NO2_RNN2

November 30, 2017

In [1]:
```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt

import tensorflow as tf
from keras.backend.tensorflow_backend import set_session
config = tf.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.3
set_session(tf.Session(config=config))
```

```
Using TensorFlow backend.
```

In [2]:
```
os.listdir('data')
```

Out[2]:
```
['AllNO2_QH.csv',
 'AllPM_QH.csv',
 'Env_QH.csv',
 'GradientTemp_15minDataSet.csv',
 'micro_sud3.pkl',
 'micro_sud3_normalized.pkl',
 'Patm_15minDataSet.csv',
 'pickles']
```

In [3]:
```
df = pd.read_pickle('data/micro_sud3_normalized.pkl')
df = df.reindex(np.random.permutation(df.index))
df = df.reset_index()

def split_dataframe(dataframe, percent):
    nb_rows = int(np.floor(percent * len(dataframe)))
    return dataframe[:nb_rows], dataframe[nb_rows:]

def dataframe_to_xy(df, look_back):
    i = look_back
    while True:
        sequence = df.iloc[i - look_back:i]
        yield np.array(sequence[['NO2_61FD', 'NO2_61F0', 'NO2_61EF', 'temp', 'rh',\
```

```
                                        'tgrad', 'pressure', 'pluvio']]).reshape(look_back, 1
            i += 1
            if i == len(df):
                i = look_back

    def dataframe_to_xy_test(df, look_back):
        X_test, y_test = [], []
        i = look_back
        while i < len(df):
            sequence = df.iloc[i - look_back:i]
            X_test.append(np.array(sequence[['NO2_61FD', 'NO2_61F0', 'NO2_61EF', 'temp', '
                                        'tgrad', 'pressure', 'pluvio']]).reshape(look_back, 1
            y_test.append(np.array(df.iloc[i]['NO2_ref']))
            i += 1
            if i == len(df):
                break

        return np.array(X_test), np.array(y_test)


    df_test, df_train = split_dataframe(df, 0.5)
    df_valid, df_test = split_dataframe(df_test, 0.5)

    X_train = dataframe_to_xy(df_train, 10)
    X_valid = dataframe_to_xy(df_valid, 10)
    X_test, y_test = dataframe_to_xy_test(df_test, 10)
```

In [4]:
```
def dataframe_to_xy_no_generator(df, look_back):
    X, y = [], []
    i = look_back
    while i < len(df):
        sequence = df.iloc[i - look_back:i]
        X.append(np.array(sequence[['NO2_61FD', 'NO2_61F0', 'NO2_61EF', 'temp', 'rh',\
                                    'tgrad', 'pressure', 'pluvio']]).reshape(look_back, 1
        y.append(np.array(df.iloc[i]['NO2_ref']))
        i += 1
        if i == len(df):
            break
    X = np.array(X)
    y = np.array(y)
    X = X.reshape((X.shape[0], X.shape[1], X.shape[3]))
    return X, y
```

In [5]:
```
X_train, y_train = dataframe_to_xy_no_generator(df_train, 24)
X_valid, y_valid = dataframe_to_xy_no_generator(df_valid, 24)
X_test, y_test = dataframe_to_xy_no_generator(df_test, 24)
```

In [6]:
```
from keras.layers import SimpleRNN, Dense, LSTM, GRU
from keras.models import Sequential
```

```
        from keras.callbacks import EarlyStopping

        def simple_rnn_model(nb_units, input_dim, loss='mean_squared_error', optimizer='adam'):
            model = Sequential()
            model.add(SimpleRNN(nb_units, input_shape=input_dim))
            model.add(Dense(nb_units, activation='relu'))
            model.add(Dense(1, kernel_initializer='normal'))
            model.compile(loss=loss, optimizer=optimizer)
            model.summary()
            return model

        def lstm_model(nb_units, input_dim, loss='mean_squared_error', optimizer='adam'):
            model = Sequential()
            model.add(LSTM(nb_units, input_shape=input_dim))
            model.add(Dense(nb_units, activation='relu'))
            model.add(Dense(1, kernel_initializer='normal'))
            model.compile(loss=loss, optimizer=optimizer)
            model.summary()
            return model

        def gru_model(nb_units, input_dim, loss='mean_squared_error', optimizer='adam'):
            model = Sequential()
            model.add(GRU(nb_units, input_shape=input_dim))
            model.add(Dense(nb_units, activation='relu'))
            model.add(Dense(1, kernel_initializer='normal'))
            model.compile(loss=loss, optimizer=optimizer)
            model.summary()
            return model

In [7]: model = simple_rnn_model(32, X_train.shape[1:])

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_1 (SimpleRNN)     (None, 32)                1312
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33
=================================================================
Total params: 2,401
Trainable params: 2,401
Non-trainable params: 0
_________________________________________________________________


In [8]: early_stopping = EarlyStopping(monitor='val_loss', verbose=1, mode='auto', patience=10)
        history = model.fit(X_train, y_train, batch_size=32, epochs=5000, validation_data=(X_va
```

```
Train on 1103 samples, validate on 539 samples
Epoch 1/5000
1103/1103 [==============================] - 1s 528us/step - loss: 2586.4411 - val_loss: 2496.8
Epoch 2/5000
1103/1103 [==============================] - 0s 145us/step - loss: 2499.1545 - val_loss: 2303.5
Epoch 3/5000
1103/1103 [==============================] - 0s 142us/step - loss: 2096.0381 - val_loss: 1652.7
Epoch 4/5000
1103/1103 [==============================] - 0s 138us/step - loss: 1456.1172 - val_loss: 1186.2
Epoch 5/5000
1103/1103 [==============================] - 0s 164us/step - loss: 1180.1366 - val_loss: 1078.2
Epoch 6/5000
1103/1103 [==============================] - 0s 155us/step - loss: 1131.2053 - val_loss: 1058.4
Epoch 7/5000
1103/1103 [==============================] - 0s 156us/step - loss: 1103.8334 - val_loss: 1034.2
Epoch 8/5000
1103/1103 [==============================] - 0s 134us/step - loss: 1075.7789 - val_loss: 1010.6
Epoch 9/5000
1103/1103 [==============================] - 0s 136us/step - loss: 1046.3148 - val_loss: 980.86
Epoch 10/5000
1103/1103 [==============================] - 0s 137us/step - loss: 1022.5039 - val_loss: 953.12
Epoch 11/5000
1103/1103 [==============================] - 0s 133us/step - loss: 997.4476 - val_loss: 932.666
Epoch 12/5000
1103/1103 [==============================] - 0s 139us/step - loss: 977.6401 - val_loss: 918.281
Epoch 13/5000
1103/1103 [==============================] - 0s 147us/step - loss: 966.6302 - val_loss: 908.983
Epoch 14/5000
1103/1103 [==============================] - 0s 134us/step - loss: 961.9571 - val_loss: 902.101
Epoch 15/5000
1103/1103 [==============================] - 0s 142us/step - loss: 956.9233 - val_loss: 898.574
Epoch 16/5000
1103/1103 [==============================] - 0s 141us/step - loss: 953.8852 - val_loss: 896.760
Epoch 17/5000
1103/1103 [==============================] - 0s 145us/step - loss: 952.5317 - val_loss: 895.740
Epoch 18/5000
1103/1103 [==============================] - 0s 138us/step - loss: 951.1972 - val_loss: 895.413
Epoch 19/5000
1103/1103 [==============================] - 0s 144us/step - loss: 949.7472 - val_loss: 895.643
Epoch 20/5000
1103/1103 [==============================] - 0s 136us/step - loss: 948.9744 - val_loss: 895.993
Epoch 21/5000
1103/1103 [==============================] - 0s 136us/step - loss: 948.3796 - val_loss: 895.769
Epoch 22/5000
1103/1103 [==============================] - 0s 137us/step - loss: 947.9621 - val_loss: 895.868
Epoch 23/5000
1103/1103 [==============================] - 0s 143us/step - loss: 947.3389 - val_loss: 896.073
Epoch 24/5000
```

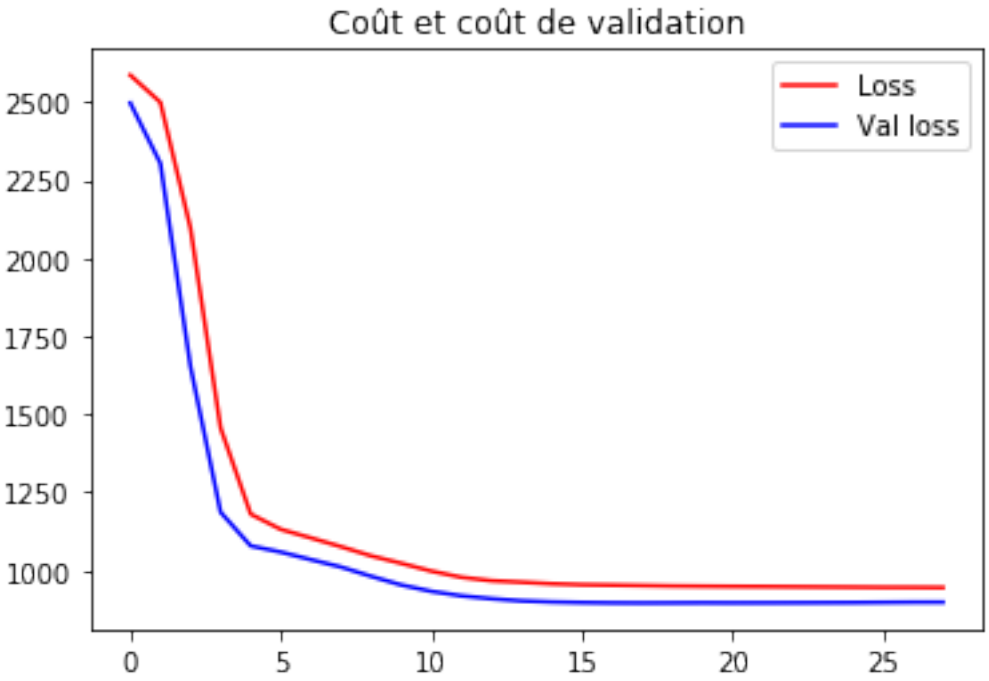
```
1103/1103 [==============================] - 0s 131us/step - loss: 946.7545 - val_loss: 896.357
Epoch 25/5000
1103/1103 [==============================] - 0s 131us/step - loss: 946.2266 - val_loss: 896.656
Epoch 26/5000
1103/1103 [==============================] - 0s 134us/step - loss: 945.6080 - val_loss: 897.296
Epoch 27/5000
1103/1103 [==============================] - 0s 135us/step - loss: 945.4295 - val_loss: 897.875
Epoch 28/5000
1103/1103 [==============================] - 0s 142us/step - loss: 944.9196 - val_loss: 897.934
Epoch 00028: early stopping
```

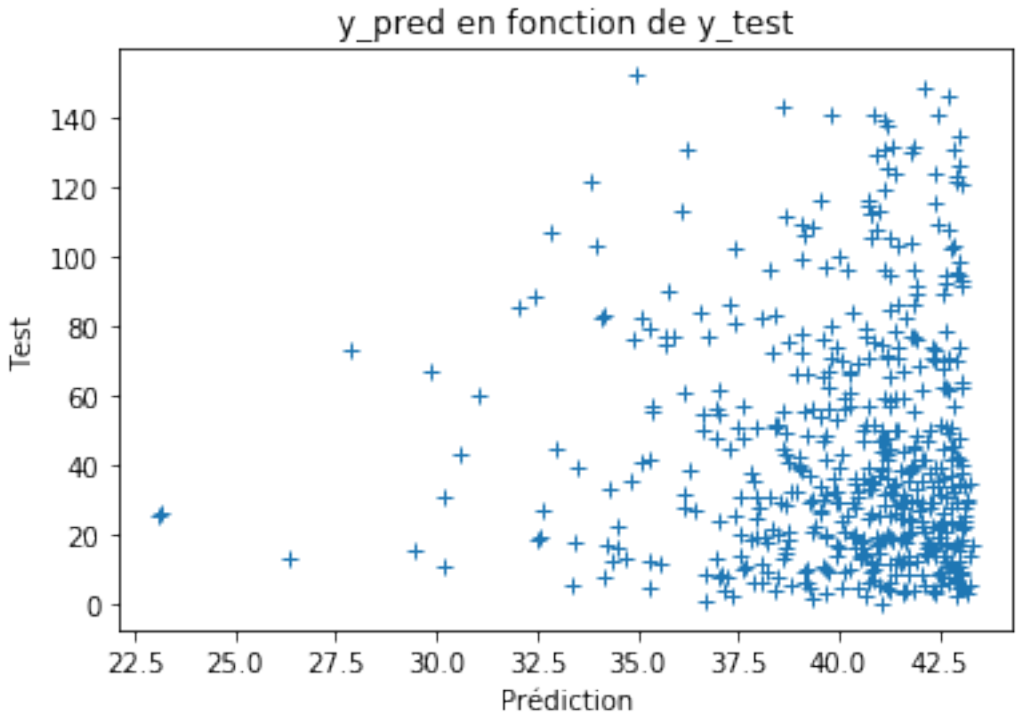
In [9]:
```
y_pred = model.predict(X_test)
plt.title('Coût et coût de validation')
line1,=plt.plot(history.history['loss'], label="Loss", linestyle='-', color='r')
line2,=plt.plot(history.history['val_loss'], label="Val loss", linestyle='-', color='b
first_legend = plt.legend(handles=[line1, line2], loc=1)

plt.show()

plt.title('y_pred en fonction de y_test')

plt.plot(y_pred[:], y_test[:], '+')
plt.ylabel('Test')
plt.xlabel('Prédiction')
plt.show()
```






```
In [10]: model = lstm_model(32, X_train.shape[1:])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 32)                5248
_________________________________________________________________
dense_3 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 6,337
Trainable params: 6,337
Non-trainable params: 0
_________________________________________________________________
```

```
In [11]: early_stopping = EarlyStopping(monitor='val_loss', verbose=1, mode='auto', patience=1
         history = model.fit(X_train, y_train, batch_size=32, epochs=5000, validation_data=(X_
```

```
Train on 1103 samples, validate on 539 samples
Epoch 1/5000
```

```
1103/1103 [==============================] - 1s 1ms/step - loss: 2602.0669 - val_loss: 2512.100
Epoch 2/5000
1103/1103 [==============================] - 0s 439us/step - loss: 2397.3056 - val_loss: 2046.6
Epoch 3/5000
1103/1103 [==============================] - 0s 451us/step - loss: 1767.7182 - val_loss: 1377.6
Epoch 4/5000
1103/1103 [==============================] - 0s 448us/step - loss: 1227.0696 - val_loss: 994.69
Epoch 5/5000
1103/1103 [==============================] - 1s 455us/step - loss: 994.5223 - val_loss: 900.312
Epoch 6/5000
1103/1103 [==============================] - 1s 456us/step - loss: 956.9231 - val_loss: 894.741
Epoch 7/5000
1103/1103 [==============================] - 1s 466us/step - loss: 955.1737 - val_loss: 894.352
Epoch 8/5000
1103/1103 [==============================] - 0s 421us/step - loss: 954.8576 - val_loss: 894.141
Epoch 9/5000
1103/1103 [==============================] - 0s 434us/step - loss: 954.6838 - val_loss: 893.986
Epoch 10/5000
1103/1103 [==============================] - 0s 448us/step - loss: 954.4832 - val_loss: 893.798
Epoch 11/5000
1103/1103 [==============================] - 0s 449us/step - loss: 954.1601 - val_loss: 893.544
Epoch 12/5000
1103/1103 [==============================] - 1s 458us/step - loss: 953.9167 - val_loss: 893.275
Epoch 13/5000
1103/1103 [==============================] - 0s 440us/step - loss: 953.6456 - val_loss: 893.064
Epoch 14/5000
1103/1103 [==============================] - 0s 438us/step - loss: 953.4636 - val_loss: 892.900
Epoch 15/5000
1103/1103 [==============================] - 1s 454us/step - loss: 953.2838 - val_loss: 892.776
Epoch 16/5000
1103/1103 [==============================] - 0s 438us/step - loss: 953.1063 - val_loss: 892.651
Epoch 17/5000
1103/1103 [==============================] - 1s 456us/step - loss: 952.9415 - val_loss: 892.571
Epoch 18/5000
1103/1103 [==============================] - 0s 433us/step - loss: 952.7677 - val_loss: 892.467
Epoch 19/5000
1103/1103 [==============================] - 0s 435us/step - loss: 952.5951 - val_loss: 892.347
Epoch 20/5000
1103/1103 [==============================] - 0s 431us/step - loss: 952.4357 - val_loss: 892.285
Epoch 21/5000
1103/1103 [==============================] - 0s 442us/step - loss: 952.2644 - val_loss: 892.182
Epoch 22/5000
1103/1103 [==============================] - 0s 440us/step - loss: 952.1163 - val_loss: 892.085
Epoch 23/5000
1103/1103 [==============================] - 0s 433us/step - loss: 951.9791 - val_loss: 891.982
Epoch 24/5000
1103/1103 [==============================] - 0s 427us/step - loss: 951.8134 - val_loss: 891.922
Epoch 25/5000
```

```
1103/1103 [==============================] - 0s 425us/step - loss: 951.6609 - val_loss: 891.798
Epoch 26/5000
1103/1103 [==============================] - 0s 438us/step - loss: 951.4890 - val_loss: 891.71
Epoch 27/5000
1103/1103 [==============================] - 0s 431us/step - loss: 951.3000 - val_loss: 891.59
Epoch 28/5000
1103/1103 [==============================] - 0s 435us/step - loss: 951.0595 - val_loss: 891.548
Epoch 29/5000
1103/1103 [==============================] - 0s 431us/step - loss: 950.8856 - val_loss: 891.468
Epoch 30/5000
1103/1103 [==============================] - 0s 431us/step - loss: 950.6964 - val_loss: 891.378
Epoch 31/5000
1103/1103 [==============================] - 0s 431us/step - loss: 950.5310 - val_loss: 891.285
Epoch 32/5000
1103/1103 [==============================] - 0s 433us/step - loss: 950.3660 - val_loss: 891.214
Epoch 33/5000
1103/1103 [==============================] - 0s 430us/step - loss: 950.2094 - val_loss: 891.144
Epoch 34/5000
1103/1103 [==============================] - 0s 421us/step - loss: 950.0384 - val_loss: 891.097
Epoch 35/5000
1103/1103 [==============================] - 0s 443us/step - loss: 949.8884 - val_loss: 891.037
Epoch 36/5000
1103/1103 [==============================] - 0s 416us/step - loss: 949.7465 - val_loss: 890.978
Epoch 37/5000
1103/1103 [==============================] - 0s 415us/step - loss: 949.5845 - val_loss: 890.931
Epoch 38/5000
1103/1103 [==============================] - 0s 415us/step - loss: 949.4621 - val_loss: 890.864
Epoch 39/5000
1103/1103 [==============================] - 0s 417us/step - loss: 949.3042 - val_loss: 890.811
Epoch 40/5000
1103/1103 [==============================] - 0s 415us/step - loss: 949.1718 - val_loss: 890.795
Epoch 41/5000
1103/1103 [==============================] - 0s 417us/step - loss: 949.0282 - val_loss: 890.753
Epoch 42/5000
1103/1103 [==============================] - 0s 421us/step - loss: 948.8836 - val_loss: 890.699
Epoch 43/5000
1103/1103 [==============================] - 0s 435us/step - loss: 948.7761 - val_loss: 890.668
Epoch 44/5000
1103/1103 [==============================] - 0s 414us/step - loss: 948.6360 - val_loss: 890.636
Epoch 45/5000
1103/1103 [==============================] - 0s 418us/step - loss: 948.5187 - val_loss: 890.582
Epoch 46/5000
1103/1103 [==============================] - 0s 420us/step - loss: 948.3814 - val_loss: 890.554
Epoch 47/5000
1103/1103 [==============================] - 0s 433us/step - loss: 948.2810 - val_loss: 890.498
Epoch 48/5000
1103/1103 [==============================] - 0s 430us/step - loss: 948.1567 - val_loss: 890.478
Epoch 49/5000
```

```
1103/1103 [==============================] - 0s 419us/step - loss: 948.0314 - val_loss: 890.449
Epoch 50/5000
1103/1103 [==============================] - 1s 458us/step - loss: 947.8954 - val_loss: 890.422
Epoch 51/5000
1103/1103 [==============================] - 1s 455us/step - loss: 947.7963 - val_loss: 890.410
Epoch 52/5000
1103/1103 [==============================] - 0s 450us/step - loss: 947.6499 - val_loss: 890.368
Epoch 53/5000
1103/1103 [==============================] - 0s 426us/step - loss: 947.5570 - val_loss: 890.346
Epoch 54/5000
1103/1103 [==============================] - 0s 422us/step - loss: 947.4204 - val_loss: 890.322
Epoch 55/5000
1103/1103 [==============================] - 0s 408us/step - loss: 947.3184 - val_loss: 890.308
Epoch 56/5000
1103/1103 [==============================] - 0s 423us/step - loss: 947.1945 - val_loss: 890.280
Epoch 57/5000
1103/1103 [==============================] - 0s 413us/step - loss: 947.0864 - val_loss: 890.247
Epoch 58/5000
1103/1103 [==============================] - 0s 412us/step - loss: 946.9647 - val_loss: 890.244
Epoch 59/5000
1103/1103 [==============================] - 0s 411us/step - loss: 946.8399 - val_loss: 890.223
Epoch 60/5000
1103/1103 [==============================] - 0s 410us/step - loss: 946.7327 - val_loss: 890.201
Epoch 61/5000
1103/1103 [==============================] - 0s 418us/step - loss: 946.6185 - val_loss: 890.192
Epoch 62/5000
1103/1103 [==============================] - 0s 450us/step - loss: 946.4953 - val_loss: 890.168
Epoch 63/5000
1103/1103 [==============================] - 0s 414us/step - loss: 946.3848 - val_loss: 890.176
Epoch 64/5000
1103/1103 [==============================] - 0s 409us/step - loss: 946.2806 - val_loss: 890.186
Epoch 65/5000
1103/1103 [==============================] - 0s 416us/step - loss: 946.1581 - val_loss: 890.165
Epoch 66/5000
1103/1103 [==============================] - 0s 417us/step - loss: 944.7420 - val_loss: 939.306
Epoch 67/5000
1103/1103 [==============================] - 0s 421us/step - loss: 962.2148 - val_loss: 890.101
Epoch 68/5000
1103/1103 [==============================] - 0s 417us/step - loss: 947.0721 - val_loss: 890.807
Epoch 69/5000
1103/1103 [==============================] - 0s 420us/step - loss: 946.5721 - val_loss: 890.084
Epoch 70/5000
1103/1103 [==============================] - 0s 429us/step - loss: 945.1687 - val_loss: 890.265
Epoch 71/5000
1103/1103 [==============================] - 0s 406us/step - loss: 944.1939 - val_loss: 890.751
Epoch 72/5000
1103/1103 [==============================] - 0s 401us/step - loss: 943.8347 - val_loss: 890.772
Epoch 73/5000
```

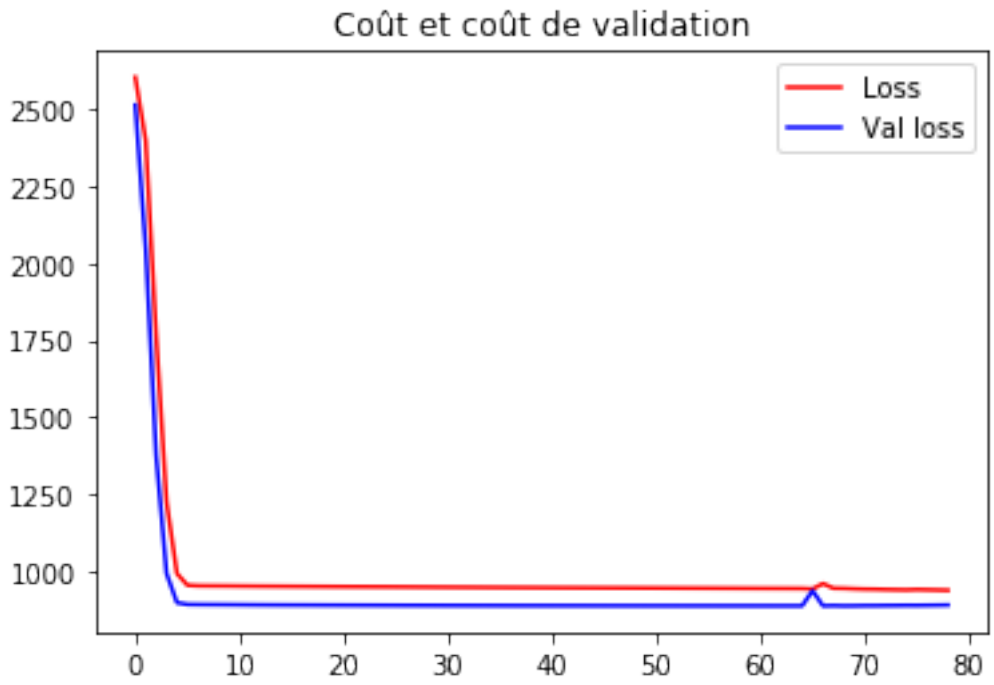
```
1103/1103 [==============================] - 0s 409us/step - loss: 942.9835 - val_loss: 891.127
Epoch 74/5000
1103/1103 [==============================] - 0s 398us/step - loss: 942.3602 - val_loss: 891.303
Epoch 75/5000
1103/1103 [==============================] - 0s 413us/step - loss: 941.8355 - val_loss: 891.643
Epoch 76/5000
1103/1103 [==============================] - 0s 397us/step - loss: 942.8349 - val_loss: 891.485
Epoch 77/5000
1103/1103 [==============================] - 0s 401us/step - loss: 942.1834 - val_loss: 891.858
Epoch 78/5000
1103/1103 [==============================] - 0s 415us/step - loss: 941.4496 - val_loss: 892.217
Epoch 79/5000
1103/1103 [==============================] - 0s 396us/step - loss: 940.6136 - val_loss: 892.709
Epoch 00079: early stopping
```

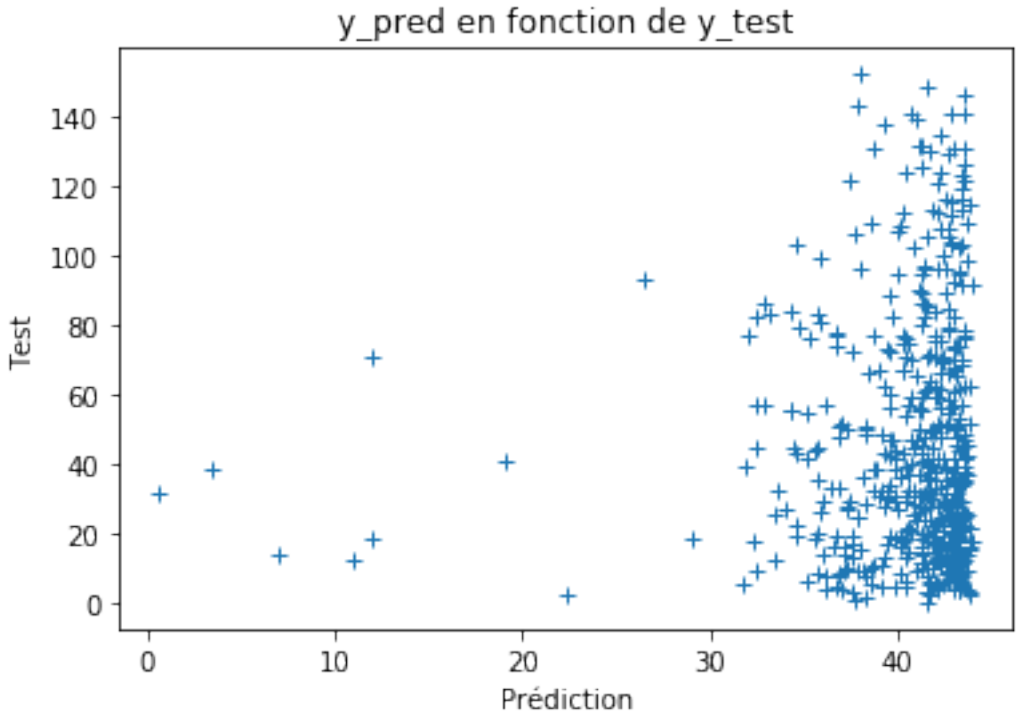
```
In [12]: y_pred = model.predict(X_test)
         plt.title('Coût et coût de validation')
         line1,=plt.plot(history.history['loss'], label="Loss", linestyle='-', color='r')
         line2,=plt.plot(history.history['val_loss'], label="Val loss", linestyle='-', color='
         first_legend = plt.legend(handles=[line1, line2], loc=1)

         plt.show()

         plt.title('y_pred en fonction de y_test')

         plt.plot(y_pred[:], y_test[:], '+')
         plt.ylabel('Test')
         plt.xlabel('Prédiction')
         plt.show()
```






```
In [13]: model = gru_model(32, X_train.shape[1:])
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_1 (GRU)                  (None, 32)                3936
_________________________________________________________________
dense_5 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 33
=================================================================
Total params: 5,025
Trainable params: 5,025
Non-trainable params: 0
_________________________________________________________________
```

```
In [14]: early_stopping = EarlyStopping(monitor='val_loss', verbose=1, mode='auto', patience=1
         history = model.fit(X_train, y_train, batch_size=32, epochs=5000, validation_data=(X_
```

```
Train on 1103 samples, validate on 539 samples
Epoch 1/5000
1103/1103 [==============================] - 1s 1ms/step - loss: 2605.0681 - val_loss: 2531.049
Epoch 2/5000
1103/1103 [==============================] - 0s 328us/step - loss: 2552.8981 - val_loss: 2347.9
Epoch 3/5000
1103/1103 [==============================] - 0s 332us/step - loss: 1990.8319 - val_loss: 1527.0
Epoch 4/5000
1103/1103 [==============================] - 0s 327us/step - loss: 1320.8043 - val_loss: 1035.4
Epoch 5/5000
1103/1103 [==============================] - 0s 326us/step - loss: 1011.5817 - val_loss: 902.88
Epoch 6/5000
1103/1103 [==============================] - 0s 328us/step - loss: 957.9183 - val_loss: 894.874
Epoch 7/5000
1103/1103 [==============================] - 0s 329us/step - loss: 955.9760 - val_loss: 894.844
Epoch 8/5000
1103/1103 [==============================] - 0s 335us/step - loss: 955.9011 - val_loss: 894.826
Epoch 9/5000
1103/1103 [==============================] - 0s 326us/step - loss: 955.8164 - val_loss: 894.795
Epoch 10/5000
1103/1103 [==============================] - 0s 328us/step - loss: 955.7115 - val_loss: 894.750
Epoch 11/5000
1103/1103 [==============================] - 0s 327us/step - loss: 955.5734 - val_loss: 894.687
Epoch 12/5000
1103/1103 [==============================] - 0s 327us/step - loss: 955.3488 - val_loss: 894.529
Epoch 13/5000
1103/1103 [==============================] - 0s 329us/step - loss: 954.7831 - val_loss: 893.955
Epoch 14/5000
1103/1103 [==============================] - 0s 330us/step - loss: 953.4876 - val_loss: 893.111
Epoch 15/5000
```

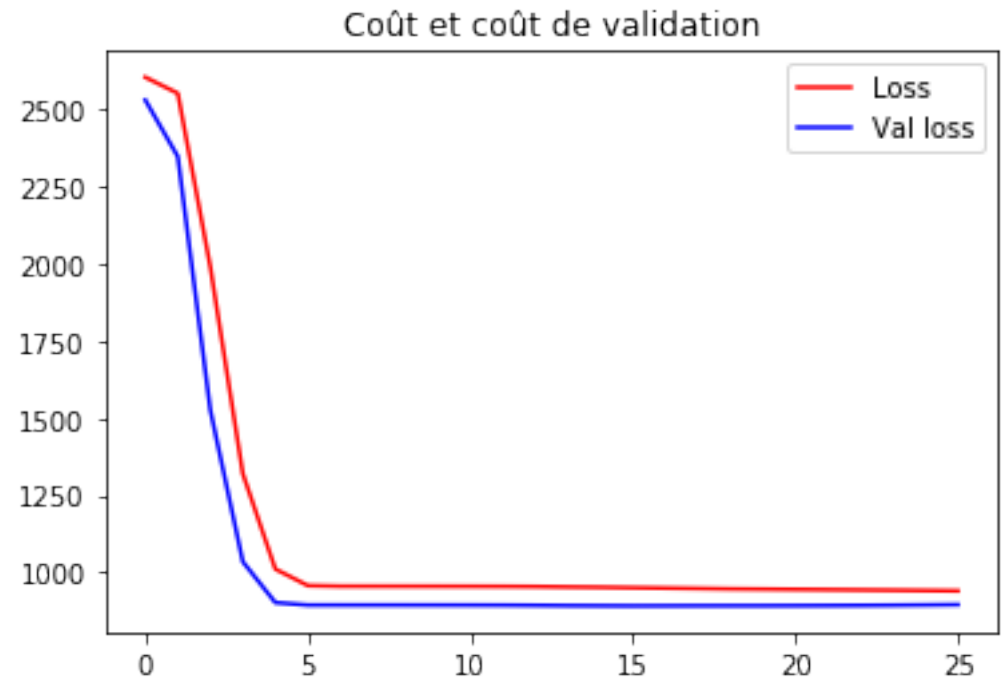
```
1103/1103 [==============================] - 0s 344us/step - loss: 952.5261 - val_loss: 892.670
Epoch 16/5000
1103/1103 [==============================] - 0s 327us/step - loss: 951.1643 - val_loss: 892.338
Epoch 17/5000
1103/1103 [==============================] - 0s 329us/step - loss: 949.8409 - val_loss: 892.483
Epoch 18/5000
1103/1103 [==============================] - 0s 327us/step - loss: 948.5831 - val_loss: 892.702
Epoch 19/5000
1103/1103 [==============================] - 0s 327us/step - loss: 947.1857 - val_loss: 892.769
Epoch 20/5000
1103/1103 [==============================] - 0s 327us/step - loss: 946.4406 - val_loss: 892.812
Epoch 21/5000
1103/1103 [==============================] - 0s 330us/step - loss: 945.6060 - val_loss: 893.016
Epoch 22/5000
1103/1103 [==============================] - 0s 327us/step - loss: 944.7546 - val_loss: 893.306
Epoch 23/5000
1103/1103 [==============================] - 0s 328us/step - loss: 943.8813 - val_loss: 893.704
Epoch 24/5000
1103/1103 [==============================] - 0s 329us/step - loss: 942.9889 - val_loss: 894.338
Epoch 25/5000
1103/1103 [==============================] - 0s 329us/step - loss: 941.9872 - val_loss: 895.179
Epoch 26/5000
1103/1103 [==============================] - 0s 328us/step - loss: 940.9058 - val_loss: 896.182
Epoch 00026: early stopping
```

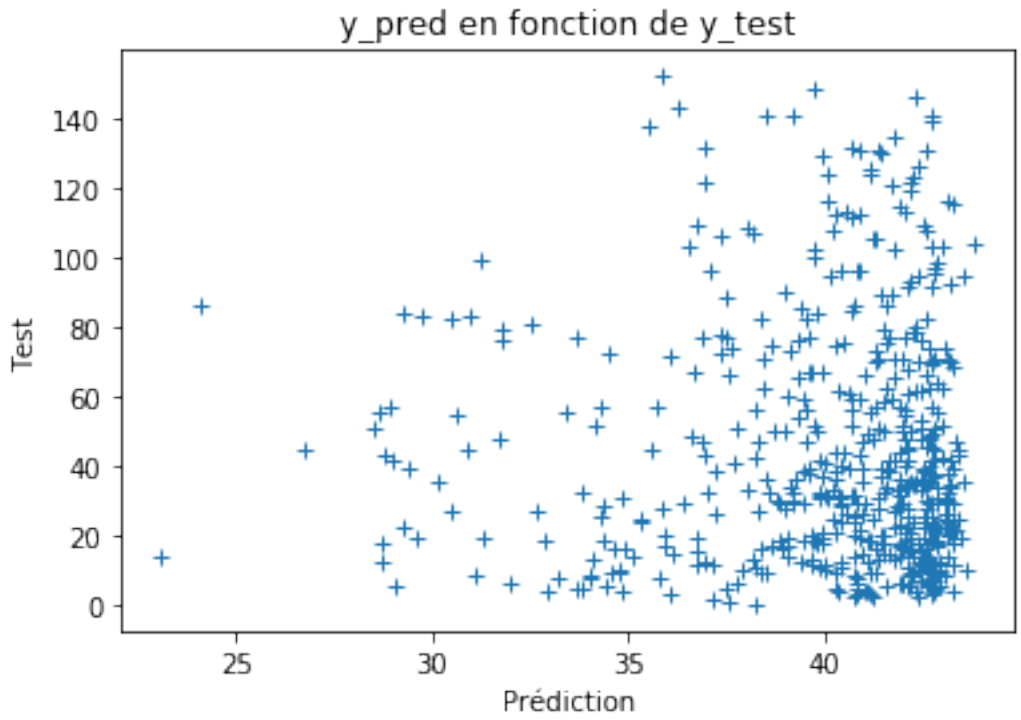
```
In [15]: y_pred = model.predict(X_test)
         plt.title('Coût et coût de validation')
         line1,=plt.plot(history.history['loss'], label="Loss", linestyle='-', color='r')
         line2,=plt.plot(history.history['val_loss'], label="Val loss", linestyle='-', color='
         first_legend = plt.legend(handles=[line1, line2], loc=1)

         plt.show()

         plt.title('y_pred en fonction de y_test')

         plt.plot(y_pred[:], y_test[:], '+')
         plt.ylabel('Test')
         plt.xlabel('Prédiction')
         plt.show()
```


Coût et coût de validation
Loss
Val loss
2500
2250
2000
1750
1500
1250
1000
0
5
10
15
20
25


y_pred en fonction de y_test
Test
Prédiction
140
120
100
80
60
40
20
0
25
30
35
40

In [ ]: