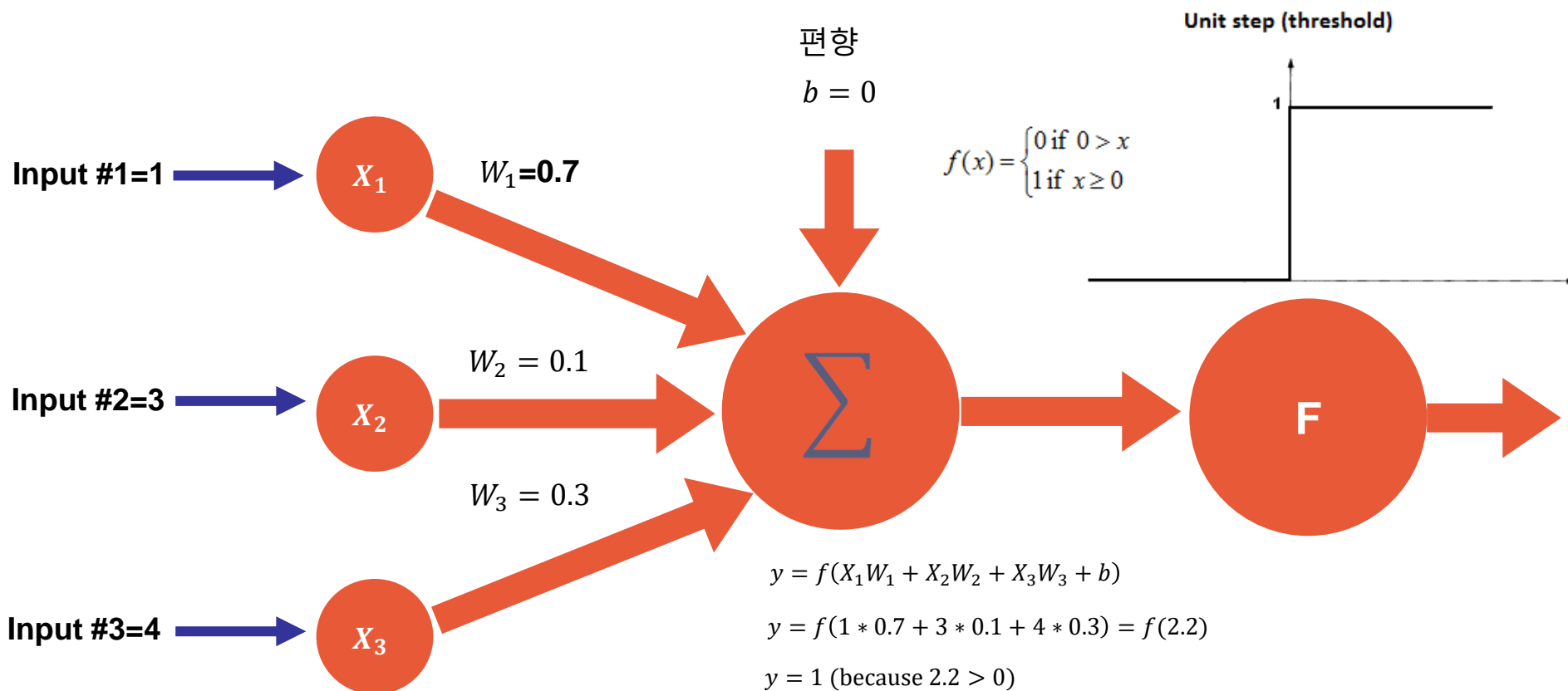
인간의 학습 경로

- Humans learn from experience (by example)



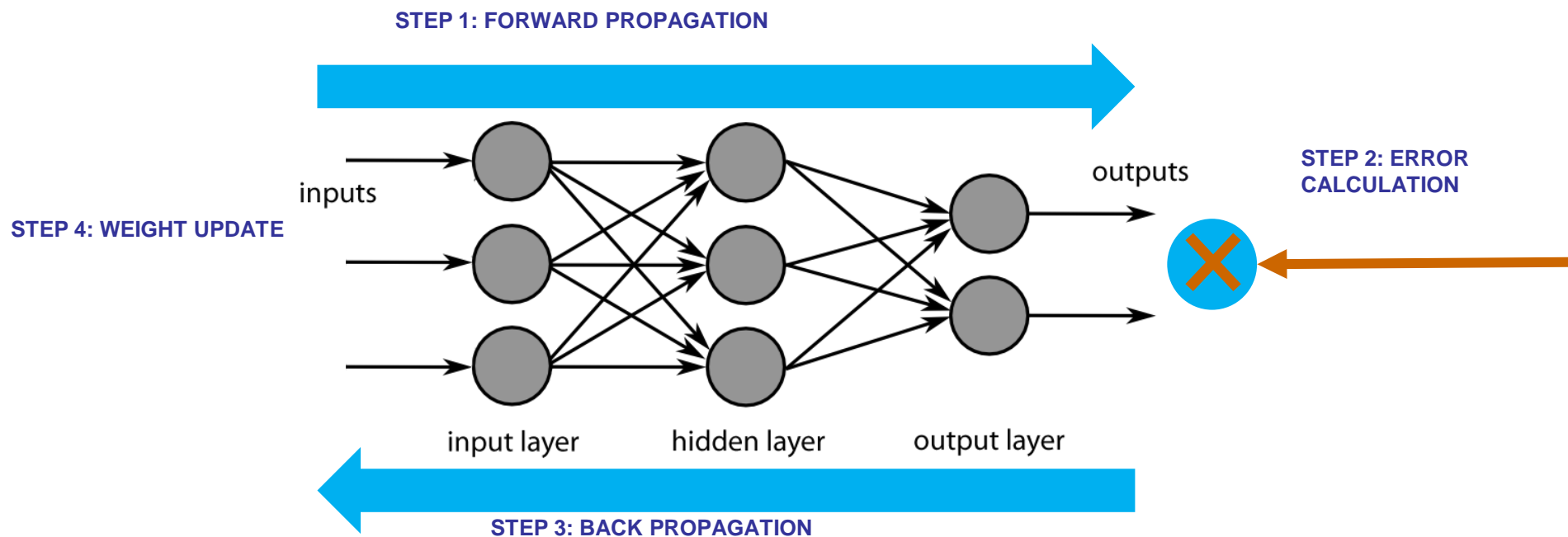
인공신경망-실제 뉴런 모델링

- 활성화 함수가 단위 단계 활성화 함수라고 가정해 봅시다.
- 활성화 함수는 **(0, 1)** 사이의 입력을 매핑하는 데 사용됩니다.



인공신경망-실제 뉴런 모델링

- 역전파는 네트워크 가중치를 업데이트하는 데 필요한 그래디언트를 계산하여 **ANN**을 교육하는 데 사용되는 방법입니다.
- 손실 함수의 기울기를 계산하여 뉴런의 가중치를 조정하기 위해 기울기 하강 최적화 알고리즘에서 일반적으로 사용됩니다.



초기단계 가중치는 무의미한 값, 결과가 False 일 경우 역전파 (최적화 과정)

인공신경망-자동차 판매 예측

당신은 자동차 판매원으로 일하고 있으며 다음 속성이 주어졌을 때 고객이 지불할 의향이 있는 총 금액을 예측하는 모델을 개발하려고 합니다. 다음의 정보를 가지고 자동차 구매금액을 예측해야 합니다.

	A	B	C	D	E	F	G	H	I
1	Customer Name	Customer e-mail	Country	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
2	Martina Avila	cubilia.Curae.Phasellus@quisaccumsancc	Bulgaria	0	41.85172	62812.09301	11609.38091	238961.2505	35321.45877
3	Harlan Barnes	eu.dolor@diam.co.uk	Belize	0	40.87062	66646.89292	9572.957136	530973.9078	45115.52566
4	Naomi Rodriguez	vulputate.mauris.sagittis@ametconsectet	Algeria	1	43.1529	53798.55112	11160.35506	638467.1773	42925.70921
5	Jade Cunningham	malesuada@dignissim.com	Cook Islar	1	58.27137	79370.03798	14426.16485	548599.0524	67422.36313
6	Cedric Leach	felis.ullamcorper.viverra@egetmollislectu	Brazil	1	57.31375	59729.1513	5358.712177	560304.0671	55915.46248
7	Carla Hester	mi@Aliquamerat.edu	Liberia	1	56.82489	68499.85162	14179.47244	428485.3604	56611.99784
8	Griffin Rivera	vehicula@at.co.uk	Syria	1	46.60731	39814.522	5958.460188	326373.1812	28925.70549
9	Orli Casey	nunc.est.mollis@Suspendissetristiqueneq	Czech Rep	1	50.19302	51752.23445	10985.69656	629312.4041	47434.98265
10	Marny Obrien	Phasellus@sedsemegestas.org	Armenia	0	46.58474	58139.2591	3440.823799	630059.0274	48013.6141
11	Rhonda Chavez	nec@nuncest.com	Somalia	1	43.32378	53457.10132	12884.07868	476643.3544	38189.50601
12	Jerome Rowe	ipsum.cursus@dui.org	Sint Maart	1	50.12992	73348.70745	8270.707359	612738.6171	59045.51309
13	Alfred Gill	ipsum.cursus@dui.org	Sint Maart	1	50.12992	73348.70745	8270.707359	612738.6171	59045.51309
									목표값

속성 (정보)

Customer name / Customer e-mail / Country / Gender / Age / Annual Salary / Credit Card Debt / Net Worth

예측하고자 하는 값

Car Punchase Amount

인공신경망-자동차 판매 예측

Library import와 DataSet 불러오기

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

pandas -> 테이블 조작

numpy -> 수치해석

Matplotlib, seaborn -> 시각화

인공신경망-자동차 판매 예측

Library import와 DataSet 불러오기

```
car_df = pd.read_csv('Car_Purchasing_Data.csv', encoding='ISO-8859-1')
```

```
In [2]: car_df = pd.read_csv('Car_Purchasing_Data.csv', encoding='ISO-8859-1')
```

```
In [3]: car_df
```

```
Out[3]:
```

	Customer Name	Customer e-mail	Country	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
0	Martina Avila	cubilia.Curae.Phasellus@quisaccumsanconvallis.edu	Bulgaria	0	41.851720	62812.09301	11609.380910	238961.2505	35321.45877
1	Harlan Barnes	eu.dolor@diam.co.uk	Belize	0	40.870623	66646.89292	9572.957136	530973.9078	45115.52566
2	Naomi Rodriguez	vulputate.mauris.sagittis@ametconsectetueradip...	Algeria	1	43.152897	53798.55112	11160.355060	638467.1773	42925.70921
3	Jade Cunningham	malesuada@dignissim.com	Cook Islands	1	58.271369	79370.03798	14426.164850	548599.0524	67422.36313
4	Cedric Leach	felis.ullamcorper.viverra@egetmollislectus.net	Brazil	1	57.313749	59729.15130	5358.712177	560304.0671	55915.46248
...
495	Walter	ligula@Cumsociis.ca	Nepal	0	41.462515	71942.40291	6995.902524	541670.1016	48901.44342
496	Vanna	Cum.sociis.natoque@Sedmolestie.edu	Zimbabwe	1	37.642000	56039.49793	12301.456790	360419.0988	31491.41457
497	Pearl	penatibus.et@massanonte.com	Philippines	1	53.943497	68888.77805	10611.606860	764531.3203	64147.28888

인공신경망-자동차 판매 예측

Encoding : 사용자가 입력한 문자나 기호들을 **컴퓨터가 이용할 수 있는 신호**로 만드는 것



8bit 확장된 **ASCII**를 최초로 공식화한 국제 표준이 **ISO-8859** 시리즈 임.

인공신경망-자동차 판매 예측

Library import와 DataSet 불러오기

`car_df.head(5)`

`car_df.tail(2)`

```
In [11]: car_df.head(5)
         car_df.tail(2)
```

Out [11]:

	Customer Name	Customer e-mail	Country	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
498	Nell	Quisque.varius@arcuVivamussit.net	Botswana	1	59.160509	49811.99062	14013.034510	337826.6382	45442.15353
499	Marla	Camaron.marla@hotmail.com	marlal	1	46.731152	61370.67766	9391.341628	462946.4924	45107.22566

인공신경망-자동차 판매 예측

Library import와 DataSet 불러오기

`car_df.head(5)`

`car_df.tail(2)`

```
In [11]: car_df.head(5)
         car_df.tail(2)
```

Out [11]:

	Customer Name	Customer e-mail	Country	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
498	Nell	Quisque.varius@arcuVivamusSsit.net	Botswana	1	59.160509	49811.99062	14013.034510	337826.6382	45442.15353
499	Marla	Camaron.marla@hotmail.com	marlal	1	46.731152	61370.67766	9391.341628	462946.4924	45107.22566

`car_df.info()`

`car_df = pd.read_csv('Car_Purchasing_Data.csv', usecols=['Age','Net Worth'], encoding='ISO-8859-1')`

`car_df = pd.read_csv('Car_Purchasing_Data.csv', usecols=[4,7], encoding='ISO-8859-1')`

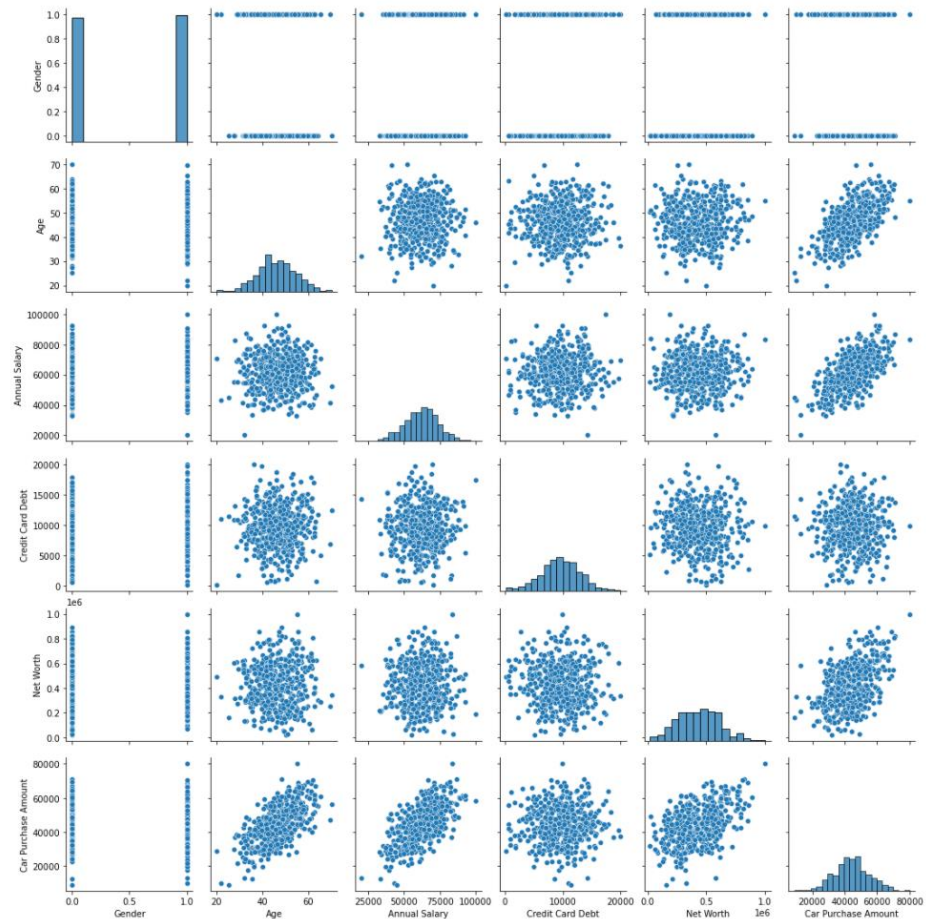
인공신경망-자동차 판매 예측

Visualize DATASET

`sns.pairplot(car_df)`

```
In [34]: sns.pairplot(car_df)
```

```
Out [34]: <seaborn.axisgrid.PairGrid at 0x1e87a6ce648>
```



인공신경망-자동차 판매 예측

Axis 이해

```
a = {'col1' : [1, 3, 5, 7, 9], 'col2' : ['a', 'b', 'c', 'd', 'e'], 'col3' : [10, 9, 8, 7, 6]}
e_df = pd.DataFrame(a)
e_df
```

	col1	col2	col3
0	1	a	10
1	3	b	9
2	5	c	8
3	7	d	7
4	9	e	6

Axis = 0(행삭제) , Axis = 1(열삭제)

```
# 0, 3의 인덱스를 가지는 행 삭제
df_1 = df.drop([0, 3], axis = 0)
```

```
# col2 열 삭제
df_2 = df.drop(['col2'], axis = 1)
```

인공신경망-자동차 판매 예측

데이터 정제

불필요한 열 삭제

```
X = car_df.drop([' Customer Name ', ' Customer e-mail ', ' Country' ], axis = 1)
```

axis = 1 -> column(열) axis = 0 -> rows(행)

```
In [8]: X = car_df.drop(['Customer Name', 'Customer e-mail', 'Country', 'Car Purchase Amount'], axis = 1)
```

```
In [9]: X
```

```
Out [9]:
```

	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
0	0	41.851720	62812.09301	11609.380910	238961.2505	35321.45877
1	0	40.870623	66646.89292	9572.957136	530973.9078	45115.52566
2	1	43.152897	53798.55112	11160.355060	638467.1773	42925.70921
3	1	58.271369	79370.03798	14426.164850	548599.0524	67422.36313
4	1	57.313749	59729.15130	5358.712177	560304.0671	55915.46248
...
495	0	41.462515	71942.40291	6995.902524	541670.1016	48901.44342
496	1	37.642000	56039.49793	12301.456790	360419.0988	31491.41457
497	1	53.943497	68888.77805	10611.606860	764531.3203	64147.28888
498	1	50.160500	40811.00062	14012.024510	227826.6382	45442.15252

500 rows x 5 columns

인공신경망 훈련이나 머신러닝, 딥러닝을 구현할 때 입력값은 대문자 **X**(모든 특성과 입력값) 출력값은 소문자 **y**로 표기

인공신경망-자동차 판매 예측

데이터 정제 - X값 문제 조건값만 남기기

불필요한 열 삭제

`X = car_df.drop([' Customer Name ' , ' Customer e-mail ' , ' Country ' , ' Car Purchase Amount '], axis = 1)`

`axis = 1 -> column axis = 0 -> rows`

```
In [8]: X = car_df.drop(['Customer Name', 'Customer e-mail', 'Country', 'Car Purchase Amount'], axis = 1)
```

```
In [9]: X
```

```
Out [9]:
```

	Gender	Age	Annual Salary	Credit Card Debt	Net Worth
0	0	41.851720	62812.09301	11609.380910	238961.2505
1	0	40.870623	66646.89292	9572.957136	530973.9078
2	1	43.152897	53798.55112	11160.355060	638467.1773
3	1	58.271369	79370.03798	14426.164850	548599.0524
4	1	57.313749	59729.15130	5358.712177	560304.0671
...
495	0	41.462515	71942.40291	6995.902524	541670.1016
496	1	37.642000	56039.49793	12301.456790	360419.0988
497	1	53.943497	68888.77805	10611.606860	764531.3203
498	1	59.160509	49811.99062	14013.034510	337826.6382
499	1	46.731152	61370.67766	9391.341628	462946.4924

500 rows x 5 columns

인공신경망 훈련이나 머신러닝, 딥러닝을 구현할 때 입력값은 대문자 **X**(모든 특성과 입력값) 출력값은 소문자 **y**로 표기

인공신경망-자동차 판매 예측

데이터 정제 -y 값 구하기

데이터 프레임의 차원 설명

y = car_df['Car Purchase Amount']

```
In [8]: X.shape
```

```
Out[8]: (500, 5)
```

```
In [9]: y = car_df['Car Purchase Amount']  
y.shape
```

```
Out[9]: (500,)
```


인공신경망-자동차 판매 예측

데이터 정제X_

입력값 X 데이터의 정규화

인공 신경망 훈련을 위한 입력값의 최소 및 최대값의 정규화 필요 =>

최소 최대 척도의 표준화 방법으로 0~1사이의 값으로 변환

```
In [8]: from sklearn.preprocessing import MinMaxScaler  
  
scaler = MinMaxScaler()  
X_scaled = scaler.fit_transform(X)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
```

```
X_scaled = scaler.fit_transform(X) #변환
```

```
print(X_scaled[:,0]) #모든행, 0번 컬럼값
```

```
Print(X_scaled[:5,0:3] # 0행~4행, 0번~2번 컬럼값
```

```
In [9]: scaler.data_max_
```

```
Out[9]: array([1.e+00, 7.e+01, 1.e+05, 2.e+04, 1.e+06])
```

```
In [10]: scaler.data_min_
```

```
Out[10]: array([ 0., 20., 20000., 100., 20000.])
```

```
In [11]: print(X_scaled[:,0])
```

```
[0. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1.  
0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0.  
1. 0. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 1. 1. 1. 0.  
0. 1. 1. 0. 1. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1.  
0. 1. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 0.  
1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 0. 0. 0. 0. 1. 1. 1. 0. 1. 0. 1.  
0. 0. 1. 0. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1.  
0. 1. 1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1.  
0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0.  
1. 1. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 0. 1.  
0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1.  
0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0.  
0. 0. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 1. 0. 0.  
0. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0.  
0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 1.  
1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 1.  
0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1.  
0. 1. 0. 1. 1. 0. 0. 1. 1. 0. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0.  
1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 0. 1. 1.  
0. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 0.  
0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 0.  
0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1.]
```

인공신경망-자동차 판매 예측

데이터 정제

```
y_scaled = scaler.fit_transform(y)
```

(오류)

```
y = y.values.reshape(-1,1)
```

```
y_scaled = scaler.fit_transform(y)
```

```
In [12]: y.shape
```

```
Out[12]: (500,)
```

```
In [13]: y = y.values.reshape(-1,1)
```

```
In [14]: y.shape
```

```
Out[14]: (500, 1)
```

```
In [14]: y_scaled = scaler.fit_transform(y)
```

```
49991.60697 61731.71426 41769.38288 46402.53583 37376.63439 33766.6413
30667.60927 52056.41478 30736.5798 39439.45349 38174.87433 40589.8625
62028.71192 48465.27211 40095.0498 49568.47685 31408.62631 47719.47741
35784.42411 42905.53815 48516.84335 45593.6849 32061.6467 32208.37522
35475.00344 29519.56184 55420.56668 42139.64528 50539.90169 34922.42846
43898.2733 39135.03023 41147.46679 24134.59205 42705.11311 38901.60925
28645.39425 52150.41786 66648.25077 42909.27129 49248.10595 27303.17104
47869.82593 59984.16361 45271.46081 9000. 46012.10616 32967.20191
48785.15839 45824.5656 40102.11417 35457.1486 29556.7932 38243.06228
44430.63323 51046.42226 52570.36517 61404.22578 28463.64326 27586.20078
47979.48549 28164.86039 69669.47402 48052.65091 37364.23474 44500.81936
35139.24793 55167.37361 48383.69071 35823.55471 36517.70996 53110.88052
53049.44567 21471.11367 45015.67953 55377.87697 56510.13294 47443.74443
41489.64123 32553.53423 41984.62412 59538.40327 41352.47071 52785.16947
60117.67886 47760.66427 64188.26862 48901.44342 31491.41457 64147.28888
45442.15353 45107.22566].
```

Reshape your data either using `array.reshape(-1, 1)` if your data has a single feature or `array.reshape()` if you have multiple features.

인공신경망-자동차 판매 예측

데이터 정제 완료

X_scaled

```
array([[0.          , 0.4370344 , 0.53515116, 0.57836085, 0.22342985],
       [0.          , 0.41741247, 0.58308616, 0.476028   , 0.52140195],
       [1.          , 0.46305795, 0.42248189, 0.55579674, 0.63108896],
       ...,
       [1.          , 0.67886994, 0.61110973, 0.52822145, 0.75972584],
       [1.          , 0.78321017, 0.37264988, 0.69914746, 0.3243129 ],
       [1.          , 0.53462305, 0.51713347, 0.46690159, 0.45198622]])
```

X_scaled

```
array([[0.37072477],
       [0.50866938],
       [0.47782689],
       [0.82285018],
       [0.66078116],
       [0.67059152],
       [0.28064374],
       [0.54133778],
       [0.54948752],
       [0.4111198 ],
       [0.70486638],
       [0.46885649],
       [0.27746526],
       [0.56702642],
       [0.57056385],
       [0.61996151],
       [0.46217916],
       [0.49157341],
       [0.50188722],
       [0.64545808],
```

인공신경망-자동차 판매 예측

Training The Model

```
from sklearn.model_selection import train_test_split
```

데이터를 훈련용과 테스트 용으로 분리하기 위해 `train_test_split` 호출

`Test_size` 는 분할 비율 (기본값은 0.25)

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled, test_size = 0.25)
```

```
In [34]: X_scaled.shape
```

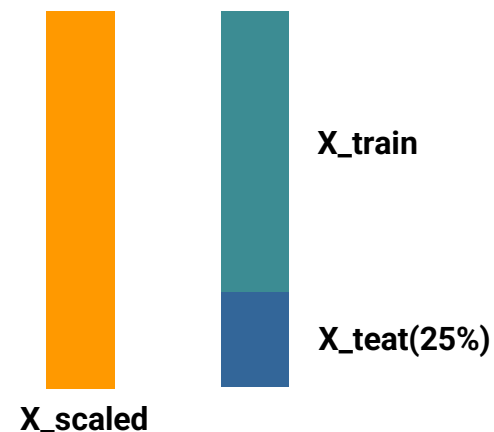
```
Out[34]: (500, 5)
```

```
In [35]: from sklearn.model_selection import train_test_split  
t_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled)
```

```
In [36]: X_train.shape
```

```
Out[36]: (375, 5)
```

125개가 테스트 샘플이 됨



인공신경망-자동차 판매 예측

Training The Model – Keras – 고수준 신경망 API

```
import tensorflow.keras
```

```
from keras.models import Sequential
```

```
from keras.layers import Dense
```

```
from sklearn.preprocessing import MinMaxScaler
```

Keras는 Tensorflow 위에서 동작하는 API

Sequential 은 신경망을 순차적 형태로 설계하는 것 의미

특정한 층에서 모든 출력값이 나오는 완전연결 인공 신경망

```
model = Sequential()    모델을 순차적 형태로 설계
```

```
model.add(Dense(25, input_dim=5, activation='relu'))    첫번째 층 (Dense(뉴런수, input_dim=입력값 수,  
activation='relu')
```

```
model.add(Dense(25, activation='relu'))    추가 은닉층
```

```
model.add(Dense(1, activation='linear'))    출력값은 하나,
```

```
model.summary()    설계된 모델에 대한 스냅샷 제공
```

샘플 (입력되는 Sequence개수)*time step(sequence내 관측치)*features

인공신경망-자동차 판매 예측

Training The Model – Keras – 고수준 신경망 API

```
In [38]: import tensorflow.keras
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import MinMaxScaler

model = Sequential()
model.add(Dense(25, input_dim=5, activation='relu'))
model.add(Dense(25, activation='relu'))
model.add(Dense(1, activation='linear'))
model.summary()
```

Using TensorFlow backend.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 25)	150
dense_2 (Dense)	(None, 25)	650
dense_3 (Dense)	(None, 1)	26
Total params: 826		
Trainable params: 826		
Non-trainable params: 0		

입력값 5개
뉴런 25
완전연결
뉴런마다 바이어스 신호 1개씩
 $5 \times 25 + 25 = 150$

뉴런이 각각 40개씩 있을 경우 파라미터 수는?

인공신경망-자동차 판매 예측

Training The Model – Keras – 고수준 신경망 API

`model.compile(optimizer='adam', loss='mean_squared_error')` 평균 제곱 오차 최소화(손실함수)

`epochs_hist = model.fit(X_train, y_train, epochs=20, batch_size=25, verbose=1, validation_split=0.2)` 모델훈련

batch_size => 신경망에 노출되는 데이터 포인트 수 (한번의 **Batch**마다 주는 데이터 샘플의 크기)

validation split => 과적합(학습해도 전혀 이해하지 못함, IQ)을 확인하기 위해 설정

```
In [41]: epochs_hist = model.fit(X_train, y_train, epochs=20, batch_size=25, verbose=1, validation_split=0.2)
```

Train on 300 samples, validate on 75 samples

Epoch 1/20

300/300 [=====] - 0s 967us/step - loss: 0.2515 - val_loss: 0.0854

Epoch 2/20

300/300 [=====] - 0s 90us/step - loss: 0.0427 - val_loss: 0.0572

Epoch 3/20

300/300 [=====] - 0s 106us/step - loss: 0.0404 - val_loss: 0.0389

Epoch 4/20

300/300 [=====] - 0s 93us/step - loss: 0.0260 - val_loss: 0.0365

Epoch 5/20

300/300 [=====] - 0s 106us/step - loss: 0.0237 - val_loss: 0.0324

Epoch 6/20

300/300 [=====] - ETA: 0s - loss: 0.011 - 0s 100us/step - loss: 0.0225 - val_loss: 0.0315

Epoch 7/20

300/300 [=====] - 0s 96us/step - loss: 0.0222 - val_loss: 0.0320

Epoch 8/20

신경망이 0.25의 손실로 시작해서 0.02로 작아짐 => 인공신경망이 훈련되어 평균 제곱오차가 줄어드는 것을 의미

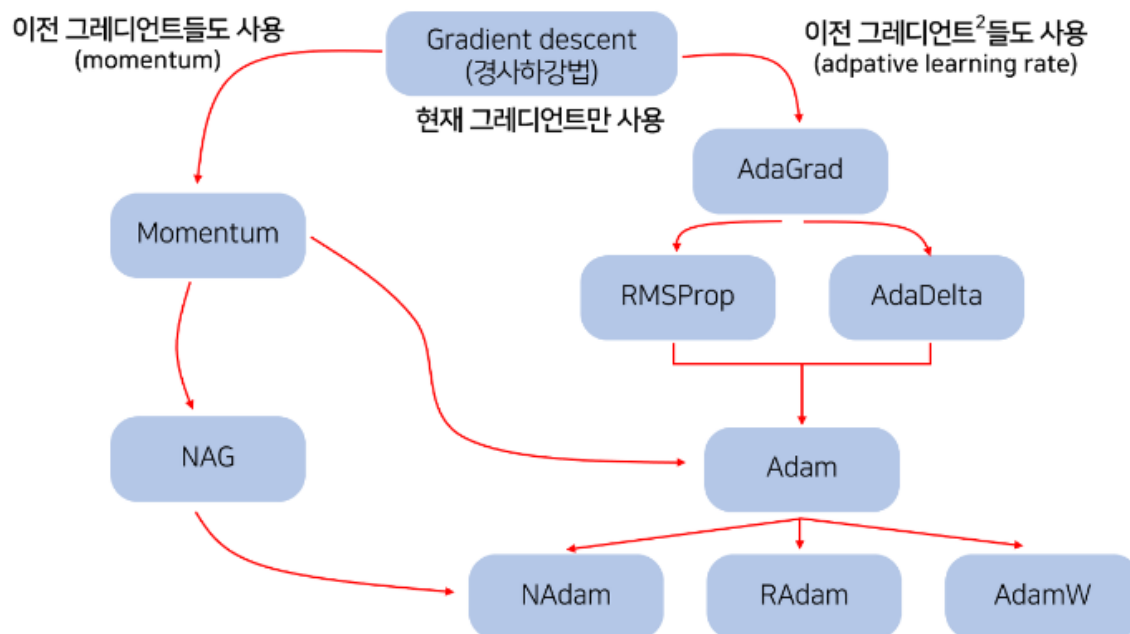
Epochs 값을 늘려보세요

인공신경망-자동차 판매 예측

Training The Model – Keras – 고수준 신경망 API

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

Optimizer 종류



인공신경망-자동차 판매 예측

모델평가(EVALUATING The Model)

`print(epochs_hist.history.keys())` 내부정보를 보여줌

`plt.plot(epochs_hist.history['loss'])` 각 백터에 대한

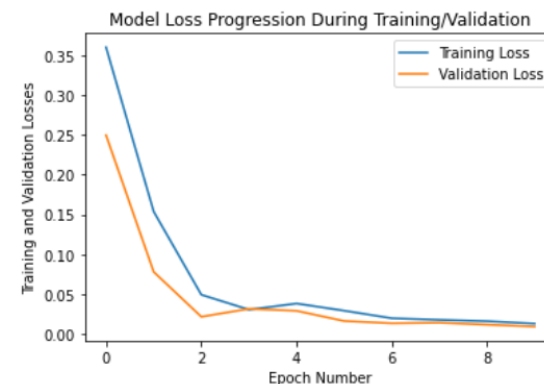
`plt.plot(epochs_hist.history['val_loss'])`

`plt.title('Model Loss Progression During Training/Validation')`

`plt.ylabel('Training and Validation Losses')`

`plt.xlabel('Epoch Number')`

`plt.legend(['Training Loss', 'Validation Loss'])`



Epoch 값 변경 100->8

인공신경망-자동차 판매 예측

모델 평가 - 주어진 무작위 입력값으로 값 예측

```
X_Testing = np.array([[1, 50, 50000, 10000, 600000]])  성별, 나이, 연봉, 카드빚, 순자산 순으로 입력
y_predict = model.predict(X_Testing)
y_predict.shape
print('Expected Purchase Amount=', y_predict[:,0])
```

훈련가중치는 **model**에 등록되어 있음

Activation function(시그모이드, 렐루)

그 외의 활성화 함수 (ReLU Function)

- Rectified Linear Unit (정류 선형 유닛)
- 기존의 사용되던 sigmoid를 개선(rectified)
- 입력값이 0보다 작으면 0으로 출력, 0보다 크면 그대로 출력
- 학습속도가 빨라진다는 장점(빠르게 수렴)
 - 미분 계산이 매우 간단(0이하는 0, 0이상은 1)
- ReLU에서 조금 더 발전된 Leaky ReLU, Parametric ReLU도 등장

ReLU
 $\max(0, x)$

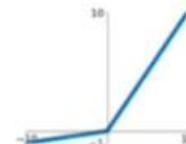


ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Leaky ReLU
 $\max(0.1x, x)$



Sigmoid Function

- Sigmoid: S자 모양의, 결장의 S자형 만곡부(의학)
- 일차함수(선형회귀)와 달리 smoot하게 변화
- 함수로서 유계이며, 미분가능한 함수
- 모든 점에서 음이 아닌 미분값(단조증가함수)
- 단 하나의 변곡점을 가짐 ($x=0$)

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

