Challenges in chaotic time series forecasting

Maximilian Hornung

April 25, 2019

Abstract

Predicting the future development of time series is of interest in different areas of computational biology. The time series in biological models exhibit challenging characteristics such as chaotic or stochastic behavior. In this report, time series prediction is done on different kind of biological time series using deep neural networks. In our evaluation, we identify challenges and limitations of this approach, and compare different architectures of deep neural networks with regard to their performance.

1 Introduction

In various kind of diseases, it is important to choose the correct treatment at the current time point. Since the human body is too complex to be described directly using mathematics, special features are modeled with mathematical systems such as the Mackey-Glass equations for respiratory and hematopoietic diseases [MG77]. These models make use of available data about previous information in order to reduce the uncertainty under which a decision is done. This is especially important when the consequences of a decision are severe, for example the decision of breast cancer treatment after radiation. This particular problem can be addressed by using a stochastic differential equation, as done by Oroji *et al* [OOY16].

In different kinds of biological models, chaotic behavior is observed. The Mackey-Glass equations possess chaotic behavior for certain choices of their parameters, as shown by Fischer [Far82]. That means that even small measurement errors of the initial state lead to exponentially increasing errors of the time series prediction. Because of that, it is important that the time series prediction should be as robust as possible against noise.

This gets imporant in particular because another characterstic of dealing with biological data is the presence of noise. In this report, we show that different kinds of noise assumptions can impact the results of time series prediction. We evaluate all our investigated problems with regard to the popular *i.i.d.* random noise as well as noise from a discretized Wiener process, i.e. memorizing random noise. Since mathematical models of biological systems can not describe their characteristics perfect and error-free, it is assumed that the latter type of noise is more realistic. But unfortunately this kind of noise increases the difficulty of time series prediction, even for simple mathematical functions, as seen in Section 3.1.

The remaining report is structured as follows. First, we perform proper time series prediction on the sinus function and show how different noise models impact the prediction ability of our network models. After that, we show how different levels of chaos impact the possibility to forecast the development of time series at the example of the *Mackey Glass* time series. We show how the stochasticity of noise impactss the possibility to predict deterministic chaotic time series at the example of the *Mackey Glass* time series. Last but not least, we report fundamental limitations of neural networks to approximate stochastic time series at the example of biological oscillators.

2 Methods

In our analysis, we use three different architectures and neural networks and optimize their hyperparameters for the respective use case. Since Hornik *et al.* have shown that feedforward neural networks are universal function approximators [HSW89], we evaluate this architecture with varying number of hidden nodes.

After that, we investigate the strength of improvement of using a recurrent neural network. This type of neural network has already been applied to time series prediction [CMA94], but suffers from the vanishing gradient problem when capturing long-term dependencies in a sequence. Because of that, we decide to evaluate only the Long Short-Term Memory (**LSTM**) network architecture [HS97], which was successfully applied in various time series prediction tasks like anomaly detection [MVSA15], stock price [FK18] and protein disorder prediction [HYPZ16].

In the last years, convolutional neural networks (**CNN**s) have improved the results in image classification [KSH12] and other computer vision tasks. A **CNN** learns features from the data in a hierarchical way, for example combining pixels to edges, edges to more complex forms etc. until a high-level classification can be done. The large success in computer vision has inspired researchears in time series prediction to also apply **CNN**s [CCC16, BBO17], so we also evaluate this architecture in our analysis. The same way as images are composed of hierarchical features (e.g. a face consisting of eyes, that consist of certain edges etc.) we assume that similar hierarchies of features can be found in time series data.

All implementation is done using the `Python` programming language in version 3.6.7. The time series data is created and loaded in the `numpy` framework in version 1.16.1. In order to train the neural networks both fast and elegant, we use the `keras` (version 2.2.4) with the `tensorflow` backend in version 1.13.1. The reason for this choice is the tight integration between `keras` and `numpy` that simplifies and increases the speed of our software development. Before running the experiments, the random number generator of `numpy` is set to the seed 0 to ensure reproducibility of our results.

3 Evaluation

3.1 Time series forecasting of periodic functions

In the first part of the analysis, we analyze the capability of different neural network architectures to perform time series prediction on the periodic sinus function. This task can be considered simple, because the different parts of the sine wave occur multiple times in the data, i.e. the function is periodic. It is therefore interesting how robust the network architectures are with regard to different kinds of noise.

In this section, we provide information about the last 10 time sequence points to the network and want to predict the next point, as given in Equation 1. That is, we want to estimate $x(t)$ with a function $F(x(t), x(t-1), \cdots, x(t-9))$ that is computed using a neural network.

$$\hat{x}(t+1) \approx F(x(t), x(t-1), \cdots, x(t-9)) \tag{1}$$

We consider two different types of noise. First, we consider *i.i.d.* gaussian noise which is added to the underlying function $f(t)$ as seen in Equation 2. The other type of noise is a discretized Wiener process i.e. a memorizing type of noise that is shown in Equation 3. For more details about Wiener processes, refer to Schilling *et al* [SP14]

$$\forall t: \quad x(t) = f(t) + y, \quad y \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

$$\forall t: x(t) = f(t) + y(t), \quad y(t) = y(t-1) + y \sim \mathcal{N}(0, \sigma^2), \quad y(0) = 0 \tag{3}$$

In Figure 1, we see that a densely connected neural network with one hidden layer of 10 nodes is capable of performing the time series prediction. Even if we apply *i.i.d.* gaussian noise with standard deviation $\sigma = 0.1$, the network can still be trained to fit the data correctly. But if the noise is not applied independently in each timestep, the convergence takes more time steps as pointed out in Figure 2. This effect is especially strong for the *LSTM* architecture, where even 100 epochs are not enough for the validation loss to converge, compared to the other approaches that need only 20 epochs.

Since the noise we apply on the data has a standard deviation $\sigma = 0.1$, it is obvious that a perfect time series prediction is not be possible. We expect that the minimum possible validation loss is as high as the variance (σ^2), and can reproduce that in all neural network architectures as shown in Table 1. The validation loss of a perfect neural network would therefore be $\sigma^2 = 0.01$, which is achieved by all architectures for *i.i.d.* noise. Only the LSTM architecture has worse performance for the memorizing noise. We assume that this is due to the slower learning of that architecture compared to the others when confronted with

Architecture	Noisefree data	*i.i.d.* noise	memorizing noise
MLP	0.0021	0.0249	0.0174
LSTM	0.0021	0.0127	0.5554
CNN	0.0021	0.0136	0.0234

Table 1: Validation loss of different neural network architectures for time series prediction on the sinus function stated in RMSE.

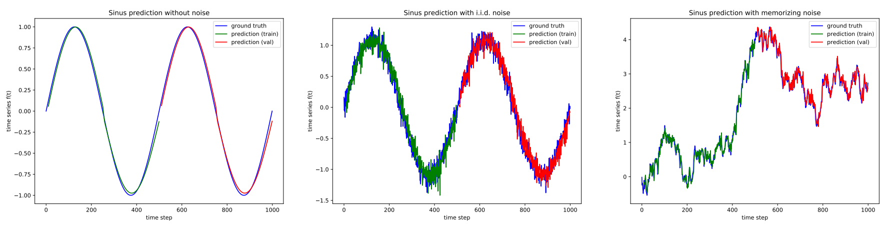

Figure 1: Impact of different kinds of noise on the time series prediction using a feedforward neural network with one hidden layer. Number of hidden units stays 10 in all simulations. The noise is normal distributed with $\sigma = 0.1$.

memorizing noise. One possible reason for this observation is that the LSTM is trying to capture temporal dependencies between points in the time series, which is confused by the temporal dependencies induced by the memorizing noise.

3.2 Mackey Glass time series forecasting

In order to model diseases related to dynamic respiratory and hematopoietic diseases, Mackey *et al.* proposed the mackey-glass equations, a kind of first-order nonlinear differential delay equations to model the number of white blood cells over time [MG77]. The solutions to these equations given in Equation 4 exhibit chaotic behavior under certain conditions [Far82]. For fixed parameters $a = 0.2$, $b = 0.1$ and $c = 10$ this system has a stable fixed point attractor for

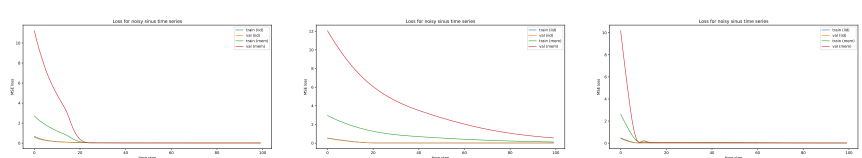

Figure 2: MSE loss over the number of trained epochs on training and on validation dataset. On the left, the classical feedforward network was used. The middle plot shows the performance of the LSTM and the right one plots the loss values of the convolutional network.

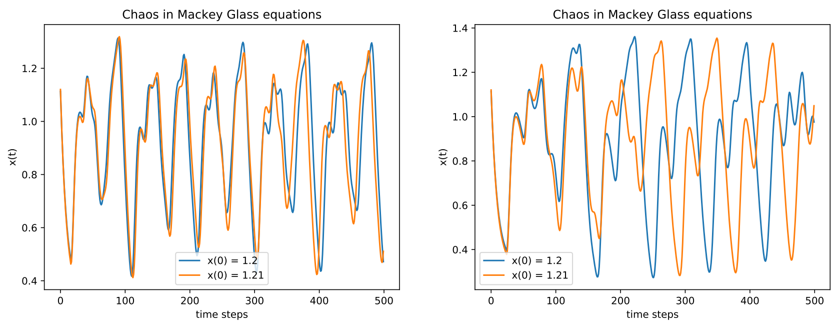

Figure 3: Influence of time delay parameter τ on the chaotic behavior of the Mackey Glass equation. The left side of the figure uses $\tau = 17$, the right side $\tau = 25$.

$\tau < 4.53$. With increasing delay time, the system gets less stable. For delay times of $4.53 < \tau < 13.3$ there is a limit cycle attractor whose period raises for $13.3 \leq \tau \leq 16.8$. For delay time $\tau > 16.8$ the system shows chaotic behavior.

$$\frac{dx}{dt} = \frac{a \cdot x(t-\tau)}{1 + x(t-\tau)^c} - b \cdot x(t) \tag{4}$$

By using the Euler method to discretize the Mackey Glass time series, we can see that the chaotic behavior gets stronger for increasing $\tau > 16.8$. In Figure 3, we simulate the solution of the Mackey Glass time series for slightly different initial conditions. It is clearly visible that for $\tau = 17$, the time series diverges slower than for $\tau = 25$.

The first approach to predict the short-time behavior of chaotic time series was done by Farmer *et al.* who proposed a `local approximation` technique [FS87]. After improval of predictions using support vector machines by Müller *et al.* [MSR+97], the focus in research shifted towards artifical neural networks which enable even better predictions. Two of the latest developments are the usage of Wavelet Networks [AZ13] and particle swarm optimization [LCSL+16].

$$x(t+1) = x(t) + \frac{\beta x(t-\tau)}{1 + x^n(t-\tau)} - \gamma x(t) \tag{5}$$

In accordance with the approach used by Caraballo *et al.* [LCSL+16], we predict $x(t+6)$ based on the information of the time series points in $x(t)$, $x(t-6)$, $x(t-12)$, and $x(t-18)$. We investigate how an increase in chaotic behavior impacts the prediction capability of the different neural network architectures. The results for the feedforward neural network are depicted in Figure 4. For this plot, a 2-layer CNN architecture was used, incoroporating "same" padding and a kernel size of 3. The first 3000 time points of the time series are used for training, the 1000 following points for validation.

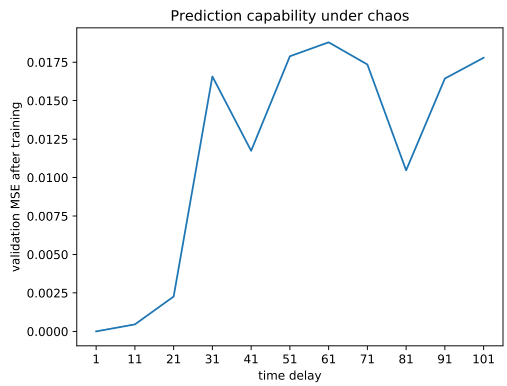

Figure 4: Validation error of the convolutional neural network for different time delay values τ of the Mackey Glass equation. The error increases strongly for values of $\tau > 20$, indicating a strong chaotic behavior from this point.

Method	$RMSE_{x(t+6)}$
Linear model	0.5503
Cascade correclation NN	0.0624
RBFNN	0.0114
ANN + PSO [LCSL+16]	0.0053

Table 2: Mackey Glass time series forecasting results using different neural network architectures.

It can be seen that for non-chaotic solutions of the macke-glass time series the model is able to predict precisely, and that for values of $\tau > 20$ the chaotic behavior is strong enough to impact the prediction. We explain the low error for $\tau = 21$ in the ability of the network to compensate for the amount of chaos up to this point.

In the next step, we compare our feedforward neural network and our LSTM architecture with the results stated by Caraballo *et al.* [LCSL+16] in Table 2. The results are based on the root mean squared error (RMSE), computed in Equation 6.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{6}$$

After selecting the best hyperparameters for our architectures, we choose to use a **CNN** architecture for Mackey Glass time series prediction, because with this architecture we achieve lowest RMSE. **TODO: clarify architecture**.

3.3 Biological oscillator time series forecasting

The next section deals with time series forecasting of biological oscillators, as described by Novak *et al* [NT08]. For different aspects of cell physiology, e.g. the DNA synthesis, this kind of oscillators can be observed. Novak *et al.* show that oscillations in simple ODEs can be produced by incorporating an explicit time delay [NT08], similar to the Mackey Glass equations [MG77].

Recently, Strömstedt *et al.* analyzed the capability of neural network architectures to model stochastic time series [SSH18] at the example of time series produced according to the characteristics of Novak *et al* [NT08]. In this section, we reproduce their results for the LSTM architecture and state fundamental limits for neural networks to approximate stochastic time series.

The time series data is generated using `gillespy` [ADHP16]. Using this framework, a simple stochastic reaction system with 2 states is set up. The protein X is produced more the less of the other protein Y exists. X and Y are both continuously reduced but Y gets reduced stronger for smaller values of Y. A set of 7 parameters is used to parametrize the system.

The motivation for using neural netowrks to generate time series is the computational expensiveness of analytical solutions like the Gillespie algorithm [Gil77]. On experiments on our hardware (see Table 3 for details) we need more than 6 seconds to simulate the biological oscillator with two states. This makes grid search approaches on the large number of parameter combinations intractable. Using the 2-layer LSTM approach predictions for short time series can be done in less than 0.2 seconds on our hardware.

The goal here is to train a LSTM based model to convert a set of input parameters into a time series. The first LSTM layer takes 7 input values and transforms them into n sequences of 7 values, with n being the desired length of the output sequence. The second LSTM layer takes this result and converts it to a sequence of length n.

The analysis of the resulting time series as depicted in Figure 5 shows the general limitation of using a deterministic neural network for predicting a stochastic time series. For one possible of parameters for the biological oscillator, the analytical solution was computed under 100 (stochastic) trajectories. After that, the network was trained to predict the time series based on the (unchanged) parameters. We see that the short-term predictions are accurate and have the same magnitude as the ground truth signal – but the more time steps are simulated, the larger is the difference between the ground truth trajectories due to stochasticity. In other words, the neural network predicts the mean of the signals and the amplitude decreases due to the increasing variance of the signal over time. It should be noted that the trends in the time series are still recognized, just with a smaller amplitude.

Processor	Intel(R) Core i5-7200U CPU @ 2.50GHz
RAM	7859 MB

Table 3: Hardware configuration used in experiments.

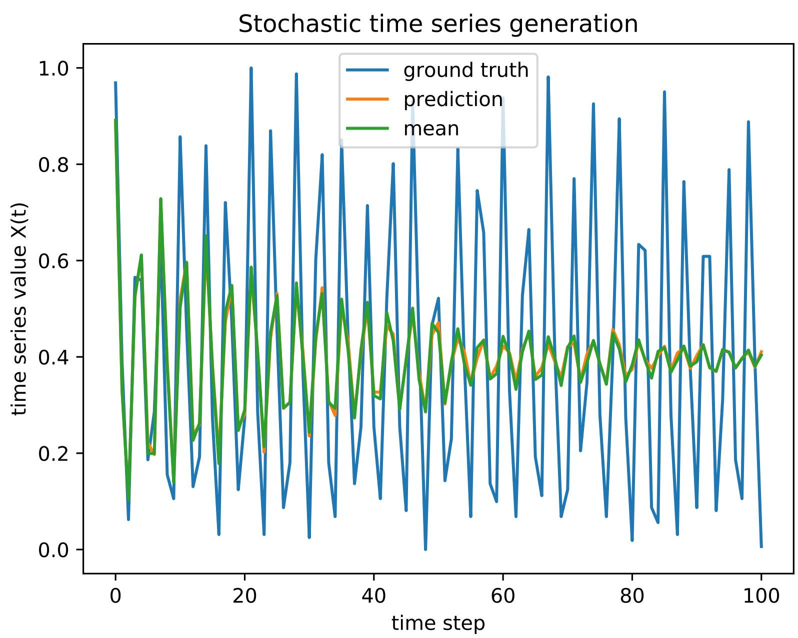

Figure 5: Prediction of stochastic time series using deterministic **LSTM** network. Sampled ground truth data are unseen for the network. Mean computed on all stochastic trajectories.

4 Conclusion

In this report, we investigate how chaotic time series can be approximated using neural networks. It is possible to improve the forecasting results by advanced architectures. Especially the convolutional neural networks outperform the feed-forward neural network with one hidden layer, although theoretically all classes are capable of approximating any continuous function. We assumed that the hierarchical features in time series can be captured better using a *LSTM* by capturing time dependencies or a *CNN* for extracting hierarchical features, but the results indicate that the convolutional neural networks also show superior performance in this area. Unfortunately, it is also shown that neural networks can only be used to generate stochastic time series in a limited way. Since the prediction operation is deterministic, the possibility of predicting the stochastic time series declines with the length of the prediction. Only the general trends can be seen from the neural network, the amplitude declines below the level of the original time series.

References

[ADHP16] John H Abel, Brian Drawert, Andreas Hellander, and Linda R Petzold. Gillespy: a python package for stochastic model building and simulation. *IEEE life sciences letters*, 2(3):35–38, 2016.

[AZ13] Antonios K Alexandridis and Achilleas D Zapranis. Wavelet neural networks: A practical guide. *Neural Networks*, 42:1–27, 2013.

[BBO17] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

[CCC16] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

[CMA94] Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

[Far82] J Doyne Farmer. Chaotic attractors of an infinite-dimensional dynamical system. *Physica D: Nonlinear Phenomena*, 4(3):366–393, 1982.

[FK18] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[FS87] J Doyne Farmer and John J Sidorowich. Predicting chaotic time series. *Physical review letters*, 59(8):845, 1987.

[Gil77] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[HYPZ16] Jack Hanson, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. *Bioinformatics*, 33(5):685–692, 2016.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LCSL+16] CH López-Caraballo, I Salfate, JA Lazzús, P Rojas, M Rivera, and L Palma-Chilla. Mackey-glass noisy chaotic time series prediction by a swarm-optimized neural network. In *Journal of Physics: Conference Series*, volume 720, page 012002. IOP Publishing, 2016.

[MG77] Michael C Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[MSR+97] K-R Müller, Alexander J Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *International Conference on Artificial Neural Networks*, pages 999–1004. Springer, 1997.

[MVSA15] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain, 2015.

[NT08] Béla Novák and John J Tyson. Design principles of biochemical oscillators. *Nature reviews Molecular cell biology*, 9(12):981, 2008.

[OOY16] Amin Oroji, Mohd Omar, and Shantia Yarahmadian. An ito stochastic differential equations model for the dynamics of the mcf-7 breast cancer cell line treated by radiotherapy. *Journal of theoretical biology*, 407:128–137, 2016.

[SP14] René L Schilling and Lothar Partzsch. *Brownian motion: an introduction to stochastic processes*. Walter de Gruyter GmbH & Co KG, 2014.

[SSH18] Adam Lindell Simon Strömstedt Hallberg. Black box time series modeling, 2018.