deep_learning

January 15, 2023

0.1 Importation of librairies

[44]:
```
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
from sklearn.preprocessing import StandardScaler
```

0.2 Data

[45]:
```
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()
```

[45]:
```
   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

[46]:
```
concrete_data.shape
```

[46]:
```
(1030, 9)
```

[47]:
```
concrete_data.describe()
```

[47]:
```
            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

[48]:
```
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
```

[49]:
```
# Consulatation de target
target.head()
```

[49]:
```
0    79.99
1    61.89
2    40.27
3    41.05
```

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

[50]:

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

[50]:

```
   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
```

[51]:

```
scaler = StandardScaler()
predictors = scaler.fit_transform(predictors)
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-
packages/sklearn/preprocessing/data.py:625: DataConversionWarning: Data with
input dtype int64, float64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/home/jupyterlab/conda/envs/python/lib/python3.7/site-
packages/sklearn/base.py:462: DataConversionWarning: Data with input dtype
int64, float64 were all converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
```

0.5 Devision of Data Set

[52]:

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

[53]:

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

[53]: (309, 8)

[54]:

```
x_train.shape
```

[54]: (721, 8)

0.6 Model

[55]:
```python
# 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
```

[56]:
```python
# build the model
model = regression_model()
```

[57]:
```python
# fit the model
model.fit(predictors, target, epochs=50)
```

```
Epoch 1/50
1030/1030 [==============================] - 0s 339us/step - loss: 1537.5973
Epoch 2/50
1030/1030 [==============================] - 0s 132us/step - loss: 1466.6248
Epoch 3/50
1030/1030 [==============================] - 0s 119us/step - loss: 1367.4639
Epoch 4/50
1030/1030 [==============================] - 0s 116us/step - loss: 1222.2084
Epoch 5/50
1030/1030 [==============================] - 0s 113us/step - loss: 1025.9937
Epoch 6/50
1030/1030 [==============================] - 0s 106us/step - loss: 785.3938
Epoch 7/50
1030/1030 [==============================] - 0s 106us/step - loss: 540.7995
Epoch 8/50
1030/1030 [==============================] - 0s 102us/step - loss: 346.7675
Epoch 9/50
1030/1030 [==============================] - 0s 110us/step - loss: 241.2475
Epoch 10/50
1030/1030 [==============================] - 0s 98us/step - loss: 202.7361
Epoch 11/50
1030/1030 [==============================] - 0s 100us/step - loss: 188.9055
Epoch 12/50
1030/1030 [==============================] - 0s 108us/step - loss: 182.3313
Epoch 13/50
1030/1030 [==============================] - 0s 100us/step - loss: 176.0290
Epoch 14/50
1030/1030 [==============================] - 0s 98us/step - loss: 171.4552
```

```
Epoch 15/50
1030/1030 [==============================] - 0s 118us/step - loss: 167.1032
Epoch 16/50
1030/1030 [==============================] - 0s 112us/step - loss: 163.1587
Epoch 17/50
1030/1030 [==============================] - 0s 99us/step - loss: 159.8215
Epoch 18/50
1030/1030 [==============================] - 0s 111us/step - loss: 156.4842
Epoch 19/50
1030/1030 [==============================] - 0s 98us/step - loss: 153.6855
Epoch 20/50
1030/1030 [==============================] - 0s 99us/step - loss: 151.2042
Epoch 21/50
1030/1030 [==============================] - 0s 101us/step - loss: 148.8558
Epoch 22/50
1030/1030 [==============================] - 0s 112us/step - loss: 146.3459
Epoch 23/50
1030/1030 [==============================] - 0s 103us/step - loss: 144.4762
Epoch 24/50
1030/1030 [==============================] - 0s 112us/step - loss: 142.2153
Epoch 25/50
1030/1030 [==============================] - 0s 101us/step - loss: 140.5851
Epoch 26/50
1030/1030 [==============================] - 0s 110us/step - loss: 138.6236
Epoch 27/50
1030/1030 [==============================] - 0s 99us/step - loss: 136.7394
Epoch 28/50
1030/1030 [==============================] - 0s 112us/step - loss: 134.9908
Epoch 29/50
1030/1030 [==============================] - 0s 101us/step - loss: 132.7900
Epoch 30/50
1030/1030 [==============================] - 0s 96us/step - loss: 131.0006
Epoch 31/50
1030/1030 [==============================] - 0s 110us/step - loss: 129.0728
Epoch 32/50
1030/1030 [==============================] - 0s 98us/step - loss: 127.4119
Epoch 33/50
1030/1030 [==============================] - 0s 99us/step - loss: 125.4132
Epoch 34/50
1030/1030 [==============================] - 0s 99us/step - loss: 123.3777
Epoch 35/50
1030/1030 [==============================] - 0s 107us/step - loss: 121.6564
Epoch 36/50
1030/1030 [==============================] - 0s 100us/step - loss: 119.7833
Epoch 37/50
1030/1030 [==============================] - 0s 112us/step - loss: 117.8392
Epoch 38/50
1030/1030 [==============================] - 0s 84us/step - loss: 116.0075
```

```
Epoch 39/50
1030/1030 [==============================] - 0s 110us/step - loss: 114.2936
Epoch 40/50
1030/1030 [==============================] - 0s 104us/step - loss: 112.9092
Epoch 41/50
1030/1030 [==============================] - 0s 109us/step - loss: 111.0420
Epoch 42/50
1030/1030 [==============================] - 0s 101us/step - loss: 109.4129
Epoch 43/50
1030/1030 [==============================] - 0s 113us/step - loss: 107.6769
Epoch 44/50
1030/1030 [==============================] - 0s 114us/step - loss: 106.1748
Epoch 45/50
1030/1030 [==============================] - 0s 99us/step - loss: 104.3573
Epoch 46/50
1030/1030 [==============================] - 0s 111us/step - loss: 102.8363
Epoch 47/50
1030/1030 [==============================] - 0s 97us/step - loss: 101.2352
Epoch 48/50
1030/1030 [==============================] - 0s 101us/step - loss: 99.3394
Epoch 49/50
1030/1030 [==============================] - 0s 109us/step - loss: 97.5534
Epoch 50/50
1030/1030 [==============================] - 0s 102us/step - loss: 95.7881
```

[57]: <keras.callbacks.History at 0x7ff2541858d0>

0.7 Evaluation

[58]:
```
y_predic = model.predict(x_test)
mean_squared_error(y_test, y_predic)
```

[58]: 100.98182985639797

0.8 Repeatition

[59]:
```
# 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: 106.1365 - val_loss: 64.5056
Epoch 2/50
 - 0s - loss: 103.7901 - val_loss: 66.4737
Epoch 3/50
 - 0s - loss: 101.7139 - val_loss: 66.1245
Epoch 4/50
```

```
 - 0s - loss: 99.4354 - val_loss: 67.1270
Epoch 5/50
 - 0s - loss: 97.8777 - val_loss: 67.0554
Epoch 6/50
 - 0s - loss: 95.7436 - val_loss: 67.2301
Epoch 7/50
 - 0s - loss: 93.9668 - val_loss: 66.7415
Epoch 8/50
 - 0s - loss: 92.4888 - val_loss: 67.2635
Epoch 9/50
 - 0s - loss: 90.9773 - val_loss: 68.0497
Epoch 10/50
 - 0s - loss: 89.0871 - val_loss: 67.4271
Epoch 11/50
 - 0s - loss: 87.4319 - val_loss: 66.8488
Epoch 12/50
 - 0s - loss: 85.9487 - val_loss: 67.2577
Epoch 13/50
 - 0s - loss: 84.2010 - val_loss: 67.7886
Epoch 14/50
 - 0s - loss: 82.5356 - val_loss: 68.3232
Epoch 15/50
 - 0s - loss: 81.1661 - val_loss: 67.5488
Epoch 16/50
 - 0s - loss: 80.3900 - val_loss: 69.9430
Epoch 17/50
 - 0s - loss: 79.2070 - val_loss: 68.7976
Epoch 18/50
 - 0s - loss: 77.6879 - val_loss: 68.5199
Epoch 19/50
 - 0s - loss: 76.0210 - val_loss: 69.7992
Epoch 20/50
 - 0s - loss: 74.5249 - val_loss: 69.1899
Epoch 21/50
 - 0s - loss: 73.0776 - val_loss: 69.2975
Epoch 22/50
 - 0s - loss: 71.7265 - val_loss: 70.1329
Epoch 23/50
 - 0s - loss: 70.4977 - val_loss: 70.3100
Epoch 24/50
 - 0s - loss: 69.4264 - val_loss: 71.8569
Epoch 25/50
 - 0s - loss: 68.0969 - val_loss: 70.0096
Epoch 26/50
 - 0s - loss: 66.5636 - val_loss: 70.6091
Epoch 27/50
 - 0s - loss: 65.1601 - val_loss: 68.9008
Epoch 28/50
```

```
 - 0s - loss: 63.9232 - val_loss: 70.3282
Epoch 29/50
 - 0s - loss: 62.5733 - val_loss: 70.1194
Epoch 30/50
 - 0s - loss: 61.5831 - val_loss: 71.1225
Epoch 31/50
 - 0s - loss: 60.3249 - val_loss: 70.4950
Epoch 32/50
 - 0s - loss: 59.1295 - val_loss: 69.9139
Epoch 33/50
 - 0s - loss: 58.6433 - val_loss: 71.3929
Epoch 34/50
 - 0s - loss: 57.0716 - val_loss: 68.6171
Epoch 35/50
 - 0s - loss: 56.0411 - val_loss: 70.9739
Epoch 36/50
 - 0s - loss: 55.7961 - val_loss: 68.6434
Epoch 37/50
 - 0s - loss: 54.8143 - val_loss: 71.7680
Epoch 38/50
 - 0s - loss: 53.8137 - val_loss: 70.2278
Epoch 39/50
 - 0s - loss: 52.9038 - val_loss: 72.1585
Epoch 40/50
 - 0s - loss: 52.4683 - val_loss: 71.7408
Epoch 41/50
 - 0s - loss: 51.4136 - val_loss: 71.5818
Epoch 42/50
 - 0s - loss: 50.6830 - val_loss: 70.9935
Epoch 43/50
 - 0s - loss: 50.0209 - val_loss: 71.1513
Epoch 44/50
 - 0s - loss: 49.4846 - val_loss: 70.7776
Epoch 45/50
 - 0s - loss: 48.9953 - val_loss: 71.4080
Epoch 46/50
 - 0s - loss: 48.4130 - val_loss: 70.6967
Epoch 47/50
 - 0s - loss: 47.8540 - val_loss: 72.3293
Epoch 48/50
 - 0s - loss: 47.2775 - val_loss: 71.4333
Epoch 49/50
 - 0s - loss: 46.6515 - val_loss: 72.3595
Epoch 50/50
 - 0s - loss: 45.9061 - val_loss: 72.3329
```

```
[59]: <keras.callbacks.History at 0x7ff24c1ba550>
```

[ ]:

```

```

[ ]:

```

```