Modeling time series using LSTMs and TCNs

Hajar Hajji

INRAE/IECL

Supervisors : Gabriel Destouet, Marianne Clausel, Emilie Joetzjer

Table of contents

1) Dataset

Variables

Variable	Description
GPP	Quantity of CO2 assimilated by the forest through photosynthesis
SWC	Soil water content
VPD	Vapor pressure deficit
Time	format : YYYY-MM-DD HH:MM:SS

Table 1: Variable Descriptions

GPP decomposition

- Mathematical formula:

$$GPP_t = \overline{GPP}_t^{365d} + \overline{GPP_t - \overline{GPP}_t^{365d}}^{24h} + \widehat{GPP}_t$$

where

$$\overline{GPP}^T = \frac{1}{N}\sum_{k=1}^{N} GPP_{t+kT}$$

GPP decomposition

- Mathematical formula:

$$GPP_t = \overline{GPP}_t^{365d} + \overline{GPP_t - \overline{GPP}_t^{365d}}^{24h} + \widehat{GPP}_t$$

where

$$\overline{GPP}^T = \frac{1}{N}\sum_{k=1}^{N} GPP_{t+kT}$$

  - $\overline{GPP}_t^{365d}$: Annual mean calculated with $N = 2$.
  - $\overline{GPP_t - \overline{GPP}_t^{365d}}^{24h}$: Day-and-night cycle calculated with $N = 7$.
  - $\widehat{GPP}_t$: Anomalies.

GPP Decomposition

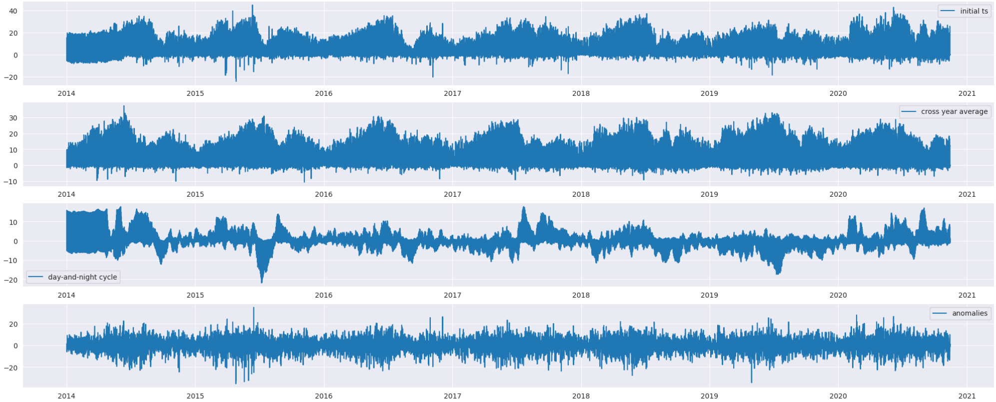


Fig. 1: GPP Decomposition

Time fetaures

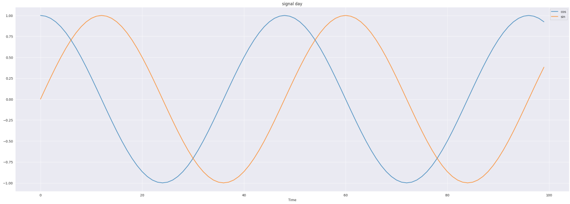


Fig. 2: Day time signal

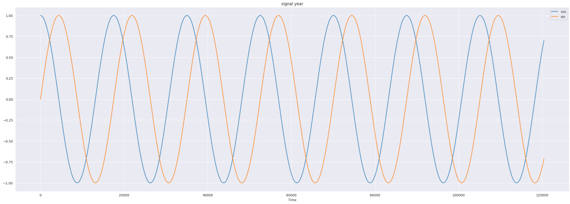


Fig. 3: Year time signal

2) LSTM : Long-Short-Term-Memory

LSTM Inputs & Outputs

- Inputs : Sequences

LSTM Inputs & Outputs

- Inputs : Sequences

$$\left.\begin{array}{l}\text{- Annual GPP}\\ \text{- Day-and-night cycle GPP}\\ \text{- Anomalies}\\ \text{- Time features}\end{array}\right\} \begin{array}{l}\text{from } t-L \text{ to } t\\ L:\text{Sequence length}\end{array}$$

LSTM Inputs & Outputs

- Inputs : Sequences

$$\left.\begin{array}{l}\text{- Annual GPP}\\ \text{- Day-and-night cycle GPP}\\ \text{- Anomalies}\\ \text{- Time features}\end{array}\right\} \begin{array}{l}\text{from } t-L \text{ to } t\\ L:\text{Sequence length}\end{array}$$

- Output :

LSTM Inputs & Outputs

- Inputs : Sequences

$$\left.\begin{array}{l}\text{- Annual GPP}\\ \text{- Day-and-night cycle GPP}\\ \text{- Anomalies}\\ \text{- Time features}\end{array}\right\}\quad \begin{array}{l}\text{from } t-L \text{ to } t\\ L:\text{Sequence length}\end{array}$$

- Output :

$$\left.\begin{array}{l}\text{- Annual GPP}\\ \text{- Day-and-night cycle GPP}\\ \text{- Anomalies}\end{array}\right\}\quad \text{at } t+1$$

Model


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 128)               69632

 dense (Dense)               (None, 3)                 387

=================================================================
Total params: 70,019
Trainable params: 70,019
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4: Model architecture

Hyperparameters

- Sequence length = 365 (7days)
- Features = 7
- Hidden units = 128
- Optimizer = Adam
- Learning rate = 0.001
- Batch size = 32

Cross year average

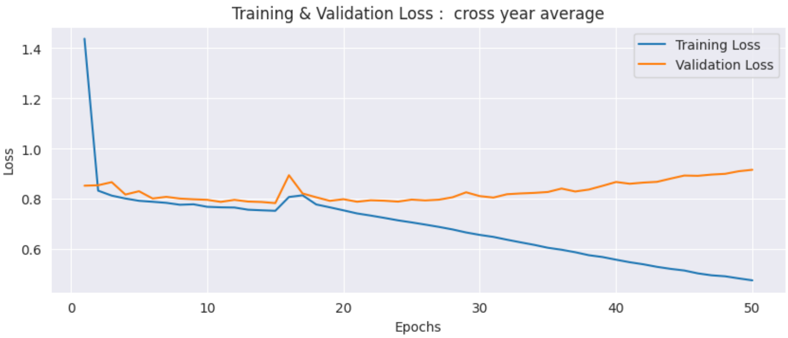


Fig. 5: Training loss VS Validation loss : Cross year average

Day-and-night cycle

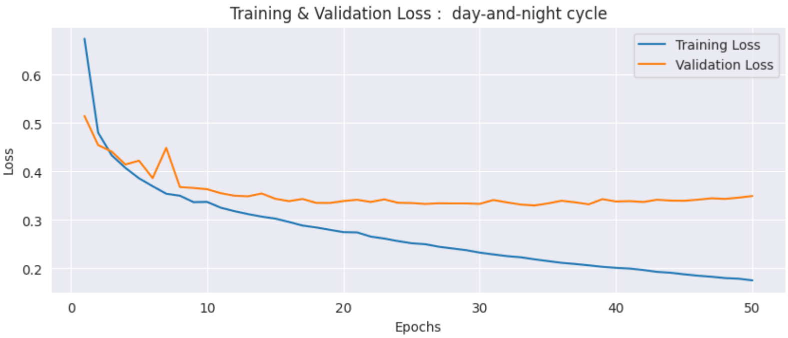


Fig. 6: Training loss VS Validation loss : Day-and-night cycle

Anomalies

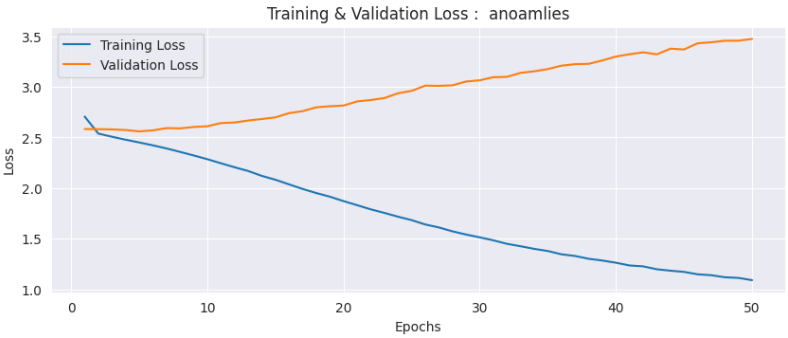


Fig. 7: Training loss VS Validation loss : Anomalies

Hyperparameters

- Sequence length = 700 (2weeks)
- Features = 7
- Hidden units = 128
- Optimizer = Adam
- Learning rate = 0.001
- Batch size = 32

Cross year average

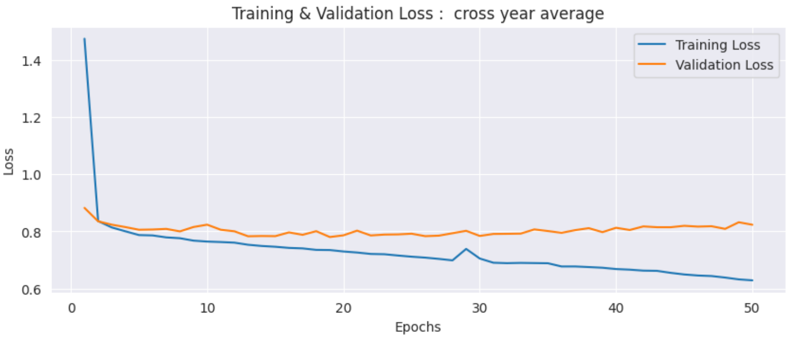


Fig. 8: Training loss VS Validation loss : Cross year average

Day-and-night cycle

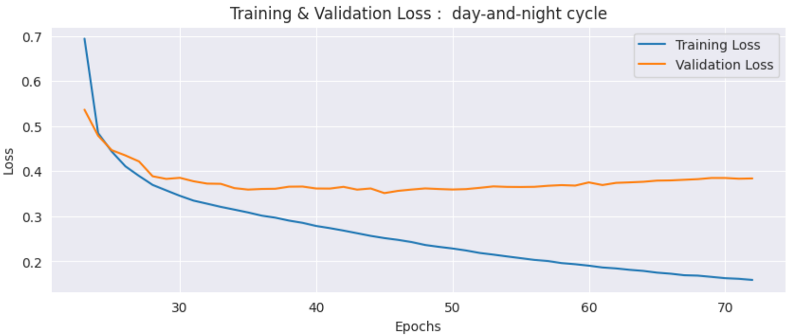


Fig. 9: Training loss VS Validation loss : Day-and-night cycle

Anomalies

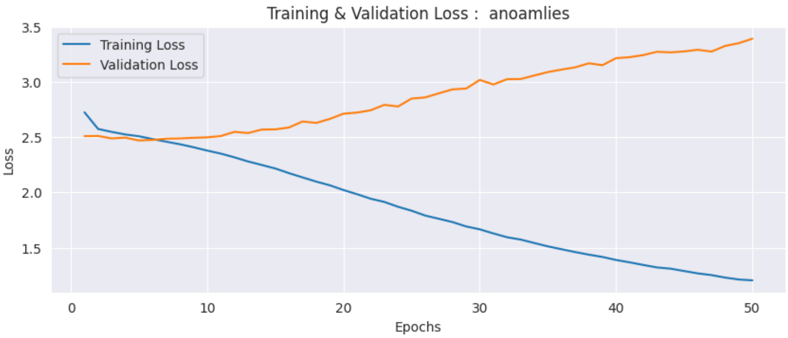


Fig. 10: Training loss VS Validation loss : Anomalies

Model

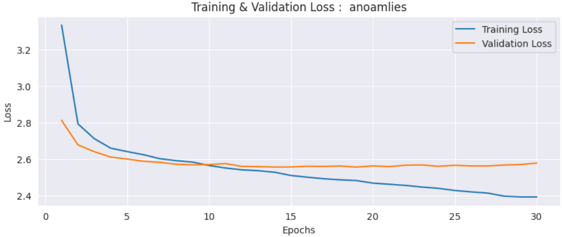

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 365, 64)           18432

 dropout_2 (Dropout)         (None, 365, 64)           0

 lstm_3 (LSTM)               (None, 64)                33024

 dropout_3 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 51,651
Trainable params: 51,651
Non-trainable params: 0
_________________________________________________________________
```

Fig. 11: Model architecture

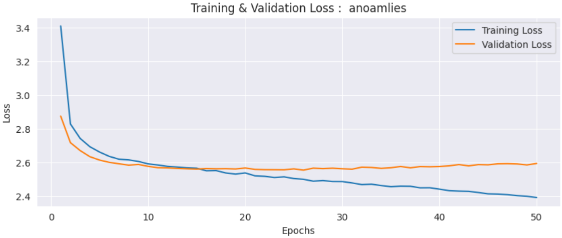
Anomalies

Learning rate $= 0.0001$

Optimizer : Adam



Optimizer : AdamW



3) TCN : Temporal Convolutional Networks

Introducing TCNs

Why privilege TCNs over RNNs ?

- Efficient parallelism.

Why privilege TCNs over RNNs ?

- Efficient parallelism.
- Stable gradients (preventing vanishing/exploding gradients..)

Introducing TCNs

Why privilege TCNs over RNNs ?

- Efficient parallelism.
- Stable gradients (preventing vanishing/exploding gradients..)
- Vision across the sequence => Flexible length inputs.

Architecture

The architecture of TCN consists of :

- Residual Blocks.

The architecture of TCN consists of :

- Residual Blocks.
  - Each residual block consists of two causal convolutional layers with the same dilation factor.

Architecture

The architecture of TCN consists of :

- Residual Blocks.
  - Each residual block consists of two causal convolutional layers with the same dilation factor.
  - Introduce complexity to each layer (Activation, Normalization, Regularization)

Architecture

The architecture of TCN consists of :

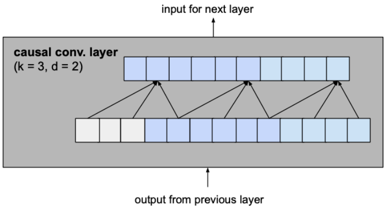
- Residual Blocks.
  - Each residual block consists of two causal convolutional layers with the same dilation factor.
  - Introduce complexity to each layer (Activation, Normalization, Regularization)
  - The output of the convolutional layers is added to the input of the residual block through a residual connection.

Architecture

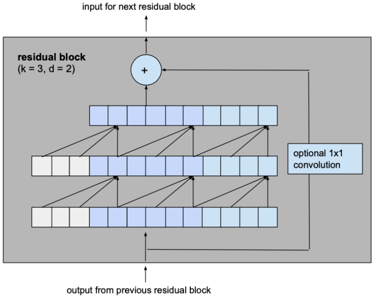
The architecture of TCN consists of :

- Residual Blocks.
    - Each residual block consists of two causal convolutional layers with the same dilation factor.
    - Introduce complexity to each layer (Activation, Normalization, Regularization)
    - The output of the convolutional layers is added to the input of the residual block through a residual connection.
- Sequential arrangement of residual blocks with increasingly dilations (with blocks).

Residual blocks



(a) Causal convolutional layer



(b) Residual block

Fig. 13: Residual blocks

Architecture

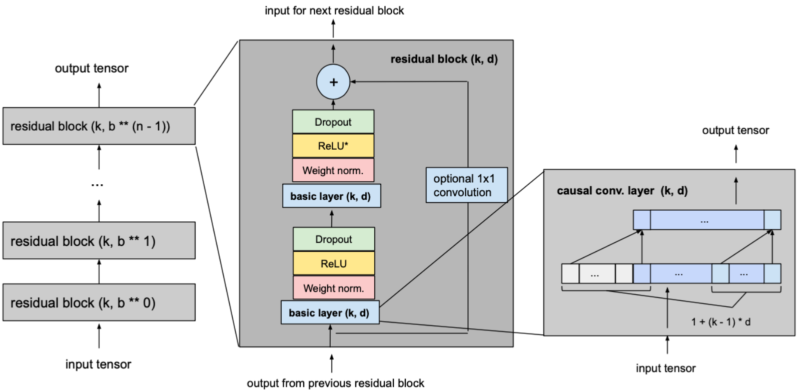


Fig. 14: The entire model

Data preprocessing

- Remove the first year (2014) and last year (2022) (due to too many SWC missing values).

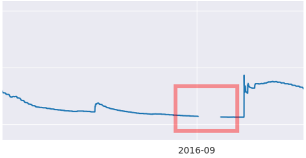


Fig. 15: Missing SWC values on September 2016

Data preprocessing

- Remove the first year (2014) and last year (2022) (due to too many SWC missing values).
- Filling SWC gaps using interpolation and sometimes backward filling.

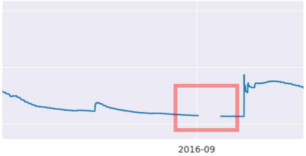


Fig. 15: Missing SWC values on September 2016

- Remove the first year (2014) and last year (2022) (due to too many SWC missing values).
- Filling SWC gaps using interpolation and sometimes backward filling.
- Penalize the loss function based on the number of SWC values that have been filled.

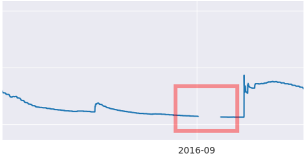


Fig. 15: Missing SWC values on September 2016

TCN Inputs &Outputs

- Inputs : Sequences

TCN Inputs &Outputs

- Inputs : Sequences

$$\left.\begin{array}{l}\text{- GPP}\\ \text{- VPD}\\ \text{- SWC}\\ \text{- Time features}\end{array}\right\} \begin{array}{l}\text{from } t-L \text{ to } t\\ L: \text{Sequence length}\end{array}$$

TCN Inputs &Outputs

- Inputs : Sequences

$$\left.\begin{array}{l}\text{- GPP}\\ \text{- VPD}\\ \text{- SWC}\\ \text{- Time features}\end{array}\right\}\quad \begin{array}{l}\text{from } t-L \text{ to } t\\ L: \text{Sequence length}\end{array}$$

- Output : GPP at t+1

Hyperparameters

- Sequence length = 400 (>one year)
- Features = 7
- Channels = [32, 64, 128]
- Kernel size = 3
- Learning rate = 0.001
- Dropout = 0.2
- Batch size = 32
- Loss function : MSE (with penalization)

TCN Training

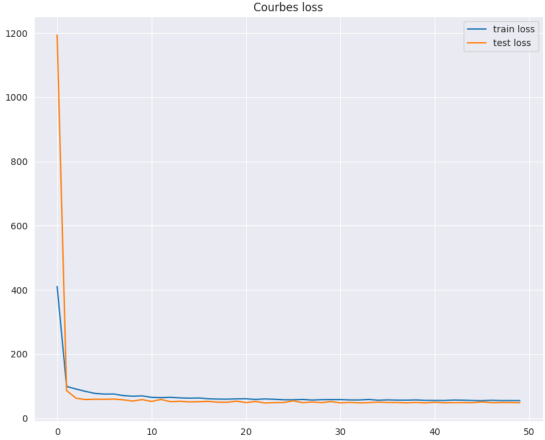


Fig. 16: Training loss VS Validation loss using TCN

Conclusion

Conclusion

Overall good performance of TCN.

Perks :

- Fast &optimal convergence.
- Not time-consuming.
- No need for manual GPP decomposition.
- Interpretability.

Drawbacks :

- Requires long sequences to capture temporal patterns $=>$ significant computational resources.