

# 311콜센터 통화량 예측

## 311 통화량 예측의 중요성

주어진 달의 서비스 통화량의 예측은 1년 동안의 통화량과 관련된 날짜와 시간을 살펴보고, 그 통화를 주간 단위로 집계하여  $x$  값이 주 번호(1-52, 또는 365일을 7일로 나눈 값)이고  $y$  값이 특정 주의 통화 수인 포인트 집합을 구성함.

## 311콜센터의 통화량 예측 과정

- 1 — 데이터 정제 및 시각화  
y축에 통화 수를, x축에 주 번호(1-52)를 플롯.
- 2 — 추세 분석  
추세를 검토하여 선, 곡선 또는 다른 것과 유사한지 확인.
- 3 — 모델 선택 및 훈련  
데이터 포인트(주 번호 및 통화 수)에 가장 잘 맞는 회귀 모델을 선택하고 훈련시킴.
- 4 — 모델 평가  
오차를 계산하고 시각화하여 모델의 성능을 평가함.
- 5 — 예측 수행  
새로운 모델을 사용하여 311이 주어진 주, 계절 및 연도에 예상할 수 있는 통화 수를 예측함.

## 실제 데이터 (xls)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R					
1	Unique Kt	Created Date	Closed Date	Agency	Agency N	Complaint	Descriptor	Location	T	Incident Z	Incident A	Street	Nar	Cross	Stre	Cross	Stre	Intersectio	Intersectio	Address	Ty	City	L
2	28157590	06/01/2014 12:00:00 AM	06/06/2014 04:03:44 PM	DOHMH	Departme	Rodent	Mouse	Sig 1-2 Family	11372	70-04	ROX	ROOSEVELT	70 STREET	BQE	WESTBOUND	ENTRANCE				ADDRESS	Jackson Hei		
3	28157974	06/01/2014 12:00:00 AM	06/10/2014 12:00:00 AM	HPD	Departme	UNSANITZ	GARBAGE	RESIDENT	10024	336	WEST	WEST 77	S	WEST	ENER	RIVERSIDE	DRIVE			ADDRESS	NEW YORK		
4	28158733	06/01/2014 12:00:00 AM	06/09/2014 12:00:00 AM	HPD	Departme	UNSANITZ	PESTS	RESIDENT	11355	140-37	AS	ASH	AVEN	KISSENA	E	BOWNE	STREET			ADDRESS	Flushing		
5	28159418	06/01/2014 12:00:00 AM	06/20/2014 12:00:00 AM	HPD	Departme	UNSANITZ	MOLD	RESIDENT	10458	2604	BAIN	BAINBRID	EAST 193	EAST 194	STREET					ADDRESS	BRONX		
6	28160237	06/01/2014 12:00:00 AM	06/10/2014 12:00:00 AM	HPD	Departme	UNSANITZ	GARBAGE	RESIDENT	10024	336	WEST	WEST 77	S	WEST	ENER	RIVERSIDE	DRIVE			ADDRESS	NEW YORK		
7	28160282	06/01/2014 12:00:00 AM	06/09/2014 12:00:00 AM	HPD	Departme	WATER	LE	HEAVY FL	RESIDENT	11355	140-37	AS	ASH	AVEN	KISSENA	E	BOWNE	STREET		ADDRESS	Flushing		
8	28162423	06/01/2014 12:00:00 AM	06/09/2014 12:00:00 AM	HPD	Departme	WATER	LE	HEAVY FL	RESIDENT	11355	140-37	AS	ASH	AVEN	KISSENA	E	BOWNE	STREET		ADDRESS	Flushing		
9	28162488	06/01/2014 12:00:00 AM	06/20/2014 12:00:00 AM	HPD	Departme	UNSANITZ	MOLD	RESIDENT	10458	2604	BAIN	BAINBRID	EAST 193	EAST 194	STREET					ADDRESS	BRONX		
10	28162548	06/01/2014 12:00:00 AM	06/09/2014 12:00:00 AM	HPD	Departme	WATER	LE	HEAVY FL	RESIDENT	11355	140-37	AS	ASH	AVEN	KISSENA	E	BOWNE	STREET		ADDRESS	Flushing		
11	28162768	06/01/2014 12:00:00 AM	06/10/2014 12:00:00 AM	DOHMH	Departme	Rodent	Mouse	Sig 3+ Family	11222	60	NORM	NORMAN	GUERNSEY	LORIMER	STREET					ADDRESS	BROOKLYN		
12	28162769	06/01/2014 12:00:00 AM	06/06/2014 12:00:00 AM	DOHMH	Departme	Rodent	Mouse	Sig 3+ Family	11358	158-10	SA	SANFORD	158	STREE	159	STREET				ADDRESS	Flushing		
13	28162823	06/01/2014 12:00:00 AM	06/05/2014 12:00:00 AM	HPD	Departme	HEAT/HO	APARTME	RESIDENT	11218	201	DITM	DITMAS	A	EAST 2	ST	EAST 3	STREET			ADDRESS	BROOKLYN		
14	28162843	06/01/2014 12:00:00 AM	06/20/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli 3+ Family	11213	1481	DEAI	DEAN	STR	REVERE	P	ALBANY	AVENUE		ADDRESS	BROOKLYN			
15	28162878	06/01/2014 12:00:00 AM	06/04/2014 03:31:51 PM	DOHMH	Departme	Rodent	Rat	Sightli 3+ Family	10456	1115	FRAH	FRANKLIN	EAST 166	EAST 167	STREET					ADDRESS	BRONX		
16	28162891	06/01/2014 12:00:00 AM	06/01/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli 3+ Family	10458	2364	TIEB	TIEBOUT	EAST 184	EAST 187	STREET					ADDRESS	BRONX		
17	28162892	06/01/2014 12:00:00 AM	06/01/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli 3+ Family	10458	2364	TIEB	TIEBOUT	EAST 184	EAST 187	STREET					ADDRESS	BRONX		
18	28162893	06/01/2014 12:00:00 AM	06/10/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli 3+ Family	10460	2146	VYSE	VYSE	AVE	EAST 181	BRONX	PARK	SOUTH		ADDRESS	BRONX			
19	28162895	06/01/2014 12:00:00 AM	06/01/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli Other (Exg	11237	HIMROD	HIMROD	WYCKOFF	ST	NICHOLAS	AVENUE				BLOCKFAX	BROOKLYN			
20	28162909	06/01/2014 12:00:00 AM	06/09/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli 1-2 Family	11223	1618	WES	WEST 7	STAVENUE	F	QUENTIN	ROAD			ADDRESS	BROOKLYN			
21	28162928	06/01/2014 12:00:00 AM	06/01/2014 12:00:00 AM	DOHMH	Departme	Rodent	Rat	Sightli Other (Exg	10025	146	WEST	WEST	95	COLUMBLA	AMSTERDAM	AVENUE			ADDRESS	NEW YORK			
22	28163037	06/01/2014 12:00:00 AM	06/16/2014 12:00:00 AM	HPD	Departme	APPLIANC	REFRIGER	RESIDENT	10458	2325	ARTH	ARTHUR	A	CRESCENT	EAST 186	STREET			ADDRESS	BRONX			
23	28163041	06/01/2014 12:00:00 AM	06/06/2014 12:00:00 AM	HPD	Departme	PLUMBINK	WATER	SL	RESIDENT	11218	483	OCEA	OCEAN	P	CORTELYC	DITMAS	AVENUE		ADDRESS	BROOKLYN			
24	28163068	06/01/2014 12:00:00 AM	06/10/2014 12:00:00 AM	DOHMH	Departme	Unsanitan	Other	Anli 1-2 Family	10453	65	EAST 1	EAST 175	TOWNSEN	WALTON	AVENUE				ADDRESS	BRONX			
25	28163069	06/01/2014 12:00:00 AM	06/16/2014 12:00:00 AM	HPD	Departme	PLUMBINK	WATER	SL	RESIDENT	11203	141	EAST	EAST 54	S	LENOX	R	CLINDEN	BOULEVARD		ADDRESS	BROOKLYN		
26	28163092	06/01/2014 12:00:00 AM	06/16/2014 12:00:00 AM	HPD	Departme	PLUMBINK	WATER	SL	RESIDENT	11203	141	EAST	EAST 54	S	LENOX	R	CLINDEN	BOULEVARD		ADDRESS	BROOKLYN		
27	28163094	06/01/2014 12:00:00 AM	06/07/2014 12:00:00 AM	HPD	Departme	UNSANITZ	GARBAGE	RESIDENT	11235	3232	SHO	SHORE	P	AHOME	CREAST	13	STREET		ADDRESS	BROOKLYN			
28	28163097	06/01/2014 12:00:00 AM	06/27/2014 12:00:00 AM	HPD	Departme	PLUMBINK	WATER	SL	RESIDENT	11421	94-46	85	F85	ROAD	94	STREET	96	STREET		ADDRESS	Woodhaver		
29	28163106	06/01/2014 12:00:00 AM	06/15/2014 12:00:00 AM	HPD	Departme	PLUMBINK	WATER	SL	RESIDENT	11203	175	EAST	EAST 52	S	WINTHRO	CLARKSON	AVENUE		ADDRESS	BROOKLYN			
30	28163117	06/01/2014 12:00:00 AM	06/20/2014 12:00:00 AM	HPD	Departme	UNSANITZ	PESTS	RESIDENT	10458	2475	HUGH	HUGHES	EAST 188	EAST 189	STREET				ADDRESS	BRONX			
31	28163121	06/01/2014 12:00:00 AM	06/19/2014 12:00:00 AM	DOHMH	Departme	Rodent	Condition	3+ Family	11201	41	SCHER	SCHERME	CLINTON	COURT	STREET				ADDRESS	BROOKLYN			
32	28163122	06/01/2014 12:00:00 AM	06/01/2014 12:00:00 AM	DOHMH	Departme	Rodent	Condition	1-2 Family	10458	3010	GRAI	GRANDCC	EAST 201	EAST 202	STREET				ADDRESS	BRONX			
33	28163130	06/01/2014 12:00:00 AM	06/05/2014 12:00:00 AM	HPD	Departme	WATER	LE	DAMP	SP	RESIDENT	10457	1663	EAST	EASTBUR	EAST MT	EAST 173	STREET		ADDRESS	BRONX			
34	28163131	06/01/2014 12:00:00 AM	06/05/2014 12:00:00 AM	HPD	Departme	WATER	LE	DAMP	SP	RESIDENT	10457	1663	EAST	EASTBUR	EAST MT	EAST 173	STREET		ADDRESS	BRONX			

1. 빈 값 찾기

2. 날짜 파싱

3. 원하는 년도에 따라 그 주의 통화량 빈도 구하기

## 실습I. 데이터 정제 및 시각화

```
8 import csv
9 import time
10 import numpy as np
11 import matplotlib.pyplot as plt
12 import tensorflow as tf
13
14 def read(filename, date_idx, date_parse, year=None, bucket=7):
15     days_in_year = 365
16
17     freq = {}
18     if year != None:
19         for period in range(0, int(days_in_year / bucket)):
20             freq[period] = 0
21
22     with open(filename, 'r') as csvfile:
23         csvreader = csv.reader(csvfile)
24         next(csvreader)
25         for row in csvreader:
26             if row[date_idx] == '':
27                 continue
```

```
        #날짜 파싱
        t = time.strptime(row[date_idx], date_parse) #'%m/%d/%Y %H:%M:%S %p'로 데이터 파싱
        #데이터 필터링 및 집계
        if year == None:
            if not t.tm_year in freq:
                freq[t.tm_year] = {}
            for period in range(0, int(days_in_year / bucket)):
                freq[t.tm_year][period] = 0

            if t.tm_yday < (days_in_year - 1):
                freq[t.tm_year][int(t.tm_yday / bucket)] += 1
            print(2)

        #특정 연도의 날짜를 데이터를 주단위로 나누어 각 기간별로 빈도를 집계하는 과정
        else:
            if t.tm_year == year and t.tm_yday < (days_in_year-1):
                freq[int(t.tm_yday / bucket)] += 1

    return freq

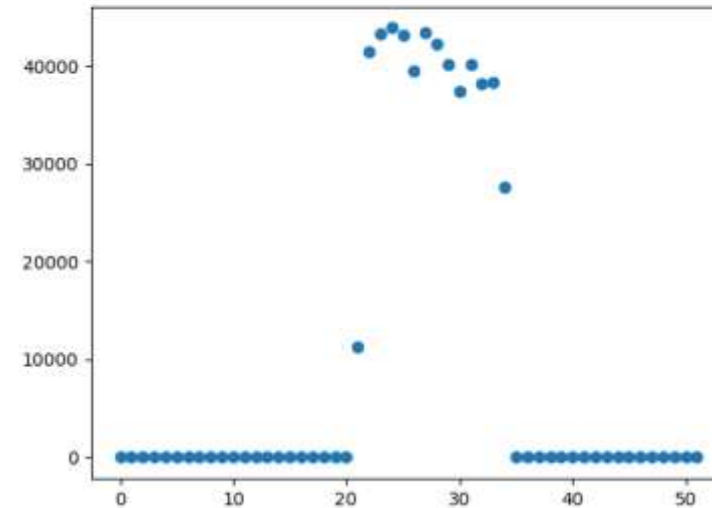
freq = read(r'C:\Users\dahae\machine_learning\311_call_center.csv' , 1, '%m/%d/%Y %H:%M:%S %p', 2014)

X_train = np.asarray(list(freq.keys()))
Y_train = np.asarray(list(freq.values()))
print(freq)
plt.scatter(X_train,Y_train)
plt.show()
```

## 1. 데이터 정제 결과 확인

```
freq (0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0,  
20: 0, 21: 10889, 22: 40240, 23: 42125, 24: 42673, 25: 41721, 26: 38446, 27: 41915, 28: 41008, 29: 39011, 30:  
36069, 31: 38821, 32: 37050, 33: 36967, 34: 26834, 35: 0, 36: 0, 37: 0, 38: 0, 39: 0, 40: 0, 41: 0, 42: 0, 43: 0, 44: 0, 45:  
0, 46: 0, 47: 0, 48: 0, 49: 0, 50: 0, 51: 0)
```

## 2. 데이터 정제 시각화

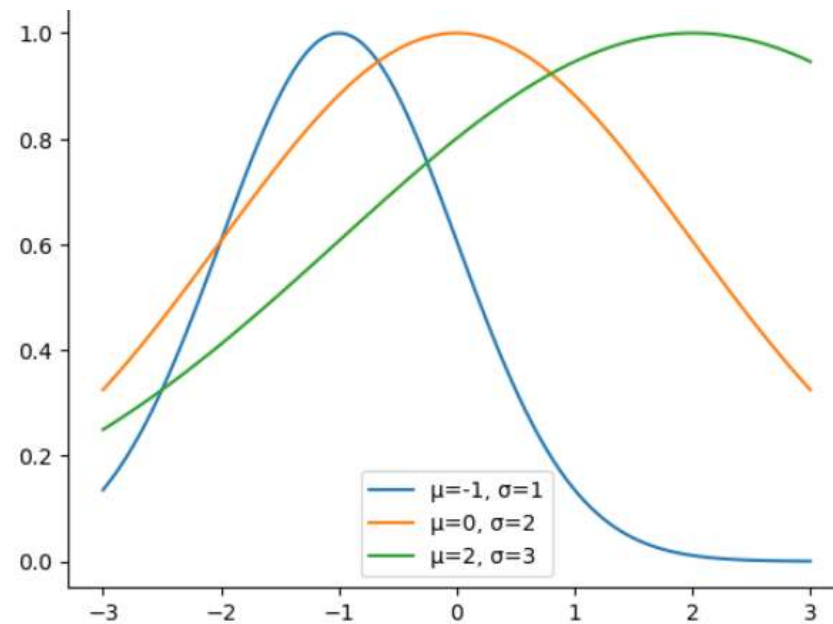


### 가우시안 분포

```
import numpy as np
import matplotlib.pyplot as plt

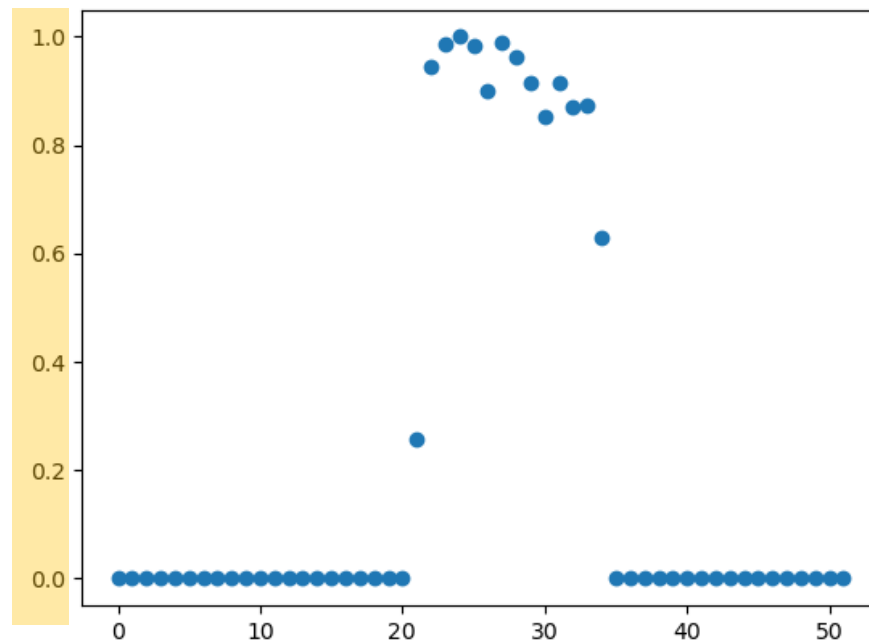
def gaussian(x, mu, sig):
    return np.exp(-np.power(x - mu, 2.) / (2 * np.power(sig, 2.)))
x_values = np.linspace(-3, 3, 120)
for mu, sig in [(-1, 1), (0, 2), (2, 3)]:
    plt.plot(x_values, gaussian(x_values, mu, sig), label=f' $\mu={mu}, \sigma={sig}$ ')

plt.legend()
plt.show()
```



정규화는 데이터를 일정한 범위로 변환하는 과정

```
maxY = np.max(Y_train)
nY_train = Y_train / np.max(Y_train)
plt.scatter(X_train, nY_train)
plt.show()
```



### 실습 3. 가우시안 모델 구축

```
class Model:
    def __init__(self):
        self.mu = tf.Variable(1.0, dtype=tf.float32)
        self.sig = tf.Variable(1.0, dtype=tf.float32)
    def __call__(self, x):
        x_c = tf.cast(x, tf.float32)
        return tf.exp(-tf.pow(x_c - self.mu, 2.) / (2. * tf.pow(self.sig, 2.)))
```

```
def cost_function(predicted_y, desired_y):
    loss_function = tf.square(predicted_y - desired_y)
    return tf.reduce_mean(loss_function)
```

비용 함수

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

learning\_rate = 1.5 → 학습률

optimizer = tf.keras.optimizers.SGD(learning\_rate)



## 실습 3. 가우시안 모델 구축

```
def train_step(model, inputs, outputs):
```

```
    with tf.GradientTape() as t:
```

```
        current_cost_function = cost_function(model(inputs), outputs)
```

```
    grads = t.gradient(current_cost_function, [model.mu, model.sig])
```

```
    optimizer.apply_gradients(zip(grads, [model.mu, model.sig]))
```

```
    return current_cost_function
```

```
model = Model()
```

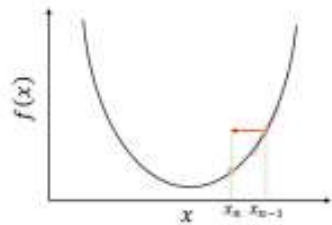
→ 기울기 계산

손실 함수 계산

손실함수를 반영한 기울기 계산

파라미터 Update

### 경사 하강법



경사 하강법 (Gradient descent)의 한 스텝

경사 하강법은  $f(x)$ 의 값이 변하지 않을 때까지 **스텝을 반복**한다.

1-D의 경우

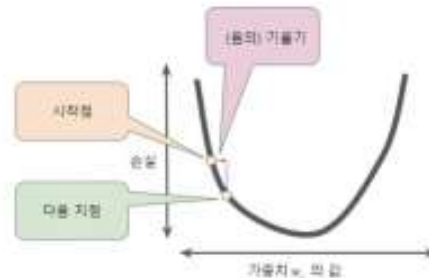
$$x_n = x_{n-1} - \alpha \frac{df(x_{n-1})}{dx}$$

N-D의 경우

$$\mathbf{x}_n = \mathbf{x}_{n-1} - \alpha \nabla f(\mathbf{x}_{n-1})$$

$\alpha$ : 학습률 (Learning rate)

< 손실 함수(Cost Function) >



$$x_0 = x_0 - \eta \frac{\partial f}{\partial x_0}$$

$\eta$ : step size(기계학습에서 learning rate라고도 함)

$\frac{\partial f}{\partial x_0}$ : 기울기를 의미한다.



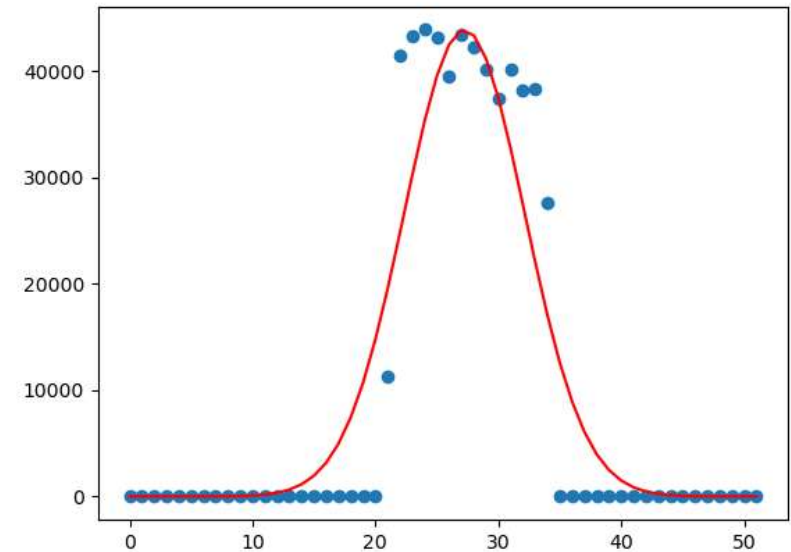
## 실습 3. 가우시안 모델 구축

```
training_epochs = 50
for epoch in range(training_epochs):
    for i in range(0, len(X_train)):
        _cost_function = train_step(model, X_train[i], nY_train[i])
    if epoch % 10 == 0:
        print("Current cost_function %f" % (_cost_function.numpy()))
```

```
mu_val = model.mu
sig_val = model.sig
print(mu_val.numpy())
print(sig_val.numpy())

plt.scatter(X_train, Y_train)
trY2 = maxY * (np.exp(-np.power(X_train - mu_val, 2.) / (2 * np.power(sig_val, 2.))))
plt.plot(X_train, trY2, 'r')
plt.show()
print("Prediction of week 35", trY2[33])
print("Actual week 35", Y_train[33])
```

→ 시각화



## 실습 4. 딥러닝 모델 코드

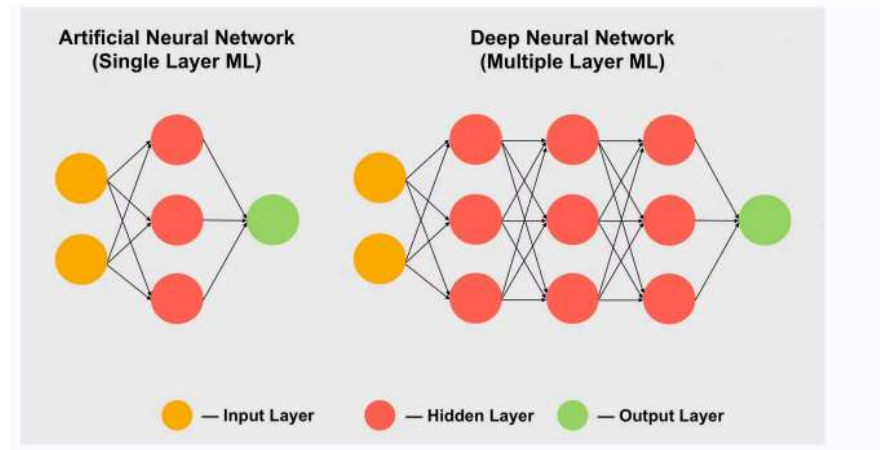
```
8
9 import csv
10 import time
11 import numpy as np
12 import matplotlib.pyplot as plt
13 from tensorflow.keras import layers, models
14 from tensorflow.keras.optimizers import SGD
15 import tensorflow as tf
16
17 def read(filename, date_idx, date_parse, year=None, bucket=7):
18     days_in_year = 365
19
20     freq = {}
21     if year != None:
22         for period in range(0, int(days_in_year / bucket)):
23             freq[period] = 0
```

```
24 with open(filename, 'r') as csvfile:
25     csvreader = csv.reader(csvfile) #csv 파일을 열고
26     next(csvreader) #첫번째 행을 건너뛰(헤더 제거)
27
28     for row in csvreader:
29         #빈 데이터 확인
30         if row[date_idx] == '': #날짜 피드가 빈 경우 건너뛰
31             continue
32
33         #날짜 파싱
34         t = time.strptime(row[date_idx], date_parse) #'%m/%d/%Y %H:%M:%S %p'로 데이터 파싱
35         print(t)
36         #데이터 필터링 및 집계
37         if year == None:
38             if not t.tm_year in freq:
39                 freq[t.tm_year] = {}
40                 for period in range(0, int(days_in_year / bucket)):
41                     freq[t.tm_year][period] = 0
42
43             if t.tm_yday < (days_in_year - 1):
44                 freq[t.tm_year][int(t.tm_yday / bucket)] += 1
45
46         else:
47             if t.tm_year == year and t.tm_yday < (days_in_year-1):
48                 freq[int(t.tm_yday / bucket)] += 1
49
50     return freq
51 freq = read('C:/Users/dahae/machine learning/311_call_center.csv', 1, '%m/%d/%Y %H:%M:%S %p', 2014)
```

## 실습 4. 딥러닝 모델 코드

```
53 X_train = np.asarray(list(freq.keys()))
54 Y_train = np.asarray(list(freq.values()))
55 #plt.scatter(X_train,Y_train)
56
57 model = models.Sequential([
58     layers.Input(shape=(1,)),
59     layers.Dense(32, activation='relu'),
60     layers.Dense(16, activation='relu'),
61     layers.Dense(8, activation='relu'),
62     layers.Dense(1)
63 ])
```

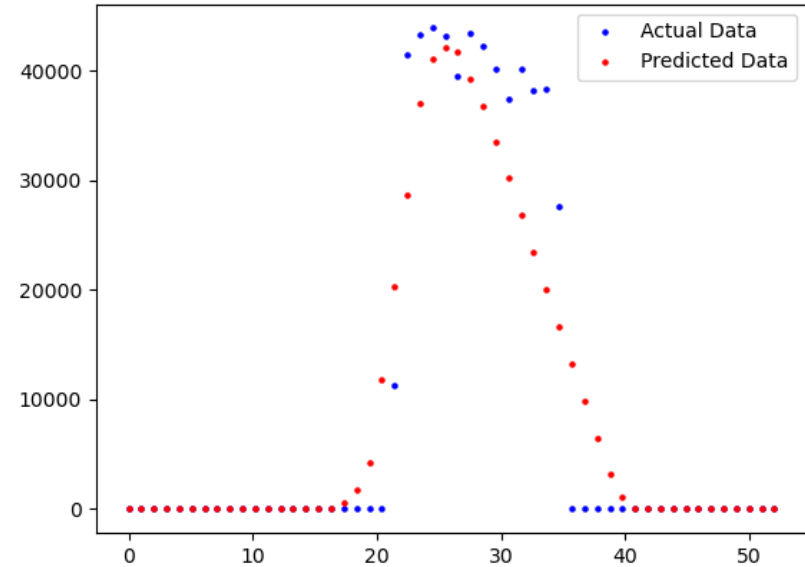
→ 모델 구축



## 실습 4. 딥러닝 모델 코드

```
65 ### start of norm on input data
66 from sklearn.preprocessing import MinMaxScaler
67 scaler = MinMaxScaler()
68 X_train = scaler.fit_transform(X_train.reshape(-1, 1)) #1D->2D
69 Y_train = scaler.fit_transform(Y_train.reshape(-1, 1)) #1D->2D
70
71 X_train = tf.convert_to_tensor(X_train, dtype=tf.float32)
72 Y_train = tf.convert_to_tensor(Y_train, dtype=tf.float32)
73
74 learning_rate = 0.1
75 training_epochs = 1000
76 optimizer = SGD(learning_rate=learning_rate)
77 model.compile(optimizer=optimizer, loss='mse', metrics=['accuracy'])
78 model.summary()
79 history = model.fit(X_train, Y_train, epochs=training_epochs, batch_size=13)
80
81 y_pred = model.predict(X_train)
82 y_pred=scaler.inverse_transform(y_pred) #원래의 크기로 복구
83 y_pred[y_pred < 0] = 0 #음수값이면 0
84 X_train = X_train.numpy()*52 #원래의 크기로 복구
85 Y_train = scaler.inverse_transform(Y_train)
86
87 plt.scatter(X_train, Y_train, color='blue', s=5, label='Actual Data')
88 plt.scatter(X_train, y_pred, color='red', s=5, label='Predicted Data')
89 plt.legend()
90 plt.show()
```

MinMaxScaler  
는 2차원 입력이  
들어가므로  
1D -> 2D



Batch의미 생각 해보고  
Batch를 작게 크게 실험해보기  
그리고 batch가 작을 때 클 때 장단점 생각해보기