Practical 2: Performing regression using feed forward neural network

```
import numpy as np
```

```
np.random.seed(42)
```

```
X = np.random.rand(1000,10)
```

```
y = np.random.rand(1000)
```

```
X.shape
```

```
(1000, 10)
```

```
y.shape
```

```
(1000,)
```

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

```
xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size = 0.25,random_state = 1)
```

```
xtrain.shape
```

```
(750, 10)
```

```
xtest.shape
```

```
(250, 10)
```

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

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

```
model = Sequential()
```

```
model.add(Dense(12,activation = 'relu',input_dim = 10))
model.add(Dense(8,activation = 'relu'))
model.add(Dense(1,activation = 'linear'))
```

```
C:\Users\CompLab14\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass an
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

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

```
model.fit(xtrain,ytrain,epochs = 400, batch_size = 50)
```

```
Epoch 1/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - loss: 0.9166 - mae: 0.8985
Epoch 2/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.4963 - mae: 0.6330
Epoch 3/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 899us/step - loss: 0.2878 - mae: 0.4581
Epoch 4/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 910us/step - loss: 0.1941 - mae: 0.3633
Epoch 5/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 964us/step - loss: 0.1350 - mae: 0.2960
Epoch 6/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 875us/step - loss: 0.1074 - mae: 0.2709
Epoch 7/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 910us/step - loss: 0.0984 - mae: 0.2620
Epoch 8/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 875us/step - loss: 0.0964 - mae: 0.2603
Epoch 9/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 900us/step - loss: 0.0935 - mae: 0.2589
Epoch 10/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 889us/step - loss: 0.0914 - mae: 0.2541
Epoch 11/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 869us/step - loss: 0.0893 - mae: 0.2517
```

```
Epoch 12/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 812us/step - loss: 0.0927 - mae: 0.2564
Epoch 13/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 840us/step - loss: 0.0911 - mae: 0.2541
Epoch 14/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 797us/step - loss: 0.0880 - mae: 0.2469
Epoch 15/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 843us/step - loss: 0.0928 - mae: 0.2561
Epoch 16/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 855us/step - loss: 0.0971 - mae: 0.2616
Epoch 17/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 873us/step - loss: 0.0867 - mae: 0.2465
Epoch 18/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 794us/step - loss: 0.0909 - mae: 0.2558
Epoch 19/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 876us/step - loss: 0.0906 - mae: 0.2562
Epoch 20/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 789us/step - loss: 0.0852 - mae: 0.2489
Epoch 21/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 894us/step - loss: 0.0881 - mae: 0.2490
Epoch 22/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 828us/step - loss: 0.0843 - mae: 0.2437
Epoch 23/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 903us/step - loss: 0.0855 - mae: 0.2478
Epoch 24/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 918us/step - loss: 0.0924 - mae: 0.2574
Epoch 25/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 891us/step - loss: 0.0887 - mae: 0.2502
Epoch 26/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 896us/step - loss: 0.0863 - mae: 0.2504
Epoch 27/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 856us/step - loss: 0.0848 - mae: 0.2448
Epoch 28/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 857us/step - loss: 0.0873 - mae: 0.2463
Epoch 29/400
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 864us/step - loss: 0.0910 - mae: 0.2552
```

```
predictions = model.predict(xtest)
```

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
```

```
from sklearn.metrics import r2_score, mean_absolute_error
```

```
r2_score(ytest,predictions)
```

```
-0.10996355407798108
```

```
mean_absolute_error(ytest,predictions)
```

```
0.26674690641510374
```

Start coding or generate with AI.