

```

import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from scipy import stats
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
import sklearn
import sys
import io

def plot_learning_curves(loss, val_loss):
    plt.figure()
    plt.plot(np.arange(len(loss)), loss, "b.-", label="Training loss")
    plt.plot(np.arange(len(val_loss)), val_loss, "r.-", label="Validation loss")
    plt.gca().xaxis.set_major_locator(mpl.ticker.MaxNLocator(integer=True))
    plt.legend(fontsize=14)
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.grid(True)

n_steps = 50
forecast = 10

#EXTRACT FLOW, Z SCORE, OUTLIERS
data_west = pd.read_csv('denoised_data_1015_east.csv')
data_west_o = np.array(data_west.FLOW)

#EXRTEND DATA
array_to_concatinate = data_west_o
for iter in range (35):
    data_west_o = np.concatenate([data_west_o,array_to_concatinate])

#SCALE AND RESHAPE DATA
scaler = MinMaxScaler()
array = data_west_o.reshape(-1, 1)
array_scaled = scaler.fit_transform(array)

flow_resaped = array_scaled[:len(array_scaled) - (len(array_scaled) % (n_steps+forecast))]
#TRAIN SET, VALIDATION SET, TEST SET
test = int(0.7 * flow_resaped.shape[0])
valid = int(0.9 * flow_resaped.shape[0])

X_train = flow_resaped[:test, :n_steps]
X_valid = flow_resaped[test:valid, :n_steps]
X_test = flow_resaped[valid:, :n_steps]
print(X_test.shape)
print(X_test[-1:].shape)

#prepare targets
Y = np.empty((flow_resaped.shape[0], n_steps, forecast))
for step_ahead in range(1, forecast + 1):
    Y[:, :, step_ahead - 1] = flow_resaped[:, step_ahead:step_ahead + n_steps, 0]

y_train = Y[:test]
y_valid = Y[test:valid]
y_test = Y[valid:]

#MODEL
def last_time_step_mse(Y_true, Y_pred):
    return keras.metrics.mean_squared_error(Y_true[:, -1], Y_pred[:, -1])

```



```
(1072, 50, 1)
(1, 50, 1)
Train on 7499 samples, validate on 2142 samples
Epoch 1/1000
7499/7499 [=====] - 5s 715us/sample - loss: 0.0432 - las
Epoch 2/1000
7499/7499 [=====] - 5s 617us/sample - loss: 0.0088 - las
Epoch 3/1000
7499/7499 [=====] - 5s 620us/sample - loss: 0.0080 - las
Epoch 4/1000
7499/7499 [=====] - 5s 630us/sample - loss: 0.0074 - las
Epoch 5/1000
704/7499 [=>.....] - ETA: 3s - loss: 0.0074 - last_time_s
```

```
-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-4-3fcaab120557> in <module>()
    73 early_stopping_cb = keras.callbacks.EarlyStopping(patience=15, restore_be
    74 history = model.fit(X_train, y_train, epochs=1000,
--> 75         validation_data=(X_valid, y_valid), callbacks=[early_
    76
    77 model.save("vawenet_1015_east.h5")
```

⏏ 3 frames

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/client/session.py in __c
1456         ret = tf_session.TF_SessionRunCallable(self._session._session,
1457         self._handle, args,
-> 1458         run_metadata_ptr)
1459     if run_metadata:
1460         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)
```

KeyboardInterrupt:

SEARCH STACK OVERFLOW

