POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework report

Homework 3

Scientific computing tools for advanced mathematical modelling

Authors: Andrea Moreschi - Andrea Novellini - Luca Sosta

Academic year: 2021-2022

1. Mathematical formulation of the problem

The challenge required us to predict the pandemic trend in the following 7 days starting from daily acquired data, developing a physics-based or data-based technique. We went for the latter option, implementing a recurrent neural network (RNN).

In particular, we were asked to create a model that was able to forecast four predictors (*New infections, Hospitalized, Recovered, Deceased*) for three different regions (*Lombardia, Lazio, Sicilia*) in order to understand the developing of the pandemic.

We achieved these results training the RNN minimizing the following cost function:

$$\mathcal{L}(w, \mathbf{y_{true}}, \mathbf{y_{pred}}) = \mathbf{MSE}(\mathbf{y_{true}}, \mathbf{y_{pred}}(w))$$

Where $\mathbf{y_{true}}$ represent the true values of one or more predictor for a 7 days window and $\mathbf{y_{pred}}(w)$ the prediction of the network.

2. Methods

Importing the dataset

First step was to import the dataset [1] and, from it, to extract meaningful information in order to make our predictions. Since the provided dataset was structured day-by-day, we made a small python script (’data_generator.py’) that for each day gathered all the necessary information and grouped them into three datasets, one for each region. Figure 1 shows a plot of the datasets.

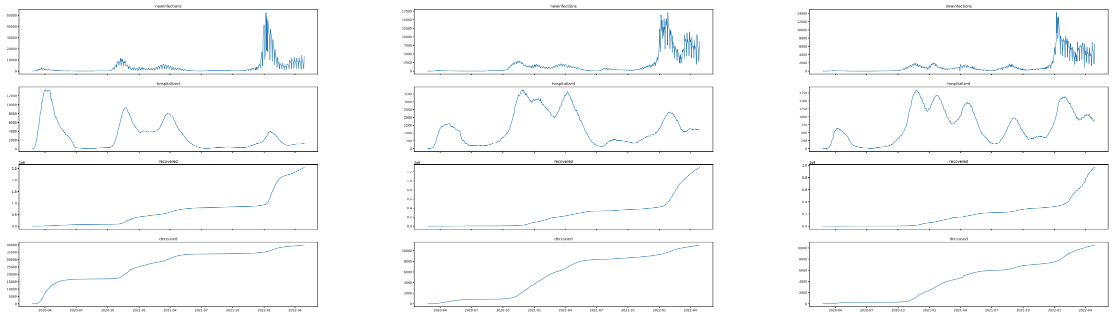
Smoothing, Encoding and Decoding

From Figure 1 we can see that the data we had were very noisy, due to the way they have been collected. It is very clear that every Sunday there is a substantial drop in the newly recorded cases. For this reason, the following logical step for us was to find a way to smooth data in order to have a more homogeneous representation of what’s going on and, moreover, to help the training of our model.

To do so, we used the Savitzky–Golay filter, that uses convolution to fit successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares, together with a logarithmic normalization of the data.

Moreover, we noticed that the data collected for the predictors were not consistent. For instance, *New infections* and *Hospitalized* were recorded as daily quantities, whereas *Deceased* and *Recovered* as cumulative ones. Therefore, to have a more consistent dataset, we encoded every predictor so that each of them represented increments between consecutive days.

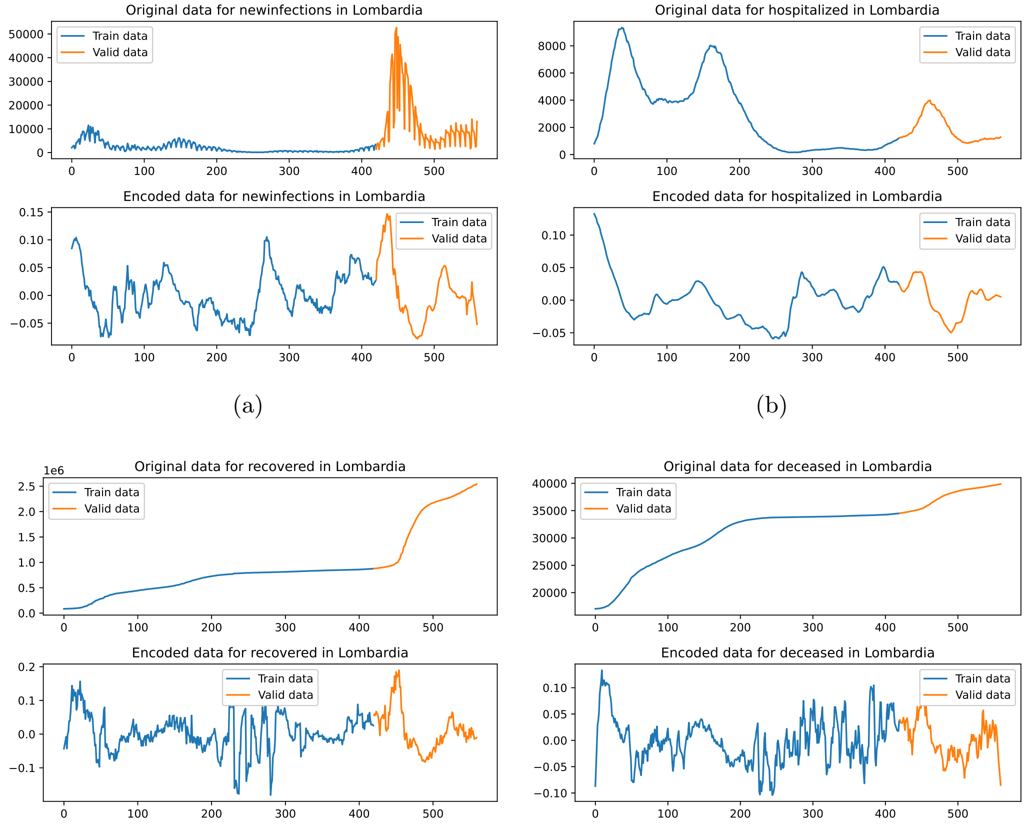
We proceeded in this way: for what concerns *New infections* and *Hospitalized* we first smoothed the dataset



(a) (b) (c)

Figure 1: (a) Lombardia, (b) Lazio, (c) Sicilia.
From the top to the bottom we have *New infections, Hospitalized, Recovered* and *Deceased*

with the aforementioned filter, applied the logarithmic function to take into account the exponential growth of the data and then, for each day, took the difference with respect to the previous one. *Deceased* and *Recovered* needed a further step: we computed the daily quantities applying the same difference also before applying the smoothing and the logarithmic function.
Now each predictor represents an exponential normalized daily increment. In Figure 2 you can see the original data and the encoded ones for Lombardia, divided into *training set* (blue) and *validation set* (orange).



(a) (b)

(c) (d)

Figure 2: (a) New infections, (b) Hospitalized, (c) Recovered, (d) Deceased

The model

We created our train and validation dataset by means of the *WindowGenerator* object. The hyperparameters that the user need to specify are the following:

- *input_width*: The length of the input window, i.e. how many days in the past I want to use to predict the future. In our case *input_width* = 28.
- *label_width*: The length of the predicted window, i.e. how many days in the future I want to predict. In our case *label_width* = 7.
- *shift*: The distance between two consecutive training windows, i.e. of how many days the next training window is shifted wrt to the previous. In our case *shift* = 7.

The architecture of the network employed can be seen in Figure 3 and it is quite simple. This is due to the fact that the temporal interval employed consists of moreover 500 days and a more complex network could have easily caused overfitting problems.
It consists of:

1. LSTM layer with 256 units, *hyperbolic tangent* activations and *sigmoid* recurrent activations.
2. Dense layer with a number of units given by the product of the number of days and the number of predictors to predict and *linear* activation.
3. Reshape layer to obtain the desired shape to postprocess the data.

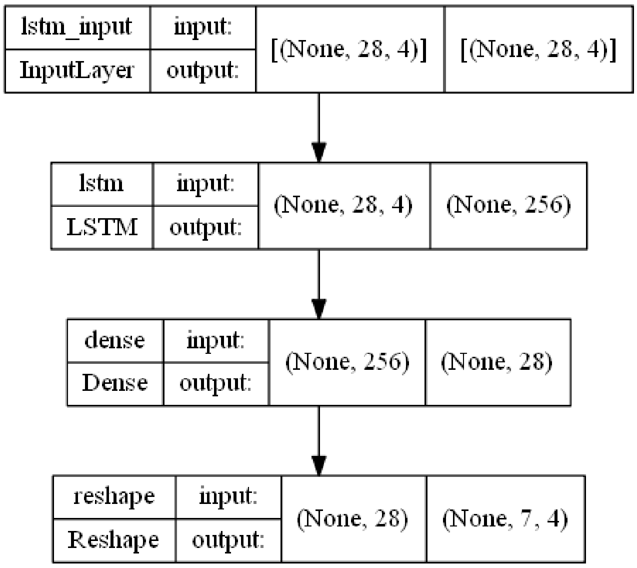
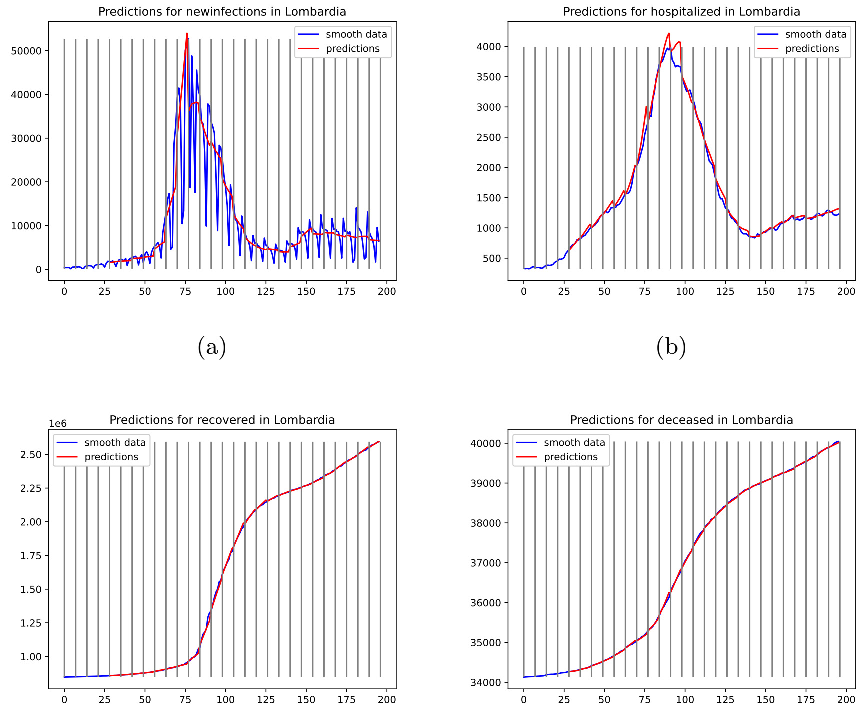

lstm_input | input: | [(None, 28, 4)] | [(None, 28, 4)]
InputLayer | output:
lstm | input: | (None, 28, 4) | (None, 256)
LSTM | output:
dense | input: | (None, 256) | (None, 28)
Dense | output:
reshape | input: | (None, 28) | (None, 7, 4)
Reshape | output:

Figure 3: Architecture used for predicting 7 days for all the 4 predictors simultaneously

3. Numerical results

In Figure 6 we can see the results of our model on a validation set never seen by the network in Lombardia. The results for the other two regions show a very similar trend and are not reported.
We can immediately notice different features of our model:

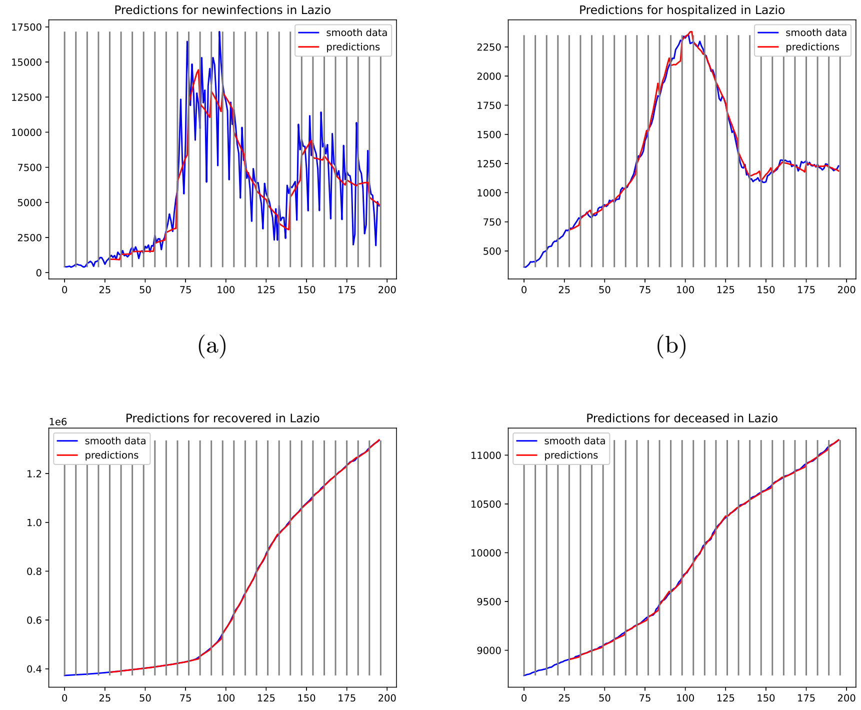
- It is able to predict very effectively the cumulative number of deceased and recovered.
- It is able to predict with a good amount of accuracy the trend of the daily hospitalized, predicting correctly maxima and minima of the functions
- It is able to catch the general trend for the new infections. Unfortunately, the model cannot predict the drop of cases during the Sundays and the consequent increase on Monday. If necessary, one could take into account this info to proceed with a post-process of the results.



(a) (b)

(c) (d)

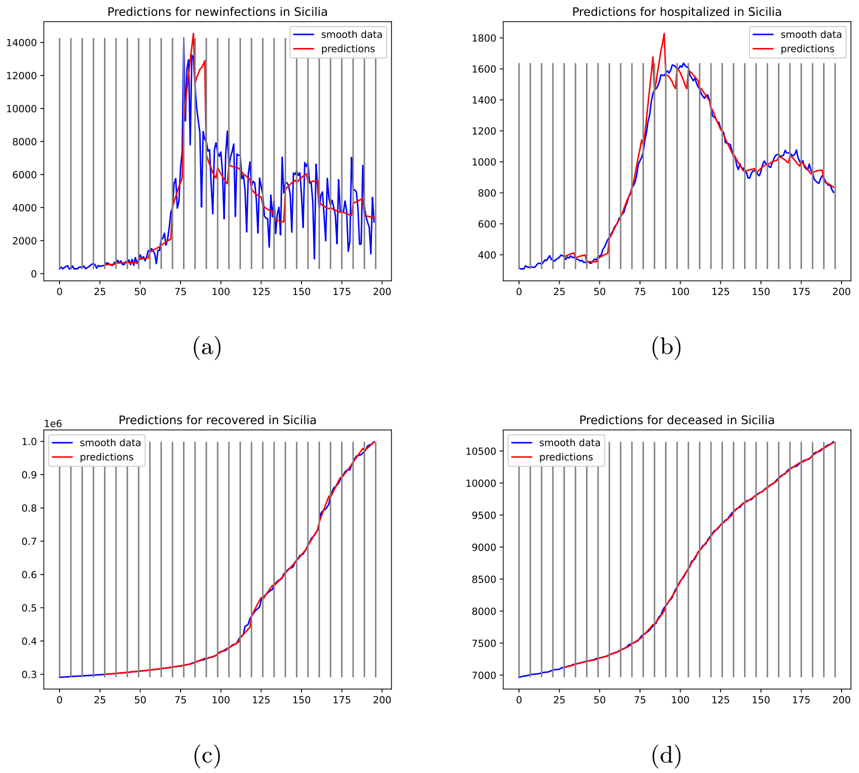
Figure 4: Predictions for: (a) New infections, (b) Hospitalized, (c) Recovered, (d) Deceased in Lombardia



(a) (b)

(c) (d)

Figure 5: Predictions for: (a) New infections, (b) Hospitalized, (c) Recovered, (d) Deceased in Lazio



(a)

(b)

(c)

(d)

Figure 6: Predictions for: (a) New infections, (b) Hospitalized, (c) Recovered, (d) Deceased in Sicilia

4. Conclusions

We constructed our model using a recurrent neural network (RNN) and the results were very satisfying taking into account the relatively low complexity of the network and the few minutes needed for the training of it. In particular, given the data from the previous 28 days, the network is able to predict the values for the next 7 days with a good amount of accuracy.
A further step in the development of the model could be, for example, a post-process of the *New infections* results in order to take into account the cases drop on Sunday.

References

[1] Presidenza del Consiglio dei Ministri. Dati covid-19 italia, 2022. `https://github.com/pcm-dpc/COVID-19`.