Multivariate Sequence To Sequence Forecasting

(B-yond Assignment)

By: Puspanjali Sahoo

Date: 1st December 2019

Table of Contents

[1 Introduction 3](#_Toc26133508)

[2 Dataset 3](#_Toc26133509)

[2.1 Understanding Data 3](#_Toc26133510)

[2.2 Visualizing the Data 3](#_Toc26133511)

[3 Methodology 4](#_Toc26133512)

[3.1 LSTM (Encoder-Decoder) Model: 4](#_Toc26133513)

[3.2 VAR (Vector Auto Regression) Model: 4](#_Toc26133514)

[4 Python Implementation 5](#_Toc26133515)

[4.1 Getting the input Data: 5](#_Toc26133516)

[4.2 Models: 5](#_Toc26133517)

[4.2.1 LSTM (Encoder-Decoder) Model: 5](#_Toc26133518)

[4.2.2 VAR (Vector Auto Regression) Model: 5](#_Toc26133519)

[4.3 Writing into the output file: 5](#_Toc26133520)

[5 Hardware & Software 5](#_Toc26133521)

[6 Submitted File information 6](#_Toc26133522)

[6.1 LSTM (Encoder-Decoder) Model: 6](#_Toc26133523)

[6.2 VAR (Vector Auto Regression) Model: 6](#_Toc26133524)

# Introduction

In this assignment, there are three features (Val\_1, Val\_2 &Val\_3) are given as input. The values of these features have interdependencies on each other that means these values maintain some kind relationship. The objective of this assignment is to predict the next 4 sequences for these 3 features and also to measure the success of the prediction model.

# Dataset

## Understanding Data

The data is stored in “ts\_test\_43.csv” file.

There are total 43 sample are given in the input file.

The 3 features are represented in 3 columns: Val\_1, Val\_2, and Val\_3 and Index is representing the row number.

Index int64

Val\_1 float64

Val\_2 float64

Val\_3 float64

## Visualizing the Data

The val\_1, Val\_2 and Val\_3 are shown in Figure 1 against the index. Figure 2 shows the val\_1, Val\_2 and Val\_3 against the time in terms of time series. The variation of Val\_1 and Val\_3 are similar not like Val\_2.

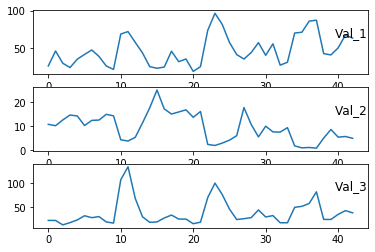


Figure 1: Variation of features against index

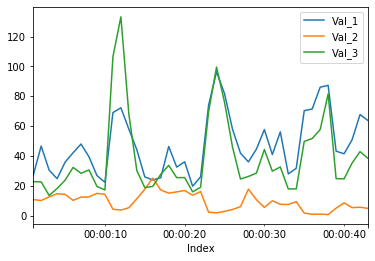


Figure 2: Variation of features against index in terms of time series.

# Methodology

After analyzing the dataset it is observed that all the 3 features i.e, Val\_1, Val\_2, Val\_3 are

interdependent that means they maintain some kind of relationship among themselves. The outputs need to predict, should also maintain that relationship or should be in that sequence. Also, it needs to predict the outputs for next 4 multiple steps.

Two different types of models are used for multi-step forecasting.

## LSTM (Encoder-Decoder) Model:

This is the model known for multivariate forecasting. In this model, multiple parallel series are given as input and multi-step series are forecasted as output. Here both the input and output series have multiple time steps. Also here both input and output are sequences. Therefore, it is known as **sequence to sequence** problem.

The model is having 2 sub-models: encoder and decoder. The encoder is used for reading and interpreting the input sequence and it’s a vanilla LSTM model. The output of the encoder is a fixed length vector representation of the input sequence and it is repeated once for each required time step in the output sequence. The output of the encoder is used as input to the decoder. The model must output a value for each value in the output time step, which can be interpreted by a single output model. The same output layer or layers is used to make each one-step prediction in the output sequence. This can be achieved by wrapping the output part of the model in a [TimeDistributed wrapper](https://machinelearningmastery.com/timedistributed-layer-for-long-short-term-memory-networks-in-python/).

## VAR (Vector Auto Regression) Model:

In the initial part of this experiment, VAR (Vector Auto Regression) model was used because VAR is the most commonly used method for time series forecasting. In a VAR model, each variable is a linear function of the past values of itself and the past values of all the other variables. **VAR is able to understand and use the relationship between several variables**. This is useful for describing the dynamic behavior of the data and also provides better forecasting results.

# Python Implementation

## Getting the input Data:

The input file (ts\_test\_43.docx) is read through google-colab file reader. Then the data was modified to get proper data format. This modified input data was loaded in pandas-dataframe. Then the data was visualized by plotting the graph of all the 3 feature against the indexed value.

Then the whole set of data is divided in training and validation set in the ratio of 80:20. And then the model was trained on the training set data and forecasted the next sequence of dataset that is equivalent to the length of the validation set. Then the results are compared with 2 matrices: MAE (Mean absolute error) and MAPE (Mean absolute percentage error).

## Models:

### LSTM (Encoder-Decoder) Model:

Here multiple time steps are used for each of the time series as input to the model and forecasted the next multiple time steps for each time series as output. Therefore, the input series is splited using “split\_mult\_seq()” function to fulfill this purpose.

The “multivariate\_lstm()” is the function that defines the LSTM encoder-decoder model. The input to this function is the length of the input list, length of the output list and the number of features. Then this model was fitted with the input dataset and the output was forecasted.

### VAR (Vector Auto Regression) Model:

Here the VAR model from “statsmodels.tsa.vector\_ar.var\_model” library is used for forecasting the output. The data type of the 1st column for this model should be a “datetime”. Therefore the index column is converted into the seconds representing it as a “datetime” object. Then this model was trained on the whole date set and the output for the next 4 time step is forecasted.

## 4.3 Writing into the output file:

The forecasted values are rounded up to 3 decimal points are dumped into an “output.csv” file with comma separated values and the next indexed values are added to the corresponding rows. Then this data is modified as the expected output format and saved in “FormatedOutput.csv” file.

# Hardware & Software

* This experiment is done in python in google-colab. The default version on python in google-colab is py3.6.7.
* The default Tesorflow version for google colab is 1.15.0. This is upgraded manually to 2.0.0.
* For implementation of the LSTM model, Keras sequential model is used.
* For reading input file in google-colab, the cookies in the browser should not be blocked.
* The Hardware Accelerator for this experiment is GPU.

# Submitted File information

## LSTM (Encoder-Decoder) Model:

1. Multivariate\_SeqToSeq\_LSTMEncDec.ipynb : Python code
2. ts\_test\_43.csv : Input file
3. input.csv: modified (Corrected format) input file
4. output.csv: output file containing predicted output of the next 4 time step
5. FormatedOutput.csv: the output of the next 4 time step is converted into required format

## VAR (Vector Auto Regression) Model:

1. MultivariateSequence\_VAR.ipynb: Python code
2. ts\_test\_43.csv : Input file
3. input.csv: modified (Corrected format) input file
4. output.csv: output file containing predicted output of the next 4 time step
5. FormatedOutput.csv: the output of the next 4 time step is converted into required format