In [1]:

**import** pandas **as** pd   
**import** numpy **as** np  
**import** seaborn **as** sns  
**%matplotlib** inline

In [2]:

**import** tensorflow **as** tf

In [3]:

data**=**pd**.**read\_csv(r"E:\Etisalat\_Deep\_learning\session 14 Dec\assigment1\Salary\_Data.csv")

In [4]:

data**.**info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 30 entries, 0 to 29  
Data columns (total 2 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 YearsExperience 30 non-null float64  
 1 Salary 30 non-null float64  
dtypes: float64(2)  
memory usage: 608.0 bytes

In [5]:

sns**.**pairplot(data)

Out[5]:

<seaborn.axisgrid.PairGrid at 0x1ab1d181670>

In [6]:

x**=**data**.**iloc[:,:**-**1]**.**values  
y**=**data**.**iloc[:,**-**1]**.**values

In [7]:

x

Out[7]:

array([[ 1.1],  
 [ 1.3],  
 [ 1.5],  
 [ 2. ],  
 [ 2.2],  
 [ 2.9],  
 [ 3. ],  
 [ 3.2],  
 [ 3.2],  
 [ 3.7],  
 [ 3.9],  
 [ 4. ],  
 [ 4. ],  
 [ 4.1],  
 [ 4.5],  
 [ 4.9],  
 [ 5.1],  
 [ 5.3],  
 [ 5.9],  
 [ 6. ],  
 [ 6.8],  
 [ 7.1],  
 [ 7.9],  
 [ 8.2],  
 [ 8.7],  
 [ 9. ],  
 [ 9.5],  
 [ 9.6],  
 [10.3],  
 [10.5]])

In [8]:

y

Out[8]:

array([ 39343., 46205., 37731., 43525., 39891., 56642., 60150.,  
 54445., 64445., 57189., 63218., 55794., 56957., 57081.,  
 61111., 67938., 66029., 83088., 81363., 93940., 91738.,  
 98273., 101302., 113812., 109431., 105582., 116969., 112635.,  
 122391., 121872.])

In [9]:

**from** sklearn.preprocessing **import** MinMaxScaler

In [10]:

scaler**=**MinMaxScaler()

In [11]:

scaler**.**fit(x)

Out[11]:

MinMaxScaler()

In [12]:

x\_sc**=**scaler**.**transform(x)

In [13]:

x\_sc

Out[13]:

array([[0. ],  
 [0.0212766 ],  
 [0.04255319],  
 [0.09574468],  
 [0.11702128],  
 [0.19148936],  
 [0.20212766],  
 [0.22340426],  
 [0.22340426],  
 [0.27659574],  
 [0.29787234],  
 [0.30851064],  
 [0.30851064],  
 [0.31914894],  
 [0.36170213],  
 [0.40425532],  
 [0.42553191],  
 [0.44680851],  
 [0.5106383 ],  
 [0.5212766 ],  
 [0.60638298],  
 [0.63829787],  
 [0.72340426],  
 [0.75531915],  
 [0.80851064],  
 [0.84042553],  
 [0.89361702],  
 [0.90425532],  
 [0.9787234 ],  
 [1. ]])

In [14]:

**from** sklearn.model\_selection **import** train\_test\_split

In [15]:

xtrain,xtest,ytrain,ytest**=**train\_test\_split(x\_sc,y,test\_size**=**0.3,random\_state**=**123)

In [16]:

xtrain**.**shape

Out[16]:

(21, 1)

In [17]:

**from** tensorflow.keras.models **import** Sequential  
**from** tensorflow.keras.layers **import** Dense

In [18]:

model**=**Sequential()  
model**.**add(Dense(2,activation**=**"relu"))   
*#model.add(Dense(4,activation="relu"))*   
model**.**add(Dense(2,activation**=**"relu"))   
model**.**add(Dense(1))

In [19]:

model**.**compile(optimizer**=**"rmsprop",loss**=**"mse")

In [20]:

model**.**fit(xtrain,ytrain,epochs**=**50,batch\_size**=**2)

Epoch 1/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072717824.0000  
Epoch 2/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072715776.0000  
Epoch 3/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072713728.0000  
Epoch 4/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072712704.0000  
Epoch 5/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072710656.0000  
Epoch 6/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072709632.0000  
Epoch 7/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072708096.0000  
Epoch 8/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072706048.0000  
Epoch 9/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072705024.0000  
Epoch 10/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072702976.0000  
Epoch 11/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072701440.0000  
Epoch 12/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072700416.0000  
Epoch 13/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072699392.0000  
Epoch 14/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072696320.0000  
Epoch 15/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072695296.0000  
Epoch 16/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072694272.0000  
Epoch 17/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072692224.0000  
Epoch 18/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072690688.0000  
Epoch 19/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072689152.0000  
Epoch 20/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072687616.0000  
Epoch 21/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072686080.0000  
Epoch 22/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072685056.0000  
Epoch 23/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072682496.0000  
Epoch 24/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072680960.0000  
Epoch 25/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072679424.0000  
Epoch 26/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072678400.0000  
Epoch 27/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072676352.0000  
Epoch 28/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072674816.0000  
Epoch 29/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072673792.0000  
Epoch 30/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072671744.0000  
Epoch 31/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072669696.0000  
Epoch 32/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072668672.0000  
Epoch 33/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072666624.0000  
Epoch 34/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072665600.0000  
Epoch 35/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072663552.0000  
Epoch 36/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072662528.0000  
Epoch 37/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072660480.0000  
Epoch 38/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072658944.0000  
Epoch 39/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072657920.0000  
Epoch 40/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072656384.0000  
Epoch 41/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072654848.0000  
Epoch 42/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072652800.0000  
Epoch 43/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072651264.0000  
Epoch 44/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072649216.0000  
Epoch 45/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072648704.0000  
Epoch 46/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072646656.0000  
Epoch 47/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072645120.0000  
Epoch 48/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072644096.0000  
Epoch 49/50  
11/11 [==============================] - 0s 1ms/step - loss: 6072642048.0000  
Epoch 50/50  
11/11 [==============================] - ETA: 0s - loss: 11607519232.000 - 0s 1ms/step - loss: 6072640512.0000

Out[20]:

<keras.callbacks.History at 0x1ab1f986610>

In [21]:

model**.**summary()

Model: "sequential"  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Layer (type) Output Shape Param #   
=================================================================  
 dense (Dense) (None, 2) 4   
   
 dense\_1 (Dense) (None, 2) 6   
   
 dense\_2 (Dense) (None, 1) 3   
   
=================================================================  
Total params: 13  
Trainable params: 13  
Non-trainable params: 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [22]:

model**.**history**.**history["loss"]

Out[22]:

[6072717824.0,  
 6072715776.0,  
 6072713728.0,  
 6072712704.0,  
 6072710656.0,  
 6072709632.0,  
 6072708096.0,  
 6072706048.0,  
 6072705024.0,  
 6072702976.0,  
 6072701440.0,  
 6072700416.0,  
 6072699392.0,  
 6072696320.0,  
 6072695296.0,  
 6072694272.0,  
 6072692224.0,  
 6072690688.0,  
 6072689152.0,  
 6072687616.0,  
 6072686080.0,  
 6072685056.0,  
 6072682496.0,  
 6072680960.0,  
 6072679424.0,  
 6072678400.0,  
 6072676352.0,  
 6072674816.0,  
 6072673792.0,  
 6072671744.0,  
 6072669696.0,  
 6072668672.0,  
 6072666624.0,  
 6072665600.0,  
 6072663552.0,  
 6072662528.0,  
 6072660480.0,  
 6072658944.0,  
 6072657920.0,  
 6072656384.0,  
 6072654848.0,  
 6072652800.0,  
 6072651264.0,  
 6072649216.0,  
 6072648704.0,  
 6072646656.0,  
 6072645120.0,  
 6072644096.0,  
 6072642048.0,  
 6072640512.0]

In [23]:

loss\_df **=** pd**.**DataFrame(model**.**history**.**history)

In [24]:

loss\_df**.**plot()

Out[24]:

<AxesSubplot:>

In [25]:

ypred**=**model**.**predict(xtest)

In [26]:

**from** sklearn.metrics **import** explained\_variance\_score

In [27]:

explained\_variance\_score(ytest,ypred)

Out[27]:

0.0

In [28]:

result**=**pd**.**DataFrame(ypred,columns**=**["ypred"])

In [29]:

result["ytest"]**=**ytest

In [30]:

result["error"]**=**result["ytest"]**-**result["ypred"]

In [31]:

result**.**head()

Out[31]:

|  | **ypred** | **ytest** | **error** |
| --- | --- | --- | --- |
| **0** | 0.54066 | 54445.0 | 54444.45934 |
| **1** | 0.54066 | 121872.0 | 121871.45934 |
| **2** | 0.54066 | 56642.0 | 56641.45934 |
| **3** | 0.54066 | 116969.0 | 116968.45934 |
| **4** | 0.54066 | 64445.0 | 64444.45934 |

In [34]:

tf**.**keras**.**models**.**save\_model(model,"test\_model.h5")

In [ ]:

In [ ]: