deep_learning

January 15, 2023

0.1 Importation of librairies

[29]:
```
import pandas as pd
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

0.2 Data

[2]:
```
concrete_data = pd.read_csv('https://s3-api.us-geo.objectstorage.softlayer.net/
↪cf-courses-data/CognitiveClass/DL0101EN/labs/data/concrete_data.csv')
concrete_data.head()
```

[2]:
```
   Cement  Blast Furnace Slag  Fly Ash  Water  Superplasticizer  \
0   540.0                 0.0      0.0  162.0               2.5
1   540.0                 0.0      0.0  162.0               2.5
2   332.5               142.5      0.0  228.0               0.0
3   332.5               142.5      0.0  228.0               0.0
4   198.6               132.4      0.0  192.0               0.0

   Coarse Aggregate  Fine Aggregate  Age  Strength
0            1040.0           676.0   28     79.99
1            1055.0           676.0   28     61.89
2             932.0           594.0  270     40.27
3             932.0           594.0  365     41.05
4             978.4           825.5  360     44.30
```

0.3 Dimensions

[3]:
```
concrete_data.shape
```

[3]: (1030, 9)

[4]:
```
concrete_data.describe()
```

[4]:
```
            Cement  Blast Furnace Slag      Fly Ash        Water  \
count  1030.000000         1030.000000  1030.000000  1030.000000
mean    281.167864           73.895825    54.188350   181.567282
std     104.506364           86.279342    63.997004    21.354219
min     102.000000            0.000000     0.000000   121.800000
25%     192.375000            0.000000     0.000000   164.900000
50%     272.900000           22.000000     0.000000   185.000000
75%     350.000000          142.950000   118.300000   192.000000
max     540.000000          359.400000   200.100000   247.000000

       Superplasticizer  Coarse Aggregate  Fine Aggregate          Age  \
count       1030.000000       1030.000000     1030.000000  1030.000000
mean           6.204660        972.918932      773.580485    45.662136
std            5.973841         77.753954       80.175980    63.169912
min            0.000000        801.000000      594.000000     1.000000
25%            0.000000        932.000000      730.950000     7.000000
50%            6.400000        968.000000      779.500000    28.000000
75%           10.200000       1029.400000      824.000000    56.000000
max           32.200000       1145.000000      992.600000   365.000000

          Strength
count  1030.000000
mean     35.817961
std      16.705742
min       2.330000
25%      23.710000
50%      34.445000
75%      46.135000
max      82.600000
```

0.4 Definition of predictor and label

[5]:
```
concrete_data_columns = concrete_data.columns

predictors = concrete_data[concrete_data_columns[concrete_data_columns !=␣
↪'Strength']] # all columns except Strength
target = concrete_data['Strength'] # Strength column
```

[9]:
```
# Consulatation de target
target.head()
```

[9]:
```
0    79.99
1    61.89
2    40.27
3    41.05
```

```
4    44.30
Name: Strength, dtype: float64
```

[8]:

```
# Consultation of predictor
predictors.head()
```

[8]:

```
   Cement  Blast Furnace Slag  Fly Ash  Water  Superplasticizer  \
0   540.0                 0.0      0.0  162.0               2.5
1   540.0                 0.0      0.0  162.0               2.5
2   332.5               142.5      0.0  228.0               0.0
3   332.5               142.5      0.0  228.0               0.0
4   198.6               132.4      0.0  192.0               0.0

   Coarse Aggregate  Fine Aggregate  Age
0            1040.0           676.0   28
1            1055.0           676.0   28
2             932.0           594.0  270
3             932.0           594.0  365
4             978.4           825.5  360
```

0.5 Devision of Data Set

[17]:

```
x_train, x_test, y_train, y_test = train_test_split(
...     predictors, target, test_size=0.3, random_state=4
... )
```

[20]:

```
# Analyse of shape
x_test.shape
```

[20]: (309, 8)

[19]:

```
x_train.shape
```

[19]: (721, 8)

0.6 Model

[25]:

```
# define regression model
def regression_model():
    # create model
    model = Sequential()
    model.add(Dense(50, activation='relu', input_shape=(8,)))
    model.add(Dense(10, activation='relu'))
    model.add(Dense(1))

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

```
    return model
```

[26]:
```
# build the model
model = regression_model()
```

```
WARNING:tensorflow:From /home/jupyterlab/conda/envs/python/lib/python3.7/site-
packages/keras/backend/tensorflow_backend.py:508: The name tf.placeholder is
deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /home/jupyterlab/conda/envs/python/lib/python3.7/site-
packages/keras/backend/tensorflow_backend.py:3837: The name tf.random_uniform is
deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /home/jupyterlab/conda/envs/python/lib/python3.7/site-
packages/keras/optimizers.py:757: The name tf.train.Optimizer is deprecated.
Please use tf.compat.v1.train.Optimizer instead.
```

[34]:
```
# fit the model
model.fit(predictors, target, epochs=50)
```

```
Epoch 1/50
1030/1030 [==============================] - 0s 102us/step - loss: 70.8455
Epoch 2/50
1030/1030 [==============================] - 0s 95us/step - loss: 62.5685
Epoch 3/50
1030/1030 [==============================] - 0s 98us/step - loss: 64.1333
Epoch 4/50
1030/1030 [==============================] - 0s 102us/step - loss: 60.4431
Epoch 5/50
1030/1030 [==============================] - 0s 96us/step - loss: 58.2598
Epoch 6/50
1030/1030 [==============================] - 0s 109us/step - loss: 55.9336
Epoch 7/50
1030/1030 [==============================] - 0s 94us/step - loss: 56.2793
Epoch 8/50
1030/1030 [==============================] - 0s 97us/step - loss: 55.9666
Epoch 9/50
1030/1030 [==============================] - 0s 100us/step - loss: 60.3578
Epoch 10/50
1030/1030 [==============================] - 0s 96us/step - loss: 65.7023
Epoch 11/50
1030/1030 [==============================] - 0s 99us/step - loss: 59.5190
Epoch 12/50
1030/1030 [==============================] - 0s 84us/step - loss: 66.9006
Epoch 13/50
1030/1030 [==============================] - 0s 98us/step - loss: 53.5756
```

```
Epoch 14/50
1030/1030 [==============================] - 0s 97us/step - loss: 52.6707
Epoch 15/50
1030/1030 [==============================] - 0s 95us/step - loss: 54.5536
Epoch 16/50
1030/1030 [==============================] - 0s 100us/step - loss: 53.0569
Epoch 17/50
1030/1030 [==============================] - 0s 96us/step - loss: 52.2070
Epoch 18/50
1030/1030 [==============================] - 0s 100us/step - loss: 53.5091
Epoch 19/50
1030/1030 [==============================] - 0s 108us/step - loss: 51.8190
Epoch 20/50
1030/1030 [==============================] - 0s 98us/step - loss: 52.8835
Epoch 21/50
1030/1030 [==============================] - 0s 96us/step - loss: 50.6676
Epoch 22/50
1030/1030 [==============================] - 0s 84us/step - loss: 55.5970
Epoch 23/50
1030/1030 [==============================] - 0s 87us/step - loss: 51.8259
Epoch 24/50
1030/1030 [==============================] - 0s 85us/step - loss: 54.8239
Epoch 25/50
1030/1030 [==============================] - 0s 95us/step - loss: 50.1635
Epoch 26/50
1030/1030 [==============================] - 0s 95us/step - loss: 52.9023
Epoch 27/50
1030/1030 [==============================] - 0s 98us/step - loss: 53.0006
Epoch 28/50
1030/1030 [==============================] - 0s 111us/step - loss: 55.6028
Epoch 29/50
1030/1030 [==============================] - 0s 97us/step - loss: 61.1550
Epoch 30/50
1030/1030 [==============================] - 0s 94us/step - loss: 64.4199
Epoch 31/50
1030/1030 [==============================] - 0s 96us/step - loss: 48.8291
Epoch 32/50
1030/1030 [==============================] - 0s 98us/step - loss: 48.6748
Epoch 33/50
1030/1030 [==============================] - 0s 98us/step - loss: 48.2662
Epoch 34/50
1030/1030 [==============================] - 0s 111us/step - loss: 50.3502
Epoch 35/50
1030/1030 [==============================] - 0s 96us/step - loss: 47.4002
Epoch 36/50
1030/1030 [==============================] - 0s 101us/step - loss: 49.8851
Epoch 37/50
1030/1030 [==============================] - 0s 110us/step - loss: 49.1929
```

```
Epoch 38/50
1030/1030 [==============================] - 0s 99us/step - loss: 51.8134
Epoch 39/50
1030/1030 [==============================] - 0s 98us/step - loss: 53.9920
Epoch 40/50
1030/1030 [==============================] - 0s 95us/step - loss: 48.8981
Epoch 41/50
1030/1030 [==============================] - 0s 98us/step - loss: 45.7841
Epoch 42/50
1030/1030 [==============================] - 0s 99us/step - loss: 46.8631
Epoch 43/50
1030/1030 [==============================] - 0s 99us/step - loss: 48.2326
Epoch 44/50
1030/1030 [==============================] - 0s 96us/step - loss: 50.6663
Epoch 45/50
1030/1030 [==============================] - 0s 105us/step - loss: 49.8427
Epoch 46/50
1030/1030 [==============================] - 0s 93us/step - loss: 48.4321
Epoch 47/50
1030/1030 [==============================] - 0s 98us/step - loss: 49.6893
Epoch 48/50
1030/1030 [==============================] - 0s 96us/step - loss: 46.0370
Epoch 49/50
1030/1030 [==============================] - 0s 107us/step - loss: 54.7251
Epoch 50/50
1030/1030 [==============================] - 0s 96us/step - loss: 53.2548
```

[34]:
```
<keras.callbacks.History at 0x7ff24c1e3c10>
```

0.7 Evaluation

[35]:
```
y_predic = model.predict(x_test)
mean_squared_error(y_test, y_predic)
```

[35]:
```
54.14465316875257
```

0.8 Repeatition

[36]:
```
# fit the model
model.fit(predictors, target, validation_split=0.3, epochs=50, verbose=2)
```

```
Train on 721 samples, validate on 309 samples
Epoch 1/50
 - 0s - loss: 57.0391 - val_loss: 47.8262
Epoch 2/50
 - 0s - loss: 47.7147 - val_loss: 65.9428
Epoch 3/50
```

```
 - 0s - loss: 56.2756 - val_loss: 38.9488
Epoch 4/50
 - 0s - loss: 44.4637 - val_loss: 41.9539
Epoch 5/50
 - 0s - loss: 46.8669 - val_loss: 64.9146
Epoch 6/50
 - 0s - loss: 44.6839 - val_loss: 44.0102
Epoch 7/50
 - 0s - loss: 44.6251 - val_loss: 91.2380
Epoch 8/50
 - 0s - loss: 46.1912 - val_loss: 52.4451
Epoch 9/50
 - 0s - loss: 47.0320 - val_loss: 41.3937
Epoch 10/50
 - 0s - loss: 44.2686 - val_loss: 58.8403
Epoch 11/50
 - 0s - loss: 43.1044 - val_loss: 54.0395
Epoch 12/50
 - 0s - loss: 45.4103 - val_loss: 49.5595
Epoch 13/50
 - 0s - loss: 43.5166 - val_loss: 67.3776
Epoch 14/50
 - 0s - loss: 43.0266 - val_loss: 61.3177
Epoch 15/50
 - 0s - loss: 41.7098 - val_loss: 56.7682
Epoch 16/50
 - 0s - loss: 42.1425 - val_loss: 75.6810
Epoch 17/50
 - 0s - loss: 43.9215 - val_loss: 75.6160
Epoch 18/50
 - 0s - loss: 41.5984 - val_loss: 49.4442
Epoch 19/50
 - 0s - loss: 44.2283 - val_loss: 98.0946
Epoch 20/50
 - 0s - loss: 45.0367 - val_loss: 49.9688
Epoch 21/50
 - 0s - loss: 43.5622 - val_loss: 67.8486
Epoch 22/50
 - 0s - loss: 43.3263 - val_loss: 60.3424
Epoch 23/50
 - 0s - loss: 47.7705 - val_loss: 52.8211
Epoch 24/50
 - 0s - loss: 49.8652 - val_loss: 83.6179
Epoch 25/50
 - 0s - loss: 42.7449 - val_loss: 76.2814
Epoch 26/50
 - 0s - loss: 42.1140 - val_loss: 53.7191
Epoch 27/50
```

```
 - 0s - loss: 42.1954 - val_loss: 59.0318
Epoch 28/50
 - 0s - loss: 43.0193 - val_loss: 68.6865
Epoch 29/50
 - 0s - loss: 42.6992 - val_loss: 82.3516
Epoch 30/50
 - 0s - loss: 43.6689 - val_loss: 65.4886
Epoch 31/50
 - 0s - loss: 38.9841 - val_loss: 59.4441
Epoch 32/50
 - 0s - loss: 45.6201 - val_loss: 123.5114
Epoch 33/50
 - 0s - loss: 40.1182 - val_loss: 60.6125
Epoch 34/50
 - 0s - loss: 39.5325 - val_loss: 62.6173
Epoch 35/50
 - 0s - loss: 41.8078 - val_loss: 78.8612
Epoch 36/50
 - 0s - loss: 39.5319 - val_loss: 54.5449
Epoch 37/50
 - 0s - loss: 44.6591 - val_loss: 62.6005
Epoch 38/50
 - 0s - loss: 38.8295 - val_loss: 56.1954
Epoch 39/50
 - 0s - loss: 40.6127 - val_loss: 86.7789
Epoch 40/50
 - 0s - loss: 38.5612 - val_loss: 52.5475
Epoch 41/50
 - 0s - loss: 45.2425 - val_loss: 81.6904
Epoch 42/50
 - 0s - loss: 40.1057 - val_loss: 144.8910
Epoch 43/50
 - 0s - loss: 43.8685 - val_loss: 82.8043
Epoch 44/50
 - 0s - loss: 41.1734 - val_loss: 57.1300
Epoch 45/50
 - 0s - loss: 39.1824 - val_loss: 83.7774
Epoch 46/50
 - 0s - loss: 42.4778 - val_loss: 60.5014
Epoch 47/50
 - 0s - loss: 49.2432 - val_loss: 143.2435
Epoch 48/50
 - 0s - loss: 42.1590 - val_loss: 120.7347
Epoch 49/50
 - 0s - loss: 41.7081 - val_loss: 96.1223
Epoch 50/50
 - 0s - loss: 38.6394 - val_loss: 91.2879
```

[36]: <keras.callbacks.History at 0x7ff24c1e3d90>

[ ]: