Predicting Temperature In Hungary Using Multiple Parameters From Ebos-datasets

Beile Zhang, Jiazhen Jia, Zhiyu Zheng

Agenda

- **Intro & Background**
- **Dataset**
- **Methods**
    - NN
    - CNN
    - LSTM
- **Comparison of different models**
- **Conclusion**

Background

- Surface air temperature (SAT) regulates the water cycle (ET), ecosystems, and human systems.
- NWP is skillful but limited by high compute cost, coarse resolution, and regional biases.
- Machine learning complements NWP by learning nonlinear, multivariate relations from gridded data.
- CNNs provide spatial inductive bias (local filters, translation equivariance) suited to geophysical grids; MLPs flatten and lose structure.

Introduction

Objective:

1. Whether machine learning can improve SAT forecasts over Hungary region
2. Examine how different model choice influence performance

Predictors:

Daily precipitation sum (RR), Mean sea-level pressure (PP), Relative humidity (HU), and Global radiation (QQ)

Target:

Daily mean temperature (TG)

Get & check data

Data source(E-OBS dataset):

'https://ncsa.osn.xsede.org/Pangeo/pangeo-forge/pangeo-forge/EOBS-feedstock/eobs-tg-tn-tx-rr-hu-pp.zarr'

'https://ncsa.osn.xsede.org/Pangeo/pangeo-forge/pangeo-forge/EOBS-feedstock/eobs-surface-downwelling.zarr'

check the data:

Time serial: 1950-01-01T00:00:00.000000000 to 2020-12-31T00:00:00.000000000

Spatial:

Latitude from: 25.05 to 71.45(for dataset 1) & 25.05 to 70.95(for dataset 2)

Longitude from: -24.95 to 45.45(for dataset 1) & -24.95 to 44.95(for dataset 2)

Data Preprocessing

1. Region & Time Selection

Region: Hungary(latitude from 45 to 49 and longitude from 16 to 23)

Time Series: from 1950-01-01 to 2020-12-31

2. Cleaning & Alignment

Sort by time and remove duplicate timestamps

Using linear interpolation(lat & lon) for NaN

3. Data split

80% for training and validation(with 64% for training & 16% for validation)

20% for testing

Data Preprocessing

4. Inputs for different model

CNN: (time, channels = 4, lat, lon) #channels = 4 variables i.e [rr,pp,hu,qq]

NN: (time, 4 * lat * lon)

LSTM:

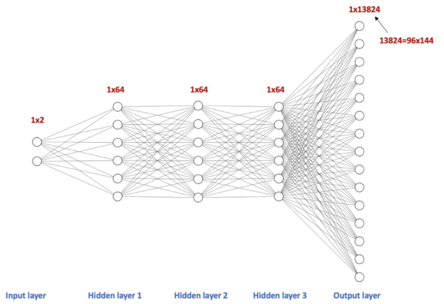
X:(batch, seq_len, 4 * lat * lon)

Y:(batch, lat * lon)

5. Normalization and replace NaN with 0

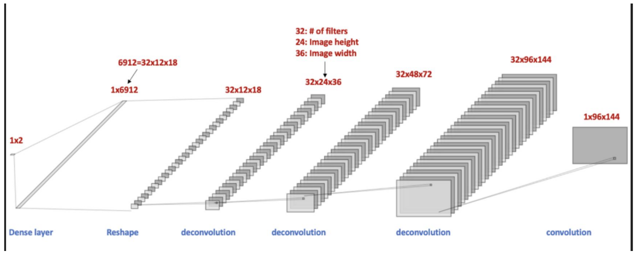
Methods

NN: Neural Network



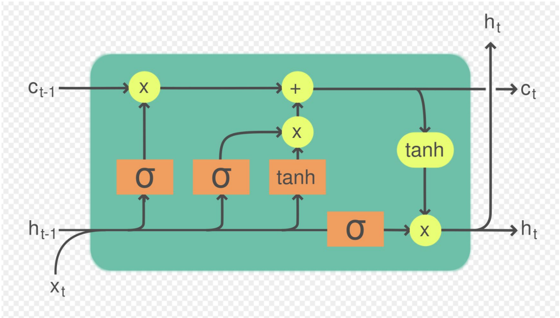
Limits: Ignoring spatial & temporal dimensions

CNN:Convolutional Neural Network



Limits: Ignoring temporal dimension

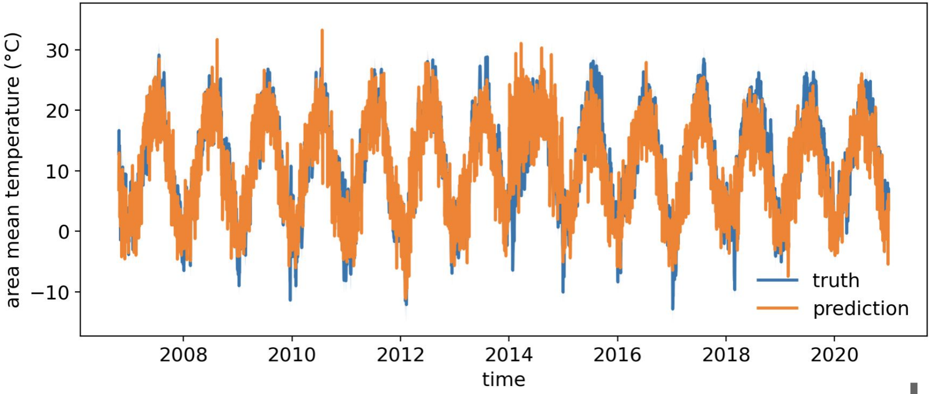
LSTM: Long-Short term memory



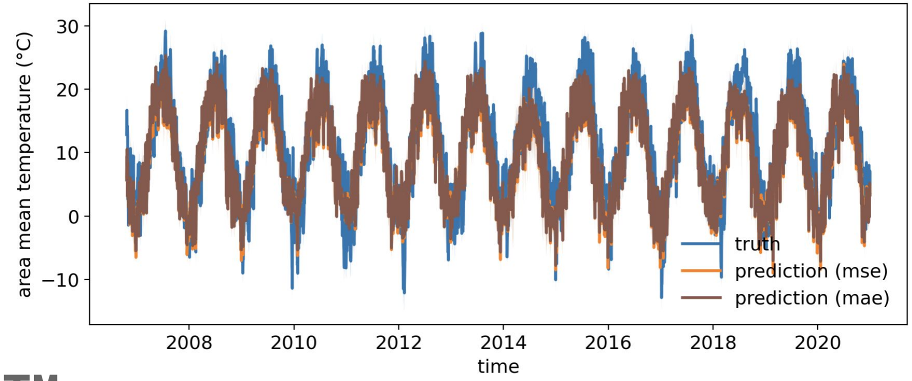
Limits: Ignoring spatial dimension

Comparison

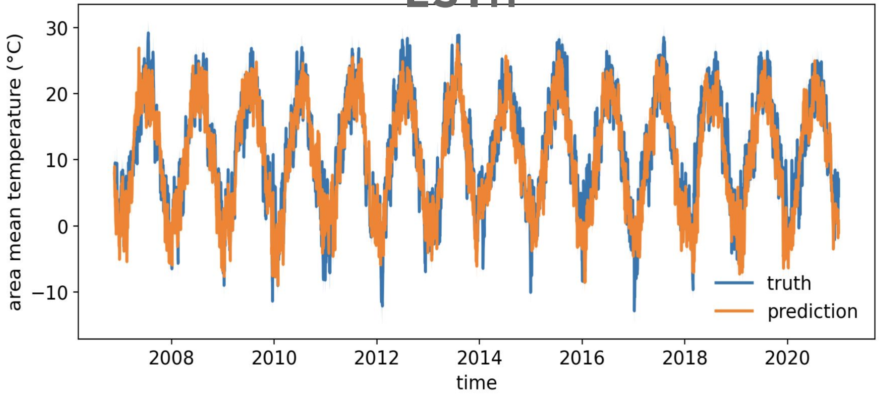
NN



CNN



LSTM



Comparison

RMSE

- NN:4.06
- CNN:
  - 365 started(1950-05-01)
    - MSE:3.25
    - MAE:2.69
  - 1-day
    - MSE:4.18
    - MAE:4.12
- LSTM: 3.16

LSTM has the best performance

1. Since temperature is strongly related to temporal dimension, and LSTM model learns weather evolution patterns
2. NaN values in our dataset will greatly affect the accuracy of NN & CNN model
3. The forget/input/output gates enable LSTM to selectively retain historical information while discarding noise, therefore overcoming the vanishing gradient problem of simple RNNs.

Conclusion

-LSTM Achieves Best Overall Performance

-Why LSTM Works Better?

-Real-World Applications Enhanced by LSTM

-Future Directions

Thank you!