DST - Final Project

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

March 2, 2022

1 Summary LSTM [3]

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

$$\boldsymbol{h}_t = \sigma_h(\boldsymbol{W}_{hi} \cdot \boldsymbol{i}_t + \boldsymbol{W}_{hh} \cdot \boldsymbol{h_{t-1}} + \boldsymbol{b}_h) \quad (1)$$

$$\boldsymbol{o}_t = \sigma_o(\boldsymbol{W}_{oh} \cdot \boldsymbol{h}_t + \boldsymbol{b}_o) \in \mathbb{R}^{d_o} \quad [2] \quad (2)$$

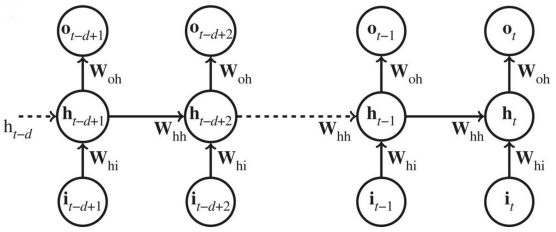

Figure 1: Schematic Sketch of a RNN [3]

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:

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

Whereby $\boldsymbol{g}_\bullet \in \mathbb{R}^{d_h \times (d_h + d_i)}$ represents the gate signal and $\boldsymbol{C}_t \in \mathbb{R}^{d_h}$ 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 $\boldsymbol{W}_{oh}$:

$$\boldsymbol{o}_t = \boldsymbol{W}_{oh} \cdot \boldsymbol{h}_t = f^w(\boldsymbol{z}_t, \boldsymbol{h}_{t-1}, \boldsymbol{C}_{t-1}) \approx F^w(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}, \ldots, \boldsymbol{z}_{t-d+1}) \quad [3] \quad (4)$$

The so-called LSTM cell-function f^w can be rewritten by iterative repition 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 $\boldsymbol{h}_{t-d}, \boldsymbol{C}_{t-d}$ can be omitted. The model is data-driven, one has not to incoperate prior knowledge (like underlying equations) in the system necessarily. Goal of the LSTM is to predict the state derivative $\dot{\boldsymbol{z}}_t$ using a short time memory of the d previous states $\boldsymbol{z}_{t:t-d+1}$. Therefore, the loss $\mathcal{L}$ shall be minimized by searching for the best parameters w^*:

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

[1] see also `https://en.wikipedia.org/wiki/Recurrent_neural_network#Elman_networks_and_Jordan_networks`

[2] where $\sigma_\bullet$ are activation functions, $\boldsymbol{W}_{*\bullet}$ weight matrices and $\boldsymbol{b}_\bullet$ bias summands.

[3] where in the following $\boldsymbol{z}_t$ is used as the input and describes the system time series.

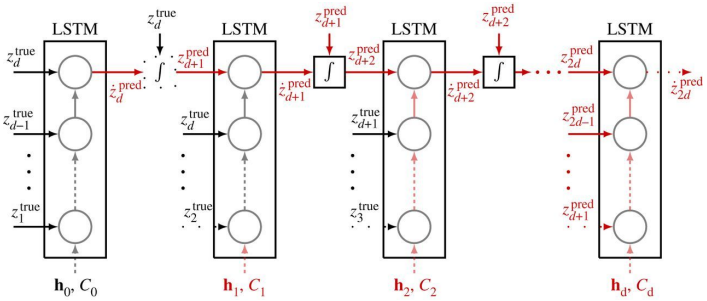

Figure 2: LSTM Model[3] Predictions are marked with red arrows, while black arrows indicate the short-term memory. The known $\boldsymbol{z}_{1:d}^{\text{true}}$ are used to predict $\dot{\boldsymbol{z}}_d^{\text{predict}}$, which can be integrated in order to get the next state $\boldsymbol{z}_{d+1}^{\text{predict}}$.

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 $\epsilon_{init} = 10^{-4}$) was applied. The inizialization 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 Results

What do you think about the idea to shortly present the results of task 3 here, i.e. saying the sigma and the cutoff frequency and maybe some training graphics ...

2.1 Lorenz63

2.2 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. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1. URL: 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.