Explicit memory tuning for reservoir computing optimization

Lina Jaurigue, Kathy Lüdge
Institut für Physik, Technische Universität Ilmenau, Ilmenau, Germany

Summary. Reservoir computing requires two main ingredients: nonlinearity and memory. In hardware-based systems these contributions are usually intertwined and can be difficult to optimize. We show that by explicitly including task relevant memory timescales, the performance can be efficiently tuned. We discuss and compare three ways to tune this memory in reservoir computing systems, output-, input-, and internal-delay.

Reservoir computing (RC) has gained substantial interest over the last years due to its simple training procedure and its potential for hardware implementation. However, to achieve the desired level of performance extensive hyperparameter optimization is usually necessary. In an experimental setting the optimization of parameters of the physical reservoir can be a time consuming process and some parameters are inaccessible. In this contribution we will show that by explicitly including task relevant time scales in the reservoir computing system the performance can be optimized and thereby the need for hyperparameter optimization can be reduced. We refer to explicitly tunable memory when the parameter of interest directly influences the memory of the system. For example, in the case of a time-delayed reservoir, the delay-time directly determines when past states of the system reenter the nonlinear node and hence on which timescales the system remembers past inputs. In contrast, in a random recurrent neural network, the memory is influenced more indirectly via parameters such as the spectral radius.

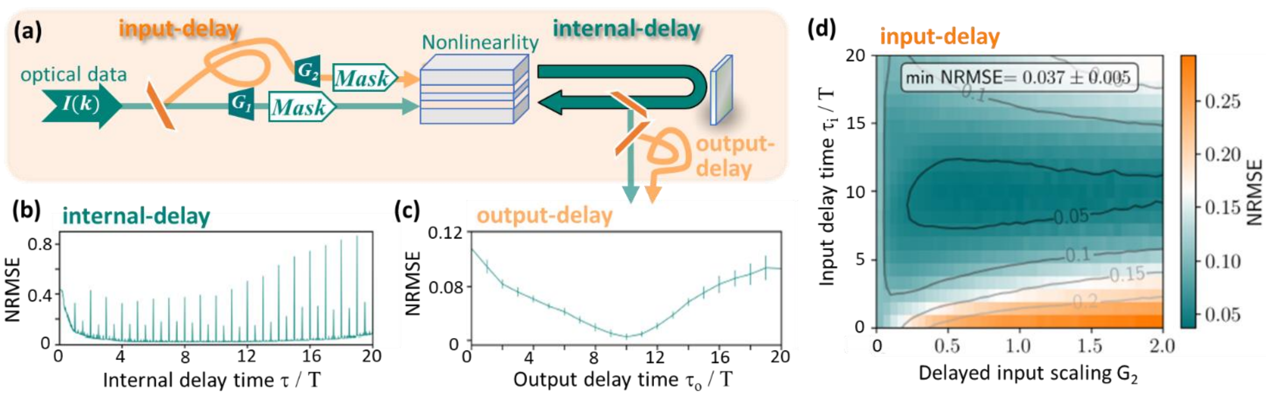

Figure 1: (a) Sketch of a photonic reservoir computing architecture that is suitable to include input-delay, output-delay or internal-delay. The 10-step ahead prediction performance for the Mackey-Glass equation of this RC system (with parameters as in [1]) is presented as a function of the internal delay (b) and the output delay (c). The effect of the input delay in (d) shows also the impact of the input scaling parameter in a 2D scan.

There are various ways in which the memory of a reservoir can be tuned or augmented. Here we will discuss two methods which can be applied to any reservoir computing setup, and in the case of a time-delayed reservoir we will also compare these methods with direct tuning of the internal delay time. Fig.1(a) shows a schematic diagram of how the input- and output-delays could be included in a photonic setup.

The first *input-delay* approach is to artificially include the memory in the input layer. For a time-series prediction task this would mean additionally inputting a delayed version of the input time-series with a separate mask and a separate input scaling (Fig1a). As an exemplary reservoir we simulate a delayed map (Eq.(1)) describing a semiconductor optical amplifier with a nonlinearity function $G(x)$, coupled to a feedback cavity with delay τ [1,2]. The term $J(k)$ includes both the normal and the delayed input as sketched in Fig.1a. The effect of the input-delay on the 10-step ahead prediction performance of the Mackey-Glass equation is depicted in Fig.1d using the normalized root mean square error (NRMSE) as a performance measure. The prediction error shows a clear optimum at a specific delay (approx. 10 clock cycles) while the input scaling, one hyperparameter, has a much smaller impact. We note that including multiple input values from a time-series for prediction tasks is a common practice in machine-learning and statistical modelling approaches, and there, deciding which past inputs should be included is then part of the feature selection process. In the reservoir computing paradigm, however, we ideally want to decrease the training and optimization steps that are necessary, which is why we propose that it is enough to include only one delayed input. This introduces only two new hyperparameters, the input-delay and the delayed-input scaling, both of which are experimentally easily tunable.

$$\mathrm{x}_{\mathrm{out}}(k) = G[K\mathrm{x}_{\mathrm{out}}(k-\tau) + J(k)] \tag{1}$$

The second *output-delay* approach is to include the memory in the output layer by constructing the state matrix out of the current states $\mathrm{x}_{\mathrm{out}}(k)$ of the reservoir and the states of the reservoir some delayed steps into the past, i.e. $\mathrm{x}_{\mathrm{out}}(k-\tau_{\mathrm{o}})$. Fig.1c shows a scan of the performance for different values of the output-delay. A clear optimum, again at the characteristic time of 10 clock-cycles, can be seen and a similar performance is achieved for the two cases of adding memory to the RC setup. Recently there have been various versions of the delayed-output approach but always implemented with a multitude of delayed output states [3,4,5,6]. Here, we propose that using only one fixed output-delay is sufficient, thus introducing only one new optimization parameter.

Lastly, we compare the effect of the previous two delay-schemes to the most common scheme of *feedback-delay* time. In delay-based RC this value is commonly chosen equal or close to the input clock time *T*. As can be seen in Fig.1b, tuning the feedback-delay time leads to improved performance over a wide range of delay times, with the minimum near 10 times the input clock time. Due to the nonlinear interaction of the inputs within the internal feedback loop, the performance is worsened at resonances between the feedback delay time τ and the input clock time, as described in [7].

As a conclusion, we show that by explicitly including memory on a timescale relevant for the Mackey-Glass task, the performance of the RC system is improved. The absolute value of the delay needed for optimization strongly depends on the task, which is why an easy tuning possibility is needed in order to realize a universal RC setup for many different tasks. All of our three schemes can offer this possibility. We also want to mention that the method of externally adding an explicit memory is not limited to the RC paradigm but can be applied to other machine learning agents like feedforward networks or statistical models.

References

[1] L. C. Jaurigue, E. Robertson, J. Wolters and K. Lüdge, "Reservoir Computing with Delayed Input for Fast and Easy Optimization", *Entropy* **23**, p.1560, 2021.

[2] L. C. Jaurigue, E. Robertson, J. Wolters and K. Lüdge, "Photonic reservoir computing with non-linear memory cells: interplay between topology, delay and delayed input", p.1220408, *SPIE proceedings ETAI*, 2022.

[3] B. A. Marquez, J. Suarez-Vargas, and B. J. Shastri, "Takens-inspired neuromorphic processor: A downsizing tool for random recurrent neural networks via feature extraction," *Phys. Rev. Research* **1**, p.033030, 2019.

[4] Y. Sakemi, K. Morino, T. Leleu, and K. Aihara, "Model-size reduction for reservoir computing by concatenating internal states through time," *Sci. Rep.* **10**, p. 21794, 2020.

[5] E. Del Frate, A. Shirin and F. Sorrentino, "Reservoir computing with random and optimized time-shifts", *Chaos* **3**, p.121103, 2021.

[6] T. L. Carroll and J. D. Hart, "Time Shifts to Reduce the Size of Reservoir Computers", *Chaos* **32**, p.083122, 2022.

[7] F. Stelzer, A. Röhm, K. Lüdge and S. Yanchuk, "Performance boost of time-delay reservoir computing by non-resonant clock cycle", *Neural Netw.* **124**, p.158, 2020.