Optimal Training of Echo State Networks via Scenario Optimization

Luca Bugliari Armenio, **Lorenzo Fagiano**, **Enrico Terzi**, **Marcello Farina**, and **Riccardo Scattolini**

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Via Ponzio 34/5, 20133, Milano, Italy.
E-mail: name.surname@polimi.it

Abstract: Echo State Networks (ESNs) are widely-used Recurrent Neural Networks. They are dynamical systems including, in state-space form, a nonlinear state equation and a linear output transformation. The common procedure to train ESNs is to randomly select the parameters of the state equation, and then to estimate those of the output equation via a standard least squares problem. Such a procedure is repeated for different instances of the random parameters characterizing the state equation, until satisfactory results are achieved. However, this trial-and-error procedure is not systematic and does not provide any guarantee about the optimality of the identification results. To solve this problem, we propose to complement the identification procedure of ESNs by applying results in scenario optimization. The resulting training procedure is theoretically sound and allows one to link precisely the number of identification instances to a guaranteed optimality bound on relevant performance indexes, such as the Root Mean Square error and the FIT index of the estimated model evaluated over a validation data-set. The proposed procedure is finally applied to the simulated model of a pH neutralization process: the obtained results confirm the validity of the approach.

Keywords: Echo State Networks, Deep Learning, Neural Network Training, Guaranteed Optimality, Scenario Optimization

1. INTRODUCTION

In recent years the control community is experiencing a renewed interest in data-driven identification/learning and control techniques (Tan et al. (2018); Deisenroth et al. (2011)), fostered by more and more powerful machine learning tools and algorithms (Pham and Liu (1995); Lee and Teng (2000); Bristow et al. (2006)). Indeed, the availability of large datasets enables effective modeling of complex dynamical systems (Kutz (2013)). Among the most employed tools, Neural Networks (NN) receive a particular attention for their flexibility in accomplishing identification tasks, spanning different classes of systems and problems (Sutskever et al. (2014); Haykin et al. (2009); Hinton et al. (2012)).

Several NN architectures have been proposed, among which Recurrent Neural Networks (RNN) are very promising due to their inherent capability to represent nonlinear dynamical systems. Different methods to train RNN are present in the literature (Jaeger (2002)), all sharing the same core idea of minimizing the prediction error by tuning the available degrees of freedom of the network, i.e. its internal weights. However, it is not rare that the training algorithm results in a tough nonlinear problem, to be solved in an iterative (and time-consuming) way (Pascanu et al. (2013)). In this paper, we focus on a particular class of RNN, i.e. the Echo State Networks (ESN) (Jaeger (2002)), that has been successfully applied in many fields, such as speech recognition (Pearlmutter (1989)), time-series prediction (Jaeger and Haas (2004)), reinforcement learning (Szita et al. (2006)), and language modeling (Tong et al. (2007)). In addition, ESN have recently been used for the design of Model Predictive Controllers with guaranteed stability properties (Bugliari Armenio et al. (2019)).

From a system theoretic standpoint, the structure of an ESN consists of (i) a nonlinear state equation, where a sigmoid function is applied to a linear combination of states, inputs, and outputs, and (ii) a linear output equation. The tuning of the ESN parameters usually consists of two phases. First, the parameters of the state equation are randomly generated; then, the output transformation parameters are computed solving a standard Least Squares (LS) problem. This procedure yields a dramatic reduction of the computational cost, however, the random choice of the state equation parameters naturally influences the final performance of the ESN, so that it is a common practice to repeatedly run the estimation procedure until a satisfactory result is achieved (Jaeger (2001)). As a consequence, no guarantees on the performance of the obtained ESN can be given, and suitable criteria to assess the quality of the trained network with respect to “ideal” results are not available.

In this paper, we solve this problem resorting to scenario optimization. Our contribution stems from a rather simple observation: the described tuning procedure for ESN can be seen as a scalar optimization problem, where one wants to find the best model according to a performance index (no matter how defined, as long as it is systematically derived for a given network) among an infinite number of possible networks, each one featuring a random part - the state equation parameters - that is sampled according to a known probability distribution. This problem can not be solved exactly, however it can be turned into a convex, finite-dimensional optimization problem by means of a sampling approach, where the chosen network is picked as the best one among a finite number N_δ of trained ones. This falls perfectly into the framework of scenario optimization theory (Campi et al. (2009); Calafiore and Campi (2005); Campi and Garatti (2018)), which provides a powerful result to certify the optimality of the chosen network with respect to a new one, obtained by sampling again from the same distribution and carrying out the LS identification phase. In particular, an upper bound on the probability that a newly trained network scores a better performance than the chosen one (*violation probability*) can be exactly computed on the basis of N_δ and, vice-versa, it is possible to choose N_δ in order to obtain a wanted violation probability. Based on this observation, in this paper we add the state equation parameters in the set of unknowns to be optimized, and solve the corresponding optimization problem in a random fashion, but with sound probabilistic optimality guarantees.

The proposed approach is finally tested on the problem of estimating the ESN model of a *pH* neutralization process (Hall and Seborg (1989b)), which represents a well recognized SISO non-linear benchmark. The computed optimality guarantees are also verified *a posteriori* over a large number of experiments, confirming the validity of the theoretical results.

Notation. Given a matrix A, we denote $A_{(i)}$ its i^{th} row and A' its transpose. $\|v\|$ is the 2-norm of vector v, $\otimes$ denotes the Kronecker product, $\mathbf{1}_{a,b}$ represents the matrix full of ones of dimensions a, b.

2. PRELIMINARIES ON ECHO STATE NETWORKS

The problem considered in this work is to identify the parameters of an ESN to reproduce the behavior of an unknown discrete-time plant, using K measured samples of its inputs $u_{sys} \in \mathbb{R}^{n_u}$ and outputs $y_{sys} \in \mathbb{R}^{n_y}$. The model structure is:

$$x(k+1) = \zeta(W_x x(k) + W_u u(k) + W_y y(k)) \tag{1a}$$

$$\phi(k+1) = u(k) \tag{1b}$$

$$y(k) = W_{out_1} x(k) + W_{out_2} \phi(k) \tag{1c}$$

where k is the discrete time index, $x \in \mathbb{R}^n$ and $\phi \in \mathbb{R}^{n_u}$ are the model states, u and y are the inputs and outputs, respectively, W_x, W_u, W_y, W_{out_1} and W_{out_2} are weight matrices of proper dimensions, and ζ is a generic sigmoid Lipschitz continuous function, typically chosen as $\tanh(\cdot)$ (Sohrab (2003)).

The properties of the ESN (1) have been studied in Bugliari Armenio et al. (2019) from a control perspective, where it has been proven that, if $\|W_x\| < 1$, the system is Incrementally Input-to-State Stable (δISS) with respect to inputs u and y (Bayer et al. (2013)). This guarantees that, running the system with the same inputs and different initial states, the transients associated to the initial conditions asymptotically vanish, thus enabling a consistent estimation procedure of the unknown parameter matrices W_{out_1} and W_{out_2}. As discussed in Kim (2005), this property is strictly related to the one of fading memory. In addition, the δISS property is fundamental to design state observers and predictive regulators with stability guarantees, see again Bugliari Armenio et al. (2019).

Training procedure for ESN

Inspired by Jaeger (2002), the standard training of the ESN model (1) proceeds according to the following steps:

(a) collect from the plant the, possibly normalized (see Jaeger (2001)), input and output sequences u_{sys} and y_{sys};
(b) define the system order $n + n_u$;
(c) generate a sparse matrix W_x, with elements sampled from a uniform distribution and such that $\|W_x\| < 1$;
(d) generate random matrices W_u and W_y of proper dimensions, with elements sampled from a uniform distribution;
(e) start from an arbitrary initial state $x(0)$ and run the state equation (1a) forced by u_{sys}, y_{sys}; collect the computed state values $x(k)$, $k = 1, \dots, K$, where K is the number of samples of the dataset;
(f) discard the first $K_0 < K$ points of u_{sys}, y_{sys}, x to remove the effects of the initial state (recall the δISS property guaranteed by the choice of W_x in step (c));
(g) store the values of $(x(k), u_{sys}(k-1))$ and $y_{sys}(k)$ for $k \geq K_0$ into matrices Φ and Y_{sys} representing the output transformation (1c) in vector form;
(h) solve the Least Squares problem $\forall i = 1, \dots, n_y$

$$\min_{W_{out_{(i)}}} \|Y_{sys,i} - \Phi W'_{out_{(i)}}\|^2,$$

where $W_{out} = [W_{out_1} \quad W_{out_2}]$ and $Y_{sys,i}$ is the vector collecting all the samples pertaining to the i^{th} output of the system.

Letting $\bar{K} = K - K_0$ and $Y = [y(K_0), y(K_0+1), \dots, y(K)]'$ the predicted output of the model, the quality of the estimated model is usually evaluated with a chosen performance index, such as the root mean-square error (RMSE):

$$RMSE = \sqrt{\frac{1}{\bar{K}} \|Y_{sys} - Y\|^2} \tag{2}$$

Another common quality index is the FIT value:

$$FIT = 100 \cdot \left(1 - \frac{\|Y_{sys} - Y\|}{\|Y_{sys} - \mathbf{1}_{\bar{K},1} \otimes \bar{y}\|}\right) \in (-\infty, 100] \tag{3}$$

where $\bar{y}$ is the mean of the output of the system. As customary in identification (Ljung (1987)), the error indexes (2) and (3) should be computed over a validation dataset to actually assess the model performances. Given the definitions (2) and (3), the larger the FIT or the smaller the $RMSE$, the better is the identified model quality.

Note however that, due to the adopted training procedure, the computed value W_{out} of the output transformation parameters is itself a random variable, since it is a function of the random variables W_x, W_u, W_y. A sensible question is then: how many times shall one repeat the training procedure before being confident about the performance achieved by the best performing model among the generated ones? We answer to this question by providing a simple and effective guideline, based on the scenario optimization theory.

3. THE SCENARIO APPROACH

The scenario approach is developed in the context of optimization in the presence of uncertainty. It considers the case of a convex program subject to constraints that depend, possibly in a nonlinear and non-convex way, on a stochastic variable $\delta \in \Delta$ characterized by a possibly unknown distribution. Specifically, consider the problem of maximizing a scalar function $f(\delta)$. In our specific application, such a function can be for example the FIT value obtained by the trained model. We can write the optimization problem as follows:

$$\max_{\delta \in \Delta} f(\delta) \tag{4}$$

Problem (4) is computationally intensive and possibly intractable due to the infinite number of constraints. For this reason, the scenario approach represents a viable solution. In this approach, a finite number N_δ of samples $\delta_i \in \Delta$, named *scenarios*, are collected, and the following finite-dimensional problem is stated.

$$\begin{aligned} h^* = \arg\min_{h} \quad & h \\ \text{s.t.} \quad & f(\delta_i) \le h, \quad \forall i = 1, \dots, N_\delta \end{aligned} \tag{5}$$

Note that problem (5) is convex, since δ_i, $i = 1, \dots, N_\delta$ are fixed values now, and it can be rewritten as

$$h^* = \max_{i=1,\dots,N_\delta} f(\delta_i).$$

Since the problem considers only a finite number of scenarios, it is not guaranteed that the obtained solution is the absolute best one, however its optimality can be quantified in probabilistic terms, making the level of reliability of the solution a design parameter. Such a parameter is then traded off with the number of scenarios, which is clearly linked to computational complexity. The following assumption is required.

Assumption 1. The N_δ scenarios are independent and identically distributed (i.i.d.)

Under Assumption 1, the following theorem holds (Campi et al. (2009)).

Theorem 1. Consider a "violation parameter" $\epsilon \in (0, 1)$ and a "confidence parameter" $\beta \in (0, 1)$. If

$$N_\delta \ge \frac{2}{\epsilon}\left(\ln\left(\frac{1}{\beta}\right) + d - 1\right) \tag{6}$$

where d is the number of optimization variables in (5), then, with probability no smaller than $1 - \beta$, h^* satisfies all constraints in Δ with probability $1 - \epsilon$, i.e.

$$Pr(f(\delta) > h^*) \le \epsilon \tag{7}$$

Notably, the confidence parameter β can be very close to zero without scaling the required scenarios dramatically, see (6), so that (7) is guaranteed with probability arbitrarily close to 1 (e.g., $1 - 10^{-7}$). In summary, the scenario approach allows one to certify the probability of violation of a found solution against new and unseen realizations of uncertainty, regardless of the probability distribution as long as it is the same one and samples are taken independently. Note that the bound (6) has the advantage of being explicit, however it is not tight: a tight and less conservative (but implicit) bound is also available from the theory (Campi and Garatti (2018)) and can be computed easily by numerical inversion.

We show next how this theory can be exploited in the training of ESN.

4. ESN TRAINING WITH OPTIMALITY GUARANTEES

When training an ESN, the function $f(\delta)$ corresponds to the FIT performance obtained after training a network whose randomly chosen parameters (i.e. W_x, W_u, and W_y) correspond to the uncertain variables δ. This means that the corresponding optimization problem (5) features just one optimization variable, i.e. h, hence $d = 1$ in (6). In view of this consideration, we propose to modify the tuning method described in Section 2 as follows:

(a) collect from the plant the, possibly normalized, input and output sequences u_{sys} and y_{sys};
(b) define the system order $n + n_u$;
(c) select the parameters β and ϵ. Compute the number of corresponding required scenarios N_δ according to (6) with $d = 1$ (or by numerical inversion of the tight bound in, e.g., Campi and Garatti (2018));
(d) for each scenario $j = 1, \dots, N_\delta$ repeat points (c)-(h) of the tuning method described in Section 2, ending with $W_{out,j}$;
(e) using the available validation dataset, compute FIT_j according to equation (3);
(f) take $j^* = \arg\max_{j=1,\dots,N_\delta} FIT_j$;
(g) select W_{x,j^*}, W_{u,j^*}, W_{y,j^*}, W_{out_1,j^*}, and W_{out_2,j^*} as the optimal parameters for model (1).

Using the scenario optimization theory, we directly obtain sound optimality guarantees on the model selected according to the procedure described above. In fact, in view of Theorem 1, it is possible to conclude that, if we train a new model with random parameters W_x, W_u, W_y, the probability that the latter outperforms the one obtained with our procedure is smaller than ϵ (with confidence $(1 - \beta) \approx 1$). To give a practical example, with a confidence parameter $\beta = 10^{-7}$ and violation parameter $\epsilon = 0.05 = 5\%$, the resulting number of training instances that needs to be carried out is $N_\delta = 645$, which is very manageable. Most importantly, note that such a number of samples *is independent from the complexity of the network*: in the example, we will need these 645 training instances no matter if the model order of the considered ESN is 10, 100, or 10^6.

So far, the approach has been presented by considering maximization of the FIT. This is without loss of generality: the procedure can be applied to any other objective function and either to minimization or maximization. For example, one can set up the equivalent procedure by considering the RMSE, in which case the problem is a minimization one, and one shall pick the ESN that scores the smallest RMSE among the N_δ trained instances.

Finally, note that the same reasoning also applies when one wants to derive guarantees on the worst-performing network, e.g. to compute a probabilistic lower bound to the FIT value or upper bound to the RMSE. In fact, it is enough to pick the worst-performing ESN among the trained ones, instead of the best-performing one, and the same probabilistic guarantees are obtained (i.e., with high probability, a newly trained network will score no worse than the selected one with probability $1 - \varepsilon$).

Remark 1. Notably, the proposed approach can be easily extended to other neural network structures, whenever the training procedure entails a random part.

5. SIMULATION RESULTS

In this section the described procedure is applied to assess the quality of ESN models of a simulated plant. Specifically, the non-linear benchmark employed in our simulations corresponds to the pH neutralization process, typically exploited for the purification of waste waters. The aim of the process is to maintain the pH of the solution in the principal tank at a neutral level, i.e. $pH = 7$.

In Figure 1 a simplified scheme of the process is reported.

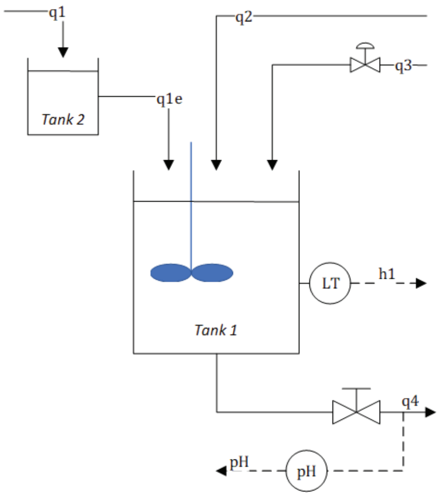


Figure 1. pH neutralization system scheme

The system is composed of two tanks: a principal one, also known as reactor tank, in which the main transformation occurs, and an acid tank, which is fed by an acid stream q_1. The reactor has three input flows, namely the acid stream q_{1e} (output of the acid tank), the buffer flow q_2 (conjugate of acid-base pair solution) and the alkaline stream q_3, and it has one output flow q_4, which is the final solution where the pH is measured and controlled. Furthermore, the system is endowed with: a level sensor of the liquid inside the reactor, a sensor for the measurement of output solution pH and an agitator that allows to keep the fluid characteristics constant in the liquid volume.

To derive a simplified state-space model of the process, the following assumptions are made:

- the dynamics of the acid tank are much faster than those of the reactor tank, hence the input flow q_{1e} is considered to be equal to q_1;
- the input flow-rate q_{1e} is considered constant;
- the input flow-rate q_2 is considered as unmeasured disturbance.

The resulting physical model of the process has one control input regulated by a valve $u = q_3$, one output $y = q_4$, one disturbance $d = q_2$ and three states, that are two ions concentrations and the solution height inside the reactor, $x = [W_{a4}\ W_{b4}\ h_1]^T$. Ultimately the process is described by the following differential state-space equations with a constraint, as reported in Hall and Seborg (1989a):

$$\begin{aligned}\dot{x}(t) &= f_1(x(t)) + f_2(x(t))u(t) + f_3(x(t))d(t) \\ c(x(t), y(t)) &= 0\end{aligned} \tag{8}$$

where

$$f_1(x(t)) = \left[\frac{q_1}{A_1 x_3}(W_{a1} - x_1), \frac{q_1}{A_1 x_3}(W_{b1} - x_2), \frac{1}{A_1}(q_1 - C_{v4}(x_3 + z)^n)\right]^T$$

$$f_2(x(t)) = \left[\frac{1}{A_1 x_3}(W_{a3} - x_1), \frac{1}{A_1 x_3}(W_{b3} - x_2), \frac{1}{A_1}\right]^T$$

$$f_3(x(t)) = \left[\frac{1}{A_1 x_3}(W_{a2} - x_1), \frac{1}{A_1 x_3}(W_{b2} - x_2), \frac{1}{A_1}\right]^T$$

and

$$c(x, y) = x_1 + 10^{y-14} + 10^{-y} + x_2 \frac{1 + 2 \cdot 10^{y - pK_2}}{1 + 10^{pK_1 - y} + 10^{y - pK_2}}$$

and pK_i is the i^{th} dissociation constant of the weak acid H_2CO_3. Table 1 presents the nominal parameter values , where $[M] = [\frac{mol}{L}]$.

Table 1. Nominal operating conditions of the pH system

$z = 11.5\,cm$	$W_{a1} = 3.00 \cdot 10^{-3}\,M$	$q_1 = 16.6\,mL/s$
$C_{v4} = 4.59$	$W_{b1} = 0.00\,M$	$q_2 = 0.55\,mL/s$
$n = 0.607$	$W_{a2} = -0.03\,M$	$q_3 = 15.6\,mL/s$
$pK_1 = 6.35$	$W_{b2} = 0.03\,M$	$q_4 = 32.8\,mL/s$
$pK_2 = 10.25$	$W_{a3} = 3.05 \cdot 10^{-3}\,M$	$A_1 = 207\,cm^2$
$h_1 = 14\,cm$	$W_{b3} = 5.00 \cdot 10^{-5}\,M$	$W_{a4} = -4.32 \cdot 10^{-4}\,M$
$pH = 7.0$	$W_{b4} = 5.28 \cdot 10^{-4}\,M$	

5.1 *Optimality guarantees & validation*

In this section we present the simulation results produced by the application of the proposed procedure.

The optimality guarantees of this approach are derived and tested by sampling a suitable number of instances N_δ. The design parameters are given in Table 2. First of all, we have employed a simulator of the real plant, including an output-measurement disturbance to retrieve realistic data useful for training the ESN models. A Multilevel Pseudo-Random Signal (MPRS), simulating the input flow rate q_3, has been used to excite the plant over the whole operating conditions using a switching period of 1000 seconds, which is longer than the real system settling time, and an amplitude included in the interval $[12.7, 16.7]$ mL/s. The corresponding output of the process takes values in between 6 and 8.65, see the validation data in Figure 2.

Table 2. Design parameters for the scenario approach

Parameter	Description	Value
β	confidence parameter	10^{-7}
ϵ	violation parameter	0.05
d	optimization variables	1
N_δ	number of instances	645

The sampling time for input and output signals used to train the different ESN of the scenario approach is $T_s = 10$

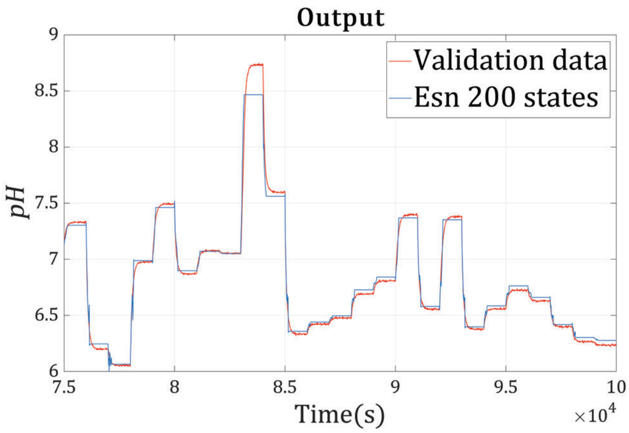


Figure 2. Validation dataset: sampled output of the real plant (red line) vs. ESN predictions with 200 states (blue line)

s to empirically obtain 30 samples in the settling time of the system's step response.

Subsequently, according to the algorithm proposed in Section 4, we trained $N_\delta = 645$ ESN models computing all the respective values of $RMSE_j$ and FIT_j, $j = 1, \ldots, 645$ on validation data, for different values of the model order (from 10 to 500 states). We have finally computed the values $\overline{FIT}$ and $\overline{RMSE}$, which correspond to the best FIT and the worst RMSE value, respectively, over the 645 instances.

Then, to verify empirically the validity of the guarantees provided by the theory, we collected other 500 new scenarios, trained as many ESNs and evaluated the associated RMSE and FIT values over the same validation dataset. In this way, we could empirically test the conservativeness of the approach, in terms of discrepancy between the set violation parameter ϵ and the share of new instances that scored better FIT (resp. worse RMSE) than the chosen model.

In Table 3 we present the absolute number of violations over the total amount of new instances considered in the testing phase (500) for different model orders. In all the cases the probabilistic bounds are verified, as the maximum number of ESNs that violate the constraints are respectively $3/500 = 0.6\%$ for the FIT and $4/500 = 0.8\%$ for the RMSE. Note that these values are definitely much smaller than the guaranteed violation probability $\epsilon = 5\%$. A possible reason for this is that, for a given sample of the internal weights, the training procedure still tries to optimize the predictive capability of the network with the least squares estimation of the output layer.

Figure 3 shows the results obtained with the scenario approach, both for the best and worst cases, using ESNs characterized by 200 internal units. In particular, Figure 3(a) reports the sampling procedure performed to obtain the best $\overline{FIT}$ and the worst $\overline{RMSE}$, while Figure 3(b) displays the testing phase on new collected instances, where the thresholds (red continuous lines) indicate the optimal values derived in the sampling phase, and the circles indicate for the ESNs that violate the bounds enforced by such optima.

Table 3. Testing violations over 500 new instances

Nb. of ESN states	Best case	Worst case	$\overline{FIT}$
10	$0/500 = 0.0\%$	$0/500 = 0.0\%$	82.30
20	$0/500 = 0.0\%$	$0/500 = 0.0\%$	84.57
30	$1/500 = 0.2\%$	$0/500 = 0.0\%$	85.40
40	$1/500 = 0.2\%$	$1/500 = 0.2\%$	86.10
50	$0/500 = 0.0\%$	$0/500 = 0.0\%$	87.08
60	$3/500 = 0.6\%$	$4/500 = 0.8\%$	87.10
70	$0/500 = 0.0\%$	$1/500 = 0.2\%$	87.73
80	$0/500 = 0.0\%$	$1/500 = 0.2\%$	88.06
90	$3/500 = 0.6\%$	$0/500 = 0.0\%$	87.79
100	$0/500 = 0.0\%$	$0/500 = 0.0\%$	87.95
120	$0/500 = 0.0\%$	$3/500 = 0.6\%$	88.42
140	$0/500 = 0.0\%$	$4/500 = 0.8\%$	88.75
160	$0/500 = 0.0\%$	$0/500 = 0.0\%$	89.04
180	$0/500 = 0.0\%$	$4/500 = 0.8\%$	89.20
200	$1/500 = 0.2\%$	$2/500 = 0.4\%$	89.15
250	$0/500 = 0.0\%$	$0/500 = 0.0\%$	89.88
300	$0/500 = 0.0\%$	$3/500 = 0.6\%$	90.15
350	$0/500 = 0.0\%$	$2/500 = 0.4\%$	90.16
400	$0/500 = 0.0\%$	$1/500 = 0.2\%$	90.11
450	$3/500 = 0.6\%$	$0/500 = 0.0\%$	89.85
500	$0/500 = 0.0\%$	$1/500 = 0.2\%$	90.08

6. CONCLUSIONS

We presented an application of the scenario approach to train Echo State Networks, a popular class of recurrent neural networks. First, the established training algorithm used to derive a state-space model of ESN is reported, where the parameters of the state equations are randomly generated. Then, this algorithm is modified to obtain the optimal network - with sound guarantees - according to the scenario approach. Eventually, the approach has been tested in order to empirically evaluate the guarantees on the FIT and RMSE values computed on a validation dataset for the models identified with ESNs. A novel set of scenarios confirmed the reliability of the solutions derived through the application of the proposed algorithm.

Future work is concerned with the extension to the case of constraints removal and the adaptation of neural networks with online data.

REFERENCES

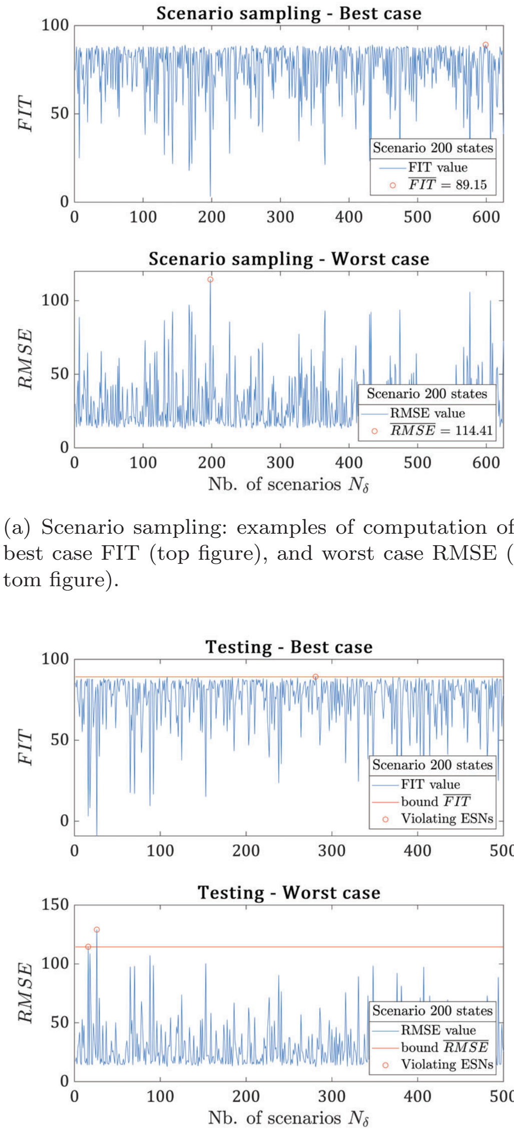
F. Bayer, M. Burger, and F. Allgower. Discrete-time incremental iss: A framework for robust nmpc. In *European Control Conference (ECC)*, pages 2068–2073. IEEE, 2013.

D.A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *IEEE control systems magazine*, 26(3):96–114, 2006.

L. Bugliari Armenio, E. Terzi, M. Farina, and R. Scattolini. Model predictive control design for dynamical systems learned by echo state networks. *IEEE Control Systems Letters*, 3(4):1044–1049, 2019.

G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, Jan 2005.

M. C. Campi and S. Garatti. *Introduction to the Scenario Approach*, volume 26. SIAM, 2018.



(a) Scenario sampling: examples of computation of the best case FIT (top figure), and worst case RMSE (bottom figure).

(b) Scenario testing: example of empirical test of the best case FIT (top figure), and worst case RMSE (bottom figure).

Figure 3. Scenario approach: example of sampling and testing of an ESN of order 200.

M.C. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009.

M.P. Deisenroth, C.E. Rasmussen, and D. Fox. Learning to control a low-cost manipulator using data-efficient reinforcement learning. *Robotics: Science and Systems VII*, pages 57–64, 2011.

R.C. Hall and D.E. Seborg. Modelling and self-tuning control of a multivariable ph neutralization process part i: modelling and multiloop control. In *American Control Conference*, pages 1822–1827. IEEE, 1989a.

R.C. Hall and D.E. Seborg. Modelling and self-tuning control of a multivariable ph neutralization process part i: Modelling and multiloop control. In *American Control Conference*, pages 1822–1827. IEEE, 1989b.

S.S. Haykin et al. *Neural networks and learning machines.* New York: Prentice Hall, 2009.

G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, and A. Senior. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

H. Jaeger. The echo state approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148 (34):13, 2001.

H. Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, volume 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.

H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

S.J. Kim. A note on the connection between incremental input-to-state stability and fading memory in nonlinear systems. 2005.

J.N. Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data.* Oxford University Press, 2013.

C.H. Lee and C.C. Teng. Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transactions on fuzzy systems*, 8(4):349–366, 2000.

L. Ljung. *System identification: theory for the user.* Prentice-hall, 1987.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

B.A. Pearlmutter. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2): 263–269, 1989.

D.T. Pham and X. Liu. *Neural Networks for identification, prediction and control.* 1995.

H.H. Sohrab. *Basic real analysis*, volume 231. Springer, 2003.

I. Sutskever, O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

I. Szita, V. Gyenes, and A. Lőrincz. Reinforcement learning with echo state networks. In *International Conference on Artificial Neural Networks*, pages 830–839. Springer, 2006.

J.H. Tan, Y. Hagiwara, W. Pang, I. Lim, S.L. Oh, M. Adam, R. San Tan, M. Chen, and U.R. Acharya. Application of stacked convolutional and long short-term memory network for accurate identification of cad ecg signals. *Computers in biology and medicine*, 94:19–26, 2018.

M.H. Tong, A.D. Bickett, E.M. Christiansen, and G.W. Cottrell. Learning grammatical structure with echo state networks. *Neural networks*, 20(3):424–432, 2007.