

▶선형회귀에서 w와 b갯수

x값(특징)의 갯수에 맞추어서 w값의 갯수도 결정됨.
b는 무조건 1개임(또는 생략가능)

예) 나이,성별,자녀수,지역에 따르는 보험계약금예측

보험계약금예측값=나이*w1+성별*w2+자녀수*w3+지역*w4 + b임

나이	성별	자녀수	지역	이미발생한데이터 보험계약금
25	1	2	1	50
20	0	0	2	30
8	1	1	5	40
15	1	0	2	50
35	0	3	1	20
40	0	5	4	80

w1	w2	w3	w4	b
0.3	0.2	0.5	0.1	0.1
나이	성별	자녀수	지역	나이*w1+성별*w2+자녀수*w3+ 지역*w4+b
25	1	2	1	8.8
20	0	0	2	6.2
8	1	1	5	3.6
15	1	0	2	4.9
35	0	3	1	12.1
40	0	5	4	14.9

새로운데이터				예측할보험계약금
54	1	0	5	오차가 가장 작은 w와 b를 찾아서 새로운 데이터에 54*최적화w1+1*최적화w2+0*최적화w3+5*최적화w4+b 로 계산함

파라미터 (Parameter)

- 파라미터는 모델 내부에서 결정되는 변수입니다.
- 예
 - weight coefficient (가중치 계수)
 - bias (편향)
- weight coefficient, bias와 같은 파라미터들은 모델 내부에서 데이터를 통해 구해집니다.

좌측의 예제(보험금)으로는 w값4개
b값1개여서
파라미터는 5개가 생성됨

개요(히든레이어와 파라미터)

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
```

```
x=np.array([1,2,3,4,5,6])
y=np.array([10,98,8,2,3,4])
```

w값 바꾸지 않기 위해 사용함,검증용이며 실제에서는 seed 하지 않음.
tf.random.set_seed(1234)

```
input_layer = tf.keras.layers.InputLayer(input_shape=(1,))
hidden_layer1 = tf.keras.layers.Dense(units=4, activation='relu')
hidden_layer2 = tf.keras.layers.Dense(units=2, activation='relu')
output_layer= tf.keras.layers.Dense(units=1)
# activation이 없으면 활성화함수 wx+b가 되어있는 것임.
```

```
model = keras.Sequential([
    input_layer,
    hidden_layer1,
    hidden_layer2,
    output_layer
])
```

```
model.compile(#optimizer='adam',
              loss='mse',
              metrics=['accuracy'])
```

```
print(model.fit(x, y))
print(model.predict(x))
print(model.evaluate(x,y))
```

1

```
model = keras.Sequential([
    0번레이어 ← tf.keras.layers.Dense(units=4,input_shape=(1,),activation='relu'),
    1번레이어 ← tf.keras.layers.Dense(units=2, activation='relu'),
    2번레이어 ← tf.keras.layers.Dense(units=1) ])
```

1

input값의 x변수의 갯수에 따라 w값의 갯수가결정됨

x=[1,2,3,4,5,6] 은 x값은 1개이며 w값 1개, b값이 1개여야함.
x=[[1,2],[2,3],[3,4],[4,8],[5,2],[6,4]]
하면 x값은 2개이며 w값도 2개, b값이 1개여야함

x변수1개일때
w1개,b1개필요

x변수2개일때
w2개,b1개필요

x1 (수면시간)	y (스트레스)
1	10
2	98
3	8
4	2
5	3
6	4

x1 (수면시간)	x2 (음식)	y (스트레스)
1	2	10
2	3	98
3	4	8
4	8	2
5	2	3
6	4	4

```
! model.summary()

Model: "sequential_10"
-----
Layer (type)                 Output Shape              Param #
-----
dense_22 (Dense)             (None, 4)                 8
dense_23 (Dense)             (None, 2)                10
dense_24 (Dense)             (None, 1)                 3
-----
Total params: 21
Trainable params: 21
Non-trainable params: 0
-----
```

0번레이어
1번레이어
2번레이어

- ▶ 0번레이어는 tf.keras.layers.Dense(units=4,input_shape=(1,),activation='relu'),
→ input_shape는 1개여서 w값이1개, b값이 1개임,
0번레이어의 출력유닛은 4개임.
- 유닛1개당 wx+b를 계산함. w,b의 2개 파라미터 필요
- 유닛4개*2파라미터=총8개의 파라미터 생성됨
계산된 결과값은 relu함수에 의해 계산결과를 다시 재 조정함.
- ▶ 1번레이어는 tf.keras.layers.Dense(units=2, activation='relu'),
→ 0번레이어의 유닛갯수가 input값이 됨. 0번레이어 유닛수는 4, x변수가 4개라는 뜻
1번레이어 출력유닛은 2개임
- 유닛1개당 w값4개+바이어스1개의 5개의 파라미터 필요
- 유닛2개*5개파라미터=총 10개의 파라미터 생성됨
계산된 결과는 relu함수에 의해 다시 재조정함
- ▶ 2번레이어는 tf.keras.layers.Dense(units=1)])
→ 1번레이어의 유닛갯수가 input값이 됨. 1번레이어의 유닛수는 2, x변수가2개라는뜻
2번레이어의 출력유닛은 1개임
- 유닛1개당 w값2개+바이어스1개 3개의 파라미터 피머
- 유닛1개*3개파라미터=총3개의 파라미터 생성됨
- activation이 없음으로 wx+b값으로만 계산됨

```
model = keras.Sequential([
    tf.keras.layers.Dense(units=4, input_shape=(1,), activation='relu'),
    tf.keras.layers.Dense(units=2, activation='relu'),
    tf.keras.layers.Dense(units=1) ])
```

x_k^T	w^*x+b	w^*x+b	w^*x+b	w^*x+b
1	0.08334005	-0.29660565	0.17884266	-0.56124383
2	0.1666801	-0.5932113	0.35768532	-1.12248766
3	0.25002015	-0.88981695	0.53652798	-1.68373149
4	0.3333602	-1.1864226	0.71537064	-2.24497532
5	0.41670025	-1.48302825	0.8942133	-2.80621915
6	0.5000403	-1.7796339	1.07305596	-3.36746298

$$f = \begin{cases} (x < 0) & f(x) = 0 \\ (x \geq 0) & f(x) = x \end{cases}$$

x_i^T	w^*x+b	w^*x+b	w^*x+b	w^*x+b
1	0.08334005	0	0.17884266	0
2	0.1666801	0	0.35768532	0
3	0.25002015	0	0.53652798	0
4	0.3333602	0	0.71537064	0
5	0.41670025	0	0.8942133	0
6	0.5000403	0	1.07305596	0

```
1 modelvalue=model.weights
2 modelvalue

[<tf.Variable 'dense_19/kernel:0' shape=(1, 4) dtype=float32,
array([[ 0.08334005, -0.29660565,  0.17884266, -0.56124383]],
dtype=float32)>,
<tf.Variable 'dense_19/bias:0' shape=(4,) dtype=float32, numpy=array([0., 0., 0., 0.],
dtype=float32)>,
<tf.Variable 'dense_20/kernel:0' shape=(4, 2) dtype=float32, numpy=
array([[ -0.34927058, -0.7225988 ],
[ 0.2960987 , -0.39196324],
[ -0.32647347, -0.73963904],
[ 0.41159463,  0.43719316]], dtype=float32)>,
<tf.Variable 'dense_20/bias:0' shape=(2,) dtype=float32, numpy=array([0., 0.], dtype=float32)>,
<tf.Variable 'dense_21/kernel:0' shape=(2, 1) dtype=float32, numpy=
array([[ 0.2757995 ],
[ -0.55681044]], dtype=float32)>,
<tf.Variable 'dense_21/bias:0' shape=(1,) dtype=float32, numpy=array([0.00316228], dtype=float32)>]
```

```
1 model.summary()
```

Model: "sequential_9"

Layer (type)	Output Shape	Param #
dense_19 (Dense)	(None, 4)	8
dense_20 (Dense)	(None, 2)	10
dense_21 (Dense)	(None, 1)	3

=====

Total params: 21
Trainable params: 21
Non-trainable params: 0

```
8 intermediate_layer_model = tf.keras.Model(inputs=model.input, outputs=model.layers[0].output)
9 intermediate_output = intermediate_layer_model(x)
10
11 print('\n===== Outputs of Hidden Layer =====')
12 print(intermediate_output)

===== Outputs of Hidden Layer =====
tf.Tensor(
[[[ 0.08334005  0.          0.17884266  0.          ]
  [ 0.16668001  0.          0.35768533  0.          ]
  [ 0.25002015  0.          0.536528      0.          ]
  [ 0.3333602   0.          0.71537066  0.          ]
  [ 0.41670024  0.          0.8942133   0.          ]
  [ 0.5000403   0.          1.073056    0.          ]], shape=(6, 4), dtype=float32)
```

layer0 출력물

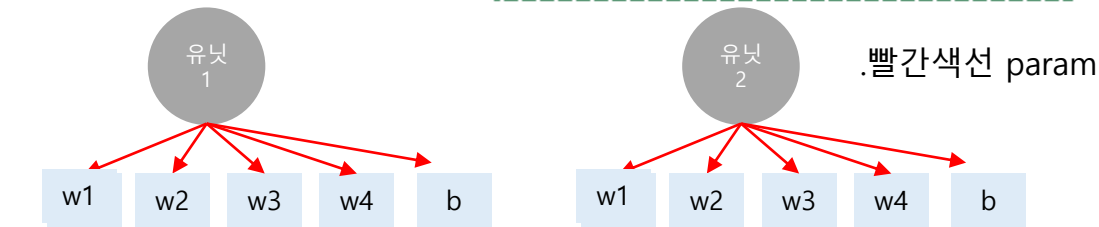
model.summary 에 대한 해석 [1번레이어] 두번째레이어

☐

[input값]은

layer[0]에서의 결과인 유닛4개
에 계산된 결과로서
x1,x2,x3,x4 값임

x1	x2	x3	x4
0.08334	0	0.178843	0
0.16668	0	0.357685	0
0.25002	0	0.536528	0
0.33336	0	0.715371	0
0.4167	0	0.894213	0
0.50004	0	1.073056	0



유닛1	x1의 w1	x2의 w2	x3의 w3	x4의 w4
w	-0.34927058	0.2960987	-0.32647347	0.41159463
b	0			

유닛2	x1의 w1	x2의 w2	x3의 w3	x4의 w4
w	-0.7225988	-0.39196324	-0.73963904	0.43719316
b	0			

x1	x2	x3	x4	유닛1의 wx+b
0.08334	0	0.17884266	0	-0.087495611
0.16668	0	0.35768532	0	-0.174991223
0.25002	0	0.53652798	0	-0.262486834
0.33336	0	0.71537064	0	-0.349982446
0.4167	0	0.8942133	0	-0.437478057
0.50004	0	1.07305596	0	-0.524973668

x1	x2	x3	x4	유닛2의 wx+b
0.08334	0	0.17884266	0	-0.192500433
0.16668	0	0.35768532	0	-0.385000867
0.25002	0	0.53652798	0	-0.5775013
0.33336	0	0.71537064	0	-0.770001734
0.4167	0	0.8942133	0	-0.962502167
0.50004	0	1.07305596	0	-1.155002601

계산된값 ==>	relu함수 수행후
유닛1의 wx+b	유닛2의 wx+b
-0.087495611	-0.192500433
-0.174991223	-0.385000867
-0.262486834	-0.5775013
-0.349982446	-0.770001734
-0.437478057	-0.962502167
-0.524973668	-1.155002601

검정테두리 layer[1] 에서 출력물

```
model = keras.Sequential([
    tf.keras.layers.Dense(units=4,input_shape=(1,),activation='relu'),
    tf.keras.layers.Dense(units=2, activation='relu'),
    tf.keras.layers.Dense(units=1) ])

```

```

1 modelvalue=model.weights
2 modelvalue

[<tf.Variable 'dense_19/kernel:0' shape=(1, 4) dtype=float32, numpy=
array([[ 0.2960987, -0.39196324, -0.32647347, -0.56124383]],
      dtype=float32)>,
<tf.Variable 'dense_19/bias:0' shape=(4,) dtype=float32, numpy=array([0., 0., 0., 0.], dtype=float32)>,
<tf.Variable 'dense_20/kernel:0' shape=(4, 2) dtype=float32, numpy=
array([[-0.34927058, -0.7225988 ],
       [ 0.2960987, -0.39196324],
       [-0.32647347, -0.73963904],
       [ 0.41159463,  0.43719316]], dtype=float32)>,
<tf.Variable 'dense_20/bias:0' shape=(2,) dtype=float32, numpy=array([0., 0.], dtype=float32)>,
<tf.Variable 'dense_21/kernel:0' shape=(2, 1) dtype=float32, numpy=
array([[ 0.2757995 ],
       [-0.55681044]], dtype=float32)>,
<tf.Variable 'dense_21/bias:0' shape=(1,) dtype=float32, numpy=array([0.00316228], dtype=float32)>]

```

유닛1 w값

유닛2 w값

w값 8개

b값2개

```

1 model.summary()

Model: "sequential_9"
-----
Layer (type)                Output Shape              Param #
-----
dense_19 (Dense)             (None, 4)                 8
dense_20 (Dense)             (None, 2)                10
dense_21 (Dense)             (None, 1)                 3
-----
Total params: 21
Trainable params: 21
Non-trainable params: 0

```

param갯수10개

```

8 intermediate_layer_model = tf.keras.Model(inputs=model.input, outputs=model.layers[1].output)
9 intermediate_output = intermediate_layer_model(x)
10
11 print('\n===== Outputs of Hidden Layer =====')
12 print(intermediate_output)

```

layer[1]출력물

```

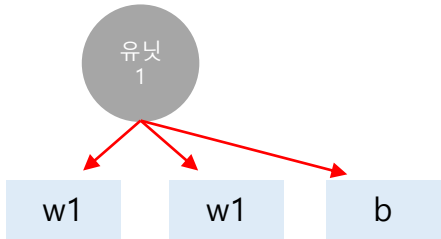
===== Outputs of Hidden Layer =====
tf.Tensor(
[[ 0.  0.]
 [ 0.  0.]
 [ 0.  0.]
 [ 0.  0.]
 [ 0.  0.]
 [ 0.  0.]], shape=(6, 2), dtype=float32)

```

model.summary 에 대한 해석 [2번레이어] 세번째레이어.

[input값]은
layer[1]에서의 결과인 유닛2개
에서 계산된
결과로서 x1,x2 값임

x1	x2
0	0
0	0
0	0
0	0
0	0
0	0



.빨간색선 param

유닛1	x1의 w1	x2의 w2
w	0.2758	-0.55681044
b	0.00316228	

x1	x2	유닛1의 wx+b	relu수행후 결과값
0	0	0.00316228	0.00316228
0	0	0.00316228	0.00316228
0	0	0.00316228	0.00316228
0	0	0.00316228	0.00316228
0	0	0.00316228	0.00316228
0	0	0.00316228	0.00316228

검정테두리 layer[2] 에서 출력물

```
model = keras.Sequential([
    tf.keras.layers.Dense(units=4,input_shape=(1,),activation='relu'),
    tf.keras.layers.Dense(units=2, activation='relu'),
    tf.keras.layers.Dense(units=1) ])
```

```
1 modelvalue=model.weights
2 modelvalue

[<tf.Variable 'dense_19/kernel:0' shape=(1, 4) dtype=float32, numpy=
array([[ 0.08334005, -0.29660565,  0.17884266, -0.56124383]],
      dtype=float32)>,
<tf.Variable 'dense_19/bias:0' shape=(4,) dtype=float32, numpy=array([0., 0., 0., 0.], dtype=float32)>,
<tf.Variable 'dense_20/kernel:0' shape=(4, 2) dtype=float32, numpy=
array([[ -0.34927058, -0.7225988 ],
       [ 0.2960987 , -0.39196324],
       [-0.32647347, -0.73963904],
       [ 0.41159463,  0.43719316]], dtype=float32)>,
<tf.Variable 'dense_20/bias:0' shape=(2,) dtype=float32, numpy=array([0., 0.], dtype=float32)>,
<tf.Variable 'dense_21/kernel:0' shape=(2, 1) dtype=float32, numpy=
array([[ 0.2757995 ],
       [-0.55681044]], dtype=float32)>,
<tf.Variable 'dense_21/bias:0' shape=(1,) dtype=float32, numpy=array([0.00316228], dtype=float32)>]
```

w값 2개

b값1개

```
1 model.summary()

Model: "sequential_9"
-----
Layer (type)                Output Shape         Param #
-----
dense_19 (Dense)             (None, 4)            8
dense_20 (Dense)             (None, 2)           10
dense_21 (Dense)             (None, 1)            3
-----
Total params: 21
Trainable params: 21
Non-trainable params: 0
```

param갯수 3

```
8 intermediate_layer_model = tf.keras.Model(inputs=model.input, outputs=model.layers[2].output)
9 intermediate_output = intermediate_layer_model(x)
10
11 print('\n===== Outputs of Hidden Layer =====')
12 print(intermediate_output)

===== Outputs of Hidden Layer =====
tf.Tensor(
[[0.00316228]
 [0.00316228]
 [0.00316228]
 [0.00316228]
 [0.00316228]
 [0.00316228]], shape=(6, 1), dtype=float32)
```

layer[2]출력물

```
model = keras.Sequential([
    tf.keras.layers.Dense(units=4,input_shape=(1,),activation='relu'),
    tf.keras.layers.Dense(units=2, activation='relu'),
    tf.keras.layers.Dense(units=1) ])
```

(input값의 shape갯수+b)*units개수
(바로위의 uniut갯수가 input이됨으로 바로위units개수+b)*units개수
(바로위의 uniut갯수가 input이됨으로 바로위units개수+b)*units개수

$(1+1)*4=8$
 $(4+1)*2=10$
 $(2+1)*1=3$

문제