

# **DST - Final Project**

Ziqiu Zhou (**3635588**), Christoph Bender (**4012810**)

March 5, 2022

# 1 Summary LSTM [3]

The invention of LSTM was motivated by the regularization of recurrent neural networks (RNNs). In addition to inputs  $\mathbf{i}_t \in \mathbb{R}^{d_i}$ , RNNs use also loops in order to include informations from previous hidden states  $\mathbf{h}_{t'}$  (where  $t' < t$ ) in the calculation of the current state  $\mathbf{h}_t \in \mathbb{R}^{d_h}$  at time  $t$ . The Elman network [1] is for example defined by <sup>1</sup>: However

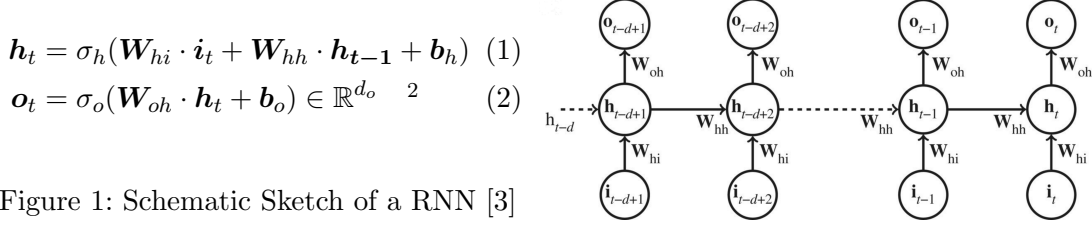


Figure 1: Schematic Sketch of a RNN [3]

RNNs struggle to recognize long-term dependencies and furthermore the gradient can vanish or explode, which also leads to problems. Because of this one uses gates. The corresponding model is called LSTM (Long Short-Term Memory). This is characterised by the following formula:

$$\begin{aligned} \mathbf{g}_t^f &= \sigma_f(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{i}_t] + \mathbf{b}_f) & \mathbf{g}_t^i &= \sigma_i(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{i}_t] + \mathbf{b}_i) \\ \tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{i}_t] + \mathbf{b}_C) & \mathbf{C}_t &= \mathbf{g}_t^f \cdot \mathbf{C}_{t-1} + \mathbf{g}_t^i \cdot \tilde{\mathbf{C}}_t \\ \mathbf{g}_t^o &= \sigma_h(\mathbf{W}_h \cdot [\mathbf{h}_{t-1}, \mathbf{i}_t] + \mathbf{b}_h) & \mathbf{h}_t &= \mathbf{g}_t^o \cdot \tanh(\mathbf{C}_t) \end{aligned} \quad (3)$$

Whereby  $\mathbf{g}_\bullet \in \mathbb{R}^{d_h \times (d_h + d_i)}$  represents the gate signal and  $\mathbf{C}_t \in \mathbb{R}^{d_h}$  the so-called cell state, which is jointly with the hidden state referred as "LSTM states". To map the hidden state to the wanted output space one also uses an additional fully connected layer  $\mathbf{W}_{oh}$ :

$$\mathbf{o}_t = \mathbf{W}_{oh} \cdot \mathbf{h}_t = f^w(\mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{C}_{t-1}) \approx F^w(\mathbf{z}_t, \mathbf{z}_{t-1}, \dots, \mathbf{z}_{t-d+1}) \quad (4)$$

The so-called LSTM cell-function  $f^w$  can be rewritten by iterative repetition as  $F^w$ , where  $w$  includes all trainable parameters. In the last step one uses the assumption that  $d$ -time steps are sufficient to compute the current output and thus  $\mathbf{h}_{t-d}, \mathbf{C}_{t-d}$  can be omitted. The model is data-driven, one has not to incorporate prior knowledge (like underlying equations) in the system necessarily. Goal of the LSTM is to predict the state derivative  $\dot{\mathbf{z}}_t$  using a short time memory of the  $d$  previous states  $\mathbf{z}_{t:t-d+1}$ . Therefore, the loss  $\mathcal{L}$  shall be minimized by searching for the best parameters  $w^*$ :

$$w^* = \arg \min_w \mathcal{L}(\{\mathbf{z}_{1:T}, \dot{\mathbf{z}}_{1:T}\}, w) = \arg \min_w \frac{1}{T-d+1} \sum_{t=d}^T \|F^w(\mathbf{z}_{t:t-d+1}) - \dot{\mathbf{z}}_t\|^2 \quad (5)$$

<sup>1</sup>see also [https://en.wikipedia.org/wiki/Recurrent\\_neural\\_network#Elman\\_networks\\_and\\_Jordan\\_networks](https://en.wikipedia.org/wiki/Recurrent_neural_network#Elman_networks_and_Jordan_networks)

<sup>2</sup>where  $\sigma_\bullet$  are activation functions,  $\mathbf{W}_\bullet$  weight matrices and  $\mathbf{b}_\bullet$  bias summands.

<sup>3</sup>where in the following  $\mathbf{z}_t$  is used as the input and describes the system time series.

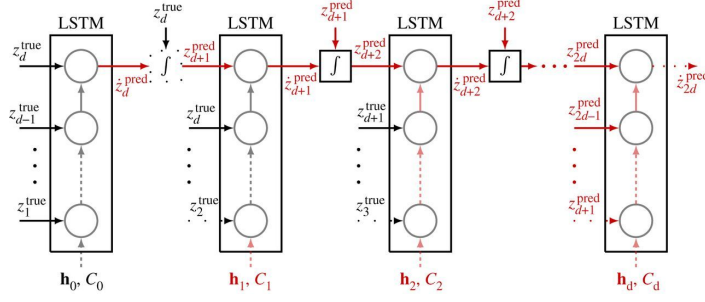


Figure 2: LSTM Model[3]  
Predictions are marked with red arrows, while black arrows indicate the short-term memory. The known  $z_{1:d}^{\text{true}}$  are used to predict  $z_d^{\text{predict}}$ , which can be integrated in order to get the next state  $z_{d+1}^{\text{predict}}$ .

The group of Vlachas trained the LSTM by using backpropagation. For this, a mini-batch optimization with the Adam method [2] and an initial learning rate  $\epsilon_{\text{init}} = 10^{-4}$  was applied. The initialization of the weights is based on the method of Xavier. One problem the group had to face was to fine tune the dimension of the hidden stated  $d_h$ . They observed that a small  $d_h$  shortened the ability to fit to the time series well, but a big  $d_h$  increased the risk of overfitting and the computation time.

In the paper they compared the LSTM Model with the GPR (Gaussian process regression) and MSM (Mean Stochastic Model). Three applications (The Lorenz 96 sytem, the Kuromoto-Sivashinsky equation an a barotropic climate model) were considered in order to evaluate the predictive accuracy of the models. They showed that in all cases the LSTM outperformed the other two models, i.e. the LSTM model was able to catpure the local dynamics more efficient.

## 2 Power spectrum metric

The second task of the project was to implement the power-spectrum correlation (PSC). This features two hyperparameters: Namely

- the smoothing factor  $\sigma$ , which is the width of the Gaussian kernel smoothing the spectrum
- and the cutoff, which excludes all frequencies above the given frequency threshold from the calculation of the power spectrum correlation.

The code snippets of *psc.py* was added in *global\_utils.py* and executed in *rnn\_statefull.py*. Furthermore the plotting functions are added in the *plotting\_utils.py* file. In order to ensure that the model draws a random initial condition and then generates a time series of length T. We sampled random indices in the data generation and for each prediction the model takes one of these random integers during testing. And starts to predict the time series starting at the chosen starting point of the whole test series.

To illustrate the LSTM model and also to make sure whether the LSTM model is working, we first apply the model on a very simple sinusoidal dataset before applying it on the Lorenz datasets (see section 3). For this we generate a time series by computing:

$$z_t = A \cdot \sin(2\pi ft + \phi_0) + c \quad (6)$$

### 3 Testing on Lorenz-datasets

#### 3.1 Lorenz63

#### 3.2 Lorenz96

### References

- [1] Jeffrey L. Elman. “Finding Structure in Time”. In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: [https://doi.org/10.1207/s15516709cog1402\\_1](https://doi.org/10.1207/s15516709cog1402_1). eprint: [https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402\\_1](https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1). URL: [https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402\\_1](https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1).
- [2] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [3] Pantelis R. Vlachas et al. “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213 (2018), p. 20170844. DOI: 10.1098/rspa.2017.0844. eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2017.0844>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2017.0844>.