LSTM Neural Networks: Input to State Stability and Probabilistic Safety Verification

Fabio Bonassi　　　　　　　　　　　　　　　　　　　　　FABIO.BONASSI@POLIMI.IT
Enrico Terzi　　　　　　　　　　　　　　　　　　　　　　ENRICO.TERZI@POLIMI.IT
Marcello Farina　　　　　　　　　　　　　　　　　　　MARCELLO.FARINA@POLIMI.IT
Riccardo Scattolini　　　　　　　　　　　　　　　RICCARDO.SCATTOLINI@POLIMI.IT
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

The goal of this paper is to analyze Long Short Term Memory (LSTM) neural networks from a dynamical system perspective. The classical recursive equations describing the evolution of LSTM can be recast in state space form, resulting in a time-invariant nonlinear dynamical system. A sufficient condition guaranteeing the Input-to-State (ISS) stability property of this class of systems is provided. The ISS property entails the boundedness of the output reachable set of the LSTM. In light of this result, a novel approach for the safety verification of the network, based on the Scenario Approach, is devised. The proposed method is eventually tested on a *pH* neutralization process.

Keywords: LSTM, Input to State Stability, Safety Verification, Scenario Approach.

1. Introduction

In recent years Neural Networks (NN) have spread as a powerful tool for multiple purposes in several fields of science and engineering (Haykin, 1994), such as system diagnosis (Filippetti et al., 1995) and time series forecasting (Azoff, 1994). As discussed in (Mandic and Chambers, 2001), among the variants proposed in the literature, Recurrent Neural Networks (RNNs) are of particular interest in' systems identification and control, owing to their ability to describe dynamical systems.

This flexibility, however, comes at the cost of more complex training algorithms (Pascanu et al., 2013), which usually suffer of the vanishing or exploding gradient problems (Hochreiter, 1998). To overcome this issue, novel recurrent architectures have been recently proposed. Among them, *Echo State Networks* (ESNs) (Jaeger, 2002; Bugliari Armenio et al., 2019), and *Long-Short Term Memory* (LSTMs) networks (Hochreiter and Schmidhuber, 1997) are considered to be the most promising ones. More specifically, the latter have gained a great success and are considered to be the state-of-the-art for many tasks entailing long-term sequential learning (Greff et al., 2016).

Despite the wide popularity of LSTMs, their use in the context of analysis and control of dynamical systems is still limited, and the associated properties, such as the stability of the equilibria, have not yet been analyzed in depth. Indeed, only a limited number of contributions is nowadays available concerning stability (Stipanović et al., 2018; Deka et al., 2019; Barabanov and Prokhorov, 2002; Miller and Hardt, 2018), equilibria computation (Amrouche et al., 2018), and industrial applications (Lanzetti et al., 2019). In the mentioned works, however, the focus is on autonomous LSTMs, i.e. in absence of exogenous inputs, and the stability analyses conducted – being based on linearization methods – are only local. This essential lack of results needs to be filled in order to provide a solid ground for developing sound control techniques tailored for LSTM.

A further requirement is the ability of LSTMs, and more in general of RNNs, to properly generalize the behavior of the real system out of a limited training set. It is advisable to exploit the available knowledge of the system to certify that the output produced by the trained neural network is always coherent with the output of the real system, at least for a class of inputs of interest. This is known as *safety verification* of the network, and it is usually done by proving that the output reachable set lays in a defined safe region (Maler, 2008). The computation of reachable sets for generic discrete-time nonlinear systems has been discussed in (Bravo et al., 2006). Techniques to estimate the reachable output set of NNs have been proposed in (Fazlyab et al., 2019a) and (Fazlyab et al., 2019b), but they are limited to feed-forward NNs and cannot be applied to LSTMs.

In light of these motivations, the purpose of this paper is twofold. First, a sufficient condition guaranteeing the Input-to-State Stability (ISS) property (Sontag, 1995; Jiang and Wang, 2001) of LSTMs is derived. ISS is important here to guarantee consistency when we identify models of plants that display stability-like properties. Also, ISS implies that the state trajectories generated by bounded inputs remain bounded, which is the main assumption that provides a ground to the safety verification approach considered here, based on output reachable sets. The latter sets are estimated, in this paper, using the Scenario Approach introduced by Campi et al. (2009). While a similar idea has been proposed by Hewing and Zeilinger (2019), where it is used to estimate the reachable set of a linear system affected by stochastic disturbances, in this paper the Scenario Approach is used in a non-linear deterministic framework, where analytical approaches could not be applied. In a different context, the use of the Scenario Approach for optimization of ESNs has also been proposed by Bugliari Armenio et al. (2020).

Notation: Given a vector v, we denote with $|v|_2$ the 2-norm of v and with $|v|_1$ its 1-norm. Moreover, we denote with v^T its transpose, with v_i its i-th component, and with $\mathrm{diag}(v)$ the diagonal matrix with vector v on the main diagonal. With reference to a sequence of vectors $\mathbf{v} = v(0), v(1), ...,$ we define its infinity norm as $\|\mathbf{v}\|_\infty = \sup_{k \in \mathbb{N}} |v(k)|_2$. Given a real matrix A we denote with $|A|_p$ its induced norm, $p = 1, 2, \infty$. At any discrete time-step k, we indicate by x^+ the value of x at the next step, i.e. $x^+ = x(k+1)$. The Hadamard product between two vectors v and w is indicated by $v \circ w$. For compactness, the sigmoidal and hyperbolic tangent activation functions are denoted as $\sigma_g(x) = \frac{1}{1+e^{-x}}$ and $\sigma_c(x) = \tanh(x)$. These functions are bounded, i.e.

$$\sigma_g(t) \in (0, 1) \quad \forall t \in \mathbb{R}, \tag{1a}$$

$$\sigma_c(t) \in (-1, 1) \quad \forall t \in \mathbb{R}, \tag{1b}$$

and Lipschitz continuous (Sohrab, 2003), with constants $L_g = 0.25$ and $L_c = 1$, respectively.

2. LSTM networks

2.1. LSTM state space form

Let us consider a single-layer LSTM network described by the following equations, obtained enforcing the original equations from (Gers and Schmidhuber, 2001) to be strictly proper:

$$x^+ = \sigma_g(W_f u + U_f \xi + b_f) \circ x + \sigma_g(W_i u + U_i \xi + b_i) \circ \sigma_c(W_c u + U_c \xi + b_c) \tag{2a}$$

$$\xi^+ = \sigma_g(W_o u + U_o \xi + b_o) \circ \sigma_c(x^+), \tag{2b}$$

$$y = C\xi + b_y, \tag{2c}$$

where $u \in \mathbb{R}^{n_u}$ is the input, $x \in \mathbb{R}^{n_x}$ is the so-called hidden state, $\xi \in \mathbb{R}^{n_x}$ is the so-called cell state, and $y \in \mathbb{R}^{n_y}$ is the system output. The tuning parameters, suitably selected during network's training procedure, are the weight matrices $W_f, W_i, W_c, W_o \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_u, 2n_x}$, $U_f, U_i, U_c, U_o \in \mathbb{R}^{n_x \times n_x}$, and the bias vectors b_f, b_i, b_c, b_o, b_y of proper dimensions. In this paper, without loss of generality, inputs are assumed to be bounded by

$$u \in \mathcal{U} = [-1, 1]^{n_u}. \tag{3a}$$

Indeed, the inputs are usually subject to saturation, and can therefore be normalized according to standard un-biasing and normalization procedures. In light of (1), from (2b) it is possible to notice that the vector of cell states is also bounded by

$$\xi \in \Xi = (-1, 1)^{n_x}. \tag{3b}$$

At this stage, let us consider the bias vector b_c as a constant input to system (2). This is merely required for the sake of consistency with the existing theory, and indeed the bias b_c is commonly identified during the training of the network. Letting $\chi = \begin{bmatrix} x^T & \xi^T \end{bmatrix}^T$, it is possible to re-write (2) in the standard form of a discrete-time, invariant, and nonlinear dynamical system:

$$\begin{aligned} \chi^+ &= f_{LSTM}(\chi, u, b_c), \\ y &= g_{LSTM}(\chi). \end{aligned} \tag{4}$$

2.2. Stability analysis

The goal of this section is to investigate the stability properties of the LSTM network (4). More specifically, we provide a sufficient condition to guarantee the ISS property of the network. Let us firstly recall some notions from (Bayer et al., 2013), required for the statement of the theoretical results of this paper. For the definitions of $\mathcal{K}$, $\mathcal{K}_\infty$ and $\mathcal{KL}$ functions, the reader is addressed to (Jiang and Wang, 2001). Considering the sequence $\mathbf{u} = u(0), u(1), \ldots$ and an initial state χ_0, we denote by $\chi(k, \chi_o, \mathbf{u}, b_c)$ the state, at time k, of system (4), fed by the input sequence $\mathbf{u}$. The following definitions can hence be stated.

Definition 1 (ISS (Jiang and Wang, 2001)) *The system (4) is Input-to-state stable with respect to inputs $u \in \mathcal{U}$ and b_c if there exist functions $\beta \in \mathcal{KL}$, and $\gamma_u, \gamma_b \in \mathcal{K}_\infty$ such that, for any $k \in \mathbb{Z}_{\geq 0}$, any initial condition χ_0, any value of b_c, and any input sequence $\mathbf{u} \in \mathcal{U}$, it holds that:*

$$|\chi(k, \chi_0, \mathbf{u}, b_c)|_2 \leq \beta(|\chi_0|_2, k) + \gamma_u(\|\mathbf{u}\|_\infty) + \gamma_b(|b_c|_2) \tag{5}$$

Definition 2 (ISS Lyapunov function (Jiang and Wang, 2001)) *A continuous function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called an ISS-Lyapunov function for (4) if there exist functions $\psi_1, \psi_2, \psi \in \mathcal{K}_\infty$ and $\sigma_u, \sigma_b \in \mathcal{K}$ such that, for all $\chi \in \mathbb{R}^{2n_x}$, for all $b_c \in \mathbb{R}^{n_x}$, and $u \in \mathbb{R}^{n_u}$, it holds that:*

$$\psi_1(|\chi|_2) \leq V(\chi) \leq \psi_2(|\chi|_2) \tag{6a}$$

$$V(f_{LSTM}(\chi, u, b_c)) - V(\chi) \leq -\psi(|\chi|_2) + \sigma_u(|u|_2) + \sigma_b(|b_c|_2) \tag{6b}$$

Theorem 3 (Jiang and Wang (2001)) *If system (4) admits a time invariant ISS Lyapunov function such that (6a) and (6b) hold, then it is ISS in the sense specified by Definition 1.*

Based on the previous results, Theorem 4 is formulated. The proof is reported in Appendix A.

Theorem 4 *Given the LSTM network* (2)*, if*

$$
\begin{aligned}
\left(1 + \sigma_g(|\begin{bmatrix} W_o & U_o & b_o \end{bmatrix}|_\infty)\right) \sigma_g(|\begin{bmatrix} W_f & U_f & b_f \end{bmatrix}|_\infty) &< 1, \\
\left(1 + \sigma_g(|\begin{bmatrix} W_o & U_o & b_o \end{bmatrix}|_\infty)\right) \sigma_g(|\begin{bmatrix} W_i & U_i & b_i \end{bmatrix}|_\infty) |U_c|_1 &< 1,
\end{aligned}
\tag{7}
$$

then (2) *is Input-to-State stable with respect to* $u \in \mathcal{U}$ *and to* b_c.

Remark 5 *Condition* (7) *involves the LSTM's weight matrices and bias vectors. Therefore, it can be employed to check a-posteriori the ISS of a trained LSTM network, or it can be used as a constraint during the training procedure of the network to enforce the ISS property.*

Remark 6 *If multi-layer LSTMs are considered, it should be noted that the cell state* ξ *of the* i*-th layer becomes the input of the next layer. Since* ξ *satisfies the same properties input* u*, namely* (3a)*, it follows that a sufficient condition for the ISS of the network is that each layer satisfies* (7)*.*

3. Safety verification

As discussed, training the LSTM network with constraints (7) explicitly enforced, allows one to obtain the ISS property. This property is extremely relevant in the context of systems control, as it guarantees the boundedness network's reachable set, and that the effect of initialization on network's output asymptotically vanishes. Nonetheless, the reliability of the network could be harmed by the limited size of the training set usually adopted, either due to data unavailability or to computational complexity reasons. Overfitting phenomena and limitations of the available numerical training procedures could, as well, undermine LSTMs' generalization capabilities. This entails the necessity to perform a safety verification of the network, in order to certify – at least for a given class of input signals – that it does not exhibit meaningless or non-physical outputs. This task is usually accomplished computing the output reachable set, and assessing that it entirely lays within a known safe set. While for feed-forward NNs some methods for the estimation of the output reachable set have been proposed, see (Fazlyab et al., 2019b), for RNNs this still represents an open problem.

First remark that, in view of Definition 1, the ISS property allows to retrieve an explicit bound on the reachable set of the network. In particular, for any initial state χ_0, and any bounded input sequence $\nu(0), \nu(1), \ldots$, the future state trajectories are guaranteed to be bounded by (5), where the functions $\beta(\cdot, \cdot)$, $\gamma_u(\cdot)$, and $\gamma_b(\cdot)$ can be computed from the ISS-Lyapunov function, as described in (Jiang and Wang, 2001). The output reachable set can then be computed applying the linear output transformation g_{LSTM}. Nonetheless, such bound may suffer from an excessive conservativeness. The tightening of ISS-based reachable set bound will be object of future research work.

In place of the ISS-based analytic (but possibly conservative) computation of the output reachable set, a probabilistic method is here proposed, based on the Scenario Approach developed in (Campi et al., 2009). Let the initial state of the LSTM network χ_0 be a random variable extracted from a set $\mathcal{X}_0$, characterized by some probability measure $\mathbb{P}_\chi$. Consider a time horizon τ, and a class $\mathcal{U}_\tau$ of input signals $u(0), \ldots, u(\tau)$, such that $u(k) \in \mathcal{U}$ for all $k = 0, \ldots, \tau$. Assume that $\mathcal{U}_\tau$ is characterized by some – possibly unknown – probability measure $\mathbb{P}_u$. To ensure the validity of the results, the class of inputs $\mathcal{U}_\tau$ shall be compliant with the policy by which the system is operated.

The goal is to find the smallest ball $\mathcal{Y}$ containing the output reachable set, i.e. the set where the output trajectories produced by any input sequences extracted from $\mathcal{U}_\tau$ and starting from any initial state drawn from $\mathcal{X}_0$ lie. The radius ρ_y^* of set $\mathcal{Y}$ is defined as the solution of

$$\rho_y^* = \min_{\rho_y} \quad \rho_y, \tag{8a}$$

$$s.t. \quad \|\mathbf{y}(\chi_0, \mathbf{u})\|_\infty \le \rho_y \qquad \forall \chi_0 \in \mathcal{X}_0, \, \forall \mathbf{u} \in \mathcal{U}\tau, \tag{8b}$$

where $\mathbf{y}(\chi_0, \mathbf{u})$ is the output sequence obtained feeding the trained LSTM (4) with the input sequence $\mathbf{u}$ and initial state χ_0. Notably, this problem cannot be solved directly, due to infinite cardinality of constraint (8b). Nonetheless, owing to the convexity of (8) with respect to the optimization variable ρ_y, the Scenario Approach proposed by Campi et al. (2009) can be exploited to recast the optimization problem as a finite-dimensional linear program, that allows to compute $\rho_y^*(\varepsilon, \beta)$ such that

$$\mathbb{P}_{\chi,u}\left\{\|\mathbf{y}(\chi_0, \mathbf{u})\|_\infty > \rho_y^*(\varepsilon, \beta)\right\} \le \varepsilon \tag{9}$$

with confidence $1 - \beta$. In this way, the deterministic problem (8b) is relaxed into a chance-constrained one. To do so, it is necessary to generate a number N of scenarios, each corresponding to a sample of the uncertain variables χ_0 and $\mathbf{u}$, drawn from the respective sets according to the associated probability density functions. We define with $\chi_0^{(i)}, \mathbf{u}^{(i)}$ the i-th scenario, where $i = 1, \ldots, N$. Theorem 1 of (Campi et al., 2009) allows to compute $\rho_y^*(\varepsilon, \beta)$ as

$$\rho_y^*(\varepsilon, \beta) = \min_{\rho_y} \quad \rho_y, $$
$$s.t. \quad \|\mathbf{y}(\chi_0^{(i)}, \mathbf{u}^{(i)})\|_\infty \le \rho_y \quad \text{for all } i = 1, ..., N, \tag{10}$$

provided that a sufficient number N of scenarios is adopted. In particular, being $d = 1$ the number of optimization variables, it is sufficient that N fulfills the following inequality (Campi et al., 2009).

$$N \ge \frac{2}{\varepsilon}\left(\ln \frac{1}{\beta} + d\right) \tag{11}$$

It is worth remarking that (10) is linear and convex, since $\mathbf{y}(\chi_0^{(i)}, \mathbf{u}^{(i)})$ does not depend on any optimization variable. To conclude, defining $\mathcal{Y}(\varepsilon, \beta) = \{y \in \mathcal{R}^{n_y} : |y|_2 \le \rho_y^*(\varepsilon, \beta)\}$, if $\mathcal{Y}$ is contained in the safe output set, then one can assess - with the mentioned probability levels - the safety of the identified LSTM network, as long as the inputs belong to the considered class.

4. Simulation example

The considered benchmark example is the *pH* neutralization process described in (Hall and Seborg, 1989) and used in (Bugliari Armenio et al., 2019) for the analysis of ESNs. The plant, depicted in Figure 1, is composed of two tanks. Tank 2 is fed by an acid stream q_1 and, as output, has the flowrate q_{1e}. Being the hydraulic dynamics faster than the others involved, we assume $q_1 = q_{1e}$. The reactor Tank 1 is fed by three flowrates, namely q_1, a buffer flowrate q_2 and an alkaline flowrate q_3. q_1 and q_2 are not manipulated variables, and represent disturbances, while a controlled valve modulates q_3. The output flowrate of the tank is q_4, where the *pH* is measured.

The simplified third-order model, see (Bugliari Armenio et al., 2019) for equations, has one input (q_3), and one output (the *pH* value). The simulator of this plant has been implemented in MATLAB, and it has been fed with a Multilevel Pseudo-Random Signal (MPRS), so as to properly excite the system, adopting a sampling time $T_s = 10$s. White noise has been added, both to input
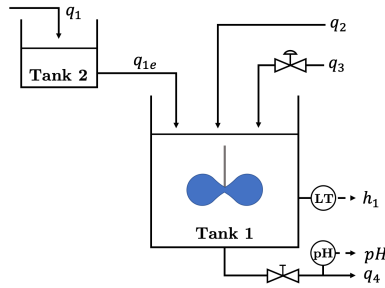
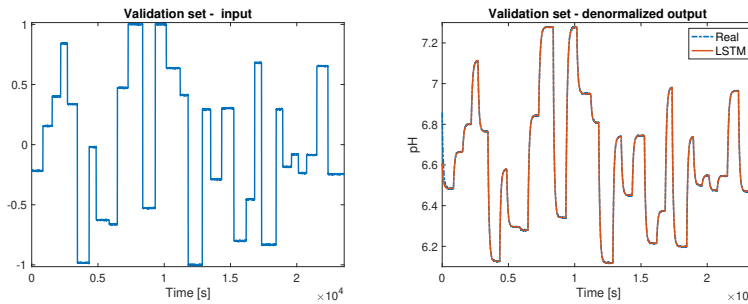Figure 1: Schematic layout of the plant

Figure 2: Performances of the LSTM on the validation dataset.

and to measured output data, to mitigate overfitting phenomena. The training and validation sets consist of 4400 and 2250 (u, y) samples, respectively. Samples have been normalized using the mean and the maximum deviation across the entire dataset.

A single-layer LSTM neural network (2) with $n_x = 5$ cells has then been trained using the MATLAB function *fmincon* to minimize the mean-square prediction error. In order to ensure network's ISS, conditions (7) have been enforced as explicit constraints during the training procedure. The modeling performances over the validation dataset are reported in Figure 2, where a simulation of the trained network, starting from a random initial state and forced by the input u, is shown. Notice that, in Figure 2, the denormalized output is depicted for the sake of clarity.

A quantitative performance index is the FIT $[\%]$ value, computed as

$$\text{FIT} = 100 \left(1 - \frac{|\mathbf{y}_{val} - \mathbf{y}_{LSTM}|_2}{|\mathbf{y}_{val}|_2} \right), \tag{12}$$

where $\mathbf{y}_{val}$ is the real system output trajectory and $\mathbf{y}_{LSTM}$ is the output of the trained LSTM network, fed by the same input. The trained network scores, after an initial transient due to the wrong initial state, FIT $= 98\%$, confirming remarkable modeling properties.

The probabilistic safety verification proposed in Section 3 has then been tested on the LSTM network. In particular, the input class $\mathcal{U}_\tau$ selected for this test is the set of MPRS, with amplitude ρ_u and with a total duration of $\tau = 2000$ time-steps, i.e. 20000 seconds. Since $n_u = 1$, any sequence $\mathbf{u}^{(i)}$ sampled from $\mathcal{U}_\tau$ is composed by steps with an amplitude uniformly extracted from $[-\rho_u, \rho_u]$, and with a random duration uniformly drawn in the range $[300, 2000]$ seconds. In Figure 3(a) some examples of input sequences thus generated have been depicted, for $\rho_u = 0.7$.

The initial state χ_0 has been extracted from a uniform distribution over $\mathcal{X}_0 = [-0.1, 0.1]^{2n_x}$. Notice that, owing to LSTM's ISS, the effect of the initial state asymptotically vanishes.
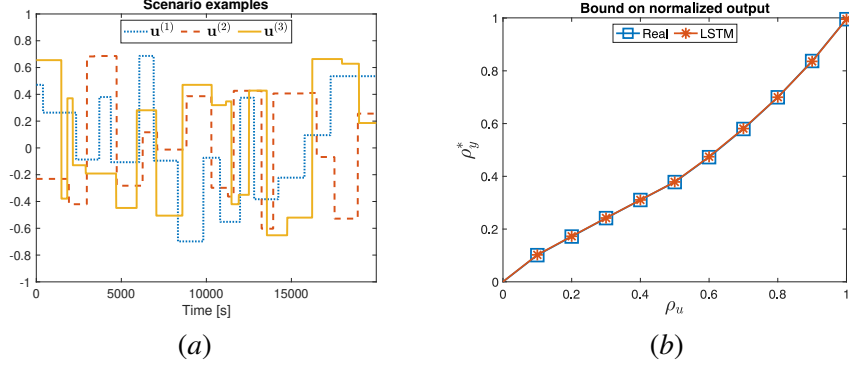
Figure 3: (a) Some examples of input sequence scenarios for $\rho_u = 0.7$; (b) Bounds ρ_y^* on the LSTM normalized output reachable set for different input bounds ρ_u, compared to those of the real system.

In light of condition (11), to have a violation probability $\varepsilon = 10^{-2}$ with confidence $\beta = 10^{-6}$, $N = 2964$ scenarios must be generated. Problem (10) is hence evaluated for different values of ρ_u, namely for $\rho_u = \{0.1, 0.2, ..., 1\}$, both for the LSTM network and the real system. Remarkably, the results reported in Figure 3(b) witness that the LSTM is guaranteed to operate almost in the same region of the real system, i.e. a ball with radius $\rho_y \approx 1$.

5. Conclusions

In this work, the stability properties of LSTM neural networks have been investigated. In particular, a condition on LSTM's parameters to guarantee the Input-to-State Stability property of the network has been derived, which can be possibly enforced during the training procedure. A probabilistic method for the safety verification of the trained network has then been formulated, which allows – for any class of input signals – to retrieve a tight bound on network's output reachable set, up to a desired confidence level. The proposed algorithm has been eventually tested on a well-known benchmark system, the *pH* neutralization process, showing satisfactory performances.

Appendix A. Proof of Theorem 4

Let us define the following variables, for the sake of conciseness:

$$
\begin{aligned}
f(u, \xi) &= W_f u + U_f \xi + b_f, & i(u, \xi) &= W_i u + U_i \xi + b_i, \\
c(u, \xi) &= W_c u + U_c \xi + b_c, & o(u, \xi) &= W_o u + U_o \xi + b_o.
\end{aligned}
\tag{13}
$$

By means of these definitions, the state equations (2a) and (2b) can be written as

$$
x^+ = \sigma_g(f(u, \xi)) \circ x + \sigma_g(i(u, \xi)) \circ \sigma_c(c(u, \xi)),
\tag{14a}
$$

$$
\xi^+ = \sigma_g(o(u, \xi)) \circ \sigma_c(x^+).
\tag{14b}
$$

Letting $\alpha_f(u, \xi) = \text{diag}\left(\sigma_g(f(u, \xi))\right), \alpha_o(u, \xi) = \text{diag}\left(\sigma_g(o(u, \xi))\right),$ and $\alpha_i(u, \xi) = \text{diag}\left(\sigma_g(i(u, \xi))\right),$ it is possible to re-write (14) as

$$
\begin{bmatrix} x^+ \\ \xi^+ \end{bmatrix} = \begin{bmatrix} \alpha_f(u, \xi)\, x + \alpha_i(u, \xi)\, \sigma_c(c(u, \xi)) \\ \alpha_o(u, \xi)\, \sigma_c\big(\alpha_f(u, \xi)\, x + \alpha_i(u, \xi)\, \sigma_c(c(u, \xi))\big) \end{bmatrix}.
\tag{15}
$$

In light of Theorem 3, consider the following candidate ISS-Lyapunov function:

$$V(\chi) = |\chi|_1 = |x|_1 + |\xi|_1. \tag{16}$$

By definition $V(\chi) \geq 0, \forall \chi \neq 0$. Condition (6a) follows trivially, since for a generic vector $v \in \mathbb{R}^{n_v}$ it holds that $|v|_2 \leq |v|_1 \leq \sqrt{n_v}\,|v|_2$. Therefore, condition (6a) is fulfilled taking

$$\psi_1(|\chi|_2) = |\chi|_2,$$
$$\psi_2(|\chi|_2) = \sqrt{2n_x}|\chi|_2.$$

We now compute $V(\chi^+) - V(\chi)$, showing that there exist functions $\psi(|\chi|_2)$, $\sigma_u(|u|_2)$ and $\sigma_b(|b_c|_2)$ fulfilling condition (6b). Using (15) one obtains

$$V(\chi^+) - V(\chi) = |\alpha_f(u,\xi)x + \alpha_i(u,\xi)\,\sigma_c(c(u,\xi))|_1 +$$
$$+ |\alpha_o(u,\xi)\,\sigma_c\big(\alpha_f(u,\xi)x + \alpha_i(u,\xi)\,\sigma_c(c(u,\xi))\big)|_1 - |x|_1 - |\xi|_1.$$

In view of the Lipschitzianity of the activation function $\sigma_c(\cdot)$, it holds that $\sigma_c(|v|_1) \leq |v|_1$, thus

$$V(\chi^+) - V(\chi) \leq \big[1 + |\alpha_o(u,\xi)|_1\big]\,|\alpha_f(u,\xi)|_1|x|_1 + \big[1 + |\alpha_o(u,\xi)|_1\big]\,|\alpha_i(u,\xi)|_1|c(u,\xi)|_1 - |x|_1 - |\xi|_1.$$

Recalling the definition of $c(u,\xi)$ in (13), it holds that

$$V(\chi^+) - V(\chi) \leq \big\{[1 + |\alpha_o(u,\xi)|_1]\,|\alpha_f(u,\xi)|_1 - 1\big\}|x|_1 + \big\{[1 + |\alpha_o(u,\xi)|_1]\,|\alpha_i(u,\xi)|_1\,|U_c|_1 - 1\big\}|\xi|_1$$
$$+ \big\{[1 + |\alpha_o(u,\xi)|_1]\,|\alpha_i(u,\xi)|_1|W_c|_1\big\}|u|_1 + \big\{[1 + |\alpha_o(u,\xi)|_1]\,|\alpha_i(u,\xi)|_1\big\}|b_c|_1. \tag{17}$$

Let $\zeta = [u^T, \xi^T, \mathbf{1}_{n_x}^T]^T$, where $\mathbf{1}_{n_x}$ is the unitary column vector with length n_x. From (3), it follows that $\zeta \in [-1,1]^{2n_x+n_u}$. By means of this, we can provide an upper bound for $|\alpha_f(u,\xi)|_1$:

$$|\alpha_f(u,\xi)|_1 \leq \max_\zeta |\mathrm{diag}\,(\sigma_g([W_f\ U_f\ b_f]\,\zeta))|_1 = \max_\zeta |\sigma_g([W_f\ U_f\ b_f]\,\zeta)|_\infty = \sigma_g(|\,[W_f\ U_f\ b_f]\,|_\infty).$$

Similarly,

$$|\alpha_o(u,\xi)|_1 \leq \sigma_g\,(|\,[W_o\ U_o\ b_o]\,|_\infty),$$
$$|\alpha_i(u,\xi)|_1 \leq \sigma_g\,(|\,[W_i\ U_i\ b_i]\,|_\infty).$$

In view of these bounds, conditions (7) ensure the existence of a strictly positive scalar δ such that

$$(1 + |\alpha_o(u,\xi)|_1)\,|\alpha_f(u,\xi)|_1 - 1 \leq -\delta,$$
$$(1 + |\alpha_o(u,\xi)|_1)\,|\alpha_i(u,\xi)|_1\,|U_c|_1 - 1 \leq -\delta, \tag{18}$$

for any $\xi \in \Xi$. Moreover, in light of (1a), it follows that $|\alpha_f(u,\xi)|_1$, $|\alpha_o(u,\xi)|_1$ and $|\alpha_i(u,\xi)|_1$ are upper-bounded by 1. Combining (17) and (18), and since $[1 + |\alpha_o(u,\xi)|_1]\,|\alpha_i(u,\xi)|_1\,|W_c|_1 \leq 2|W_c|_1$ and $[1 + |\alpha_o(u,\xi)|_1]|\alpha_i(u,\xi)|_1\,|b_c|_1 \leq 2|b_c|_1$, it is possible to derive that

$$V(\chi^+) - V(\chi) \leq -\delta|x|_1 - \delta|\xi|_1 + 2|W_c|_1|u|_1 + 2|b_c|_1 \leq -\delta|\chi|_2 + 2|W_c|_1\sqrt{n_u}|u|_2 + 2\sqrt{n_x}|b_c|_2.$$

In view of these results, it is possible to conclude that conditions (6b) are satisfied with

$$-\psi(|\chi|_2) = -\delta\,|\chi|_2,$$
$$\sigma_u(|u|_2) = 2|W_c|_1\sqrt{n_u}\,|u|_2,$$
$$\sigma_b(|b_c|_2) = 2\sqrt{n_x}\,|b_c|_2.$$

System (15) admits a time invariant ISS-Lyapunov function and, in view of Theorem 3, it is ISS. ∎

References

Massinissa Amrouche et al. Long short-term memory neural network equilibria computation and analysis. *NIPS 2018 Workshop Spatiotemporal Blind Submission*, 2018.

E Michael Azoff. *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc., 1994.

Nikita E Barabanov and Danil V Prokhorov. Stability analysis of discrete-time recurrent neural networks. *IEEE Transactions on Neural Networks*, 13(2):292–303, 2002.

Florian Bayer, Mathias Bürger, and Frank Allgöwer. Discrete-time incremental ISS: A framework for robust NMPC. In *2013 European Control Conference (ECC)*, pages 2068–2073. IEEE, 2013.

José M Bravo, Teodoro Alamo, and Eduardo F Camacho. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745–1751, 2006.

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

Luca Bugliari Armenio, Lorenzo Fagiano, Enrico Terzi, Marcello Farina, and Riccardo Scattolini. Scenario optimization for optimal training of echo state networks. *IFAC World Congress 2020 (accepted)*, 2020.

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

Shankar A Deka, Dušan M Stipanović, Boris Murmann, and Claire J Tomlin. Global asymptotic stability and stabilization of long short-term memory neural networks with constant weights and biases. *Journal of Optimization Theory and Applications*, 181(1):231–243, 2019.

Mahyar Fazlyab, Manfred Morari, and George J Pappas. Probabilistic verification and reachability analysis of neural networks via semidefinite programming. *arXiv preprint arXiv:1910.04249*, 2019a.

Mahyar Fazlyab, Manfred Morari, and George J Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *arXiv preprint arXiv:1903.01287*, 2019b.

Fiorenzo Filippetti, Giovanni Franceschini, and Carla Tassoni. Neural networks aided on-line diagnostics of induction motor rotor faults. *Conference Record of the 1993 IEEE Industry Applications Conference Twenty-Eighth IAS Annual Meeting*, 1995.

Felix A Gers and E Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2016.

Raymond C Hall and Dale E Seborg. Modelling and self-tuning control of a multivariable pH neutralization process part I: Modelling and multiloop control. In *1989 American Control Conference*, pages 1822–1827. IEEE, 1989.

Simon Haykin. *Neural networks: a comprehensive foundation.* Prentice Hall PTR, 1994.

Lukas Hewing and Melanie N Zeilinger. Scenario-based probabilistic reachable sets for recursively feasible stochastic model predictive control. *IEEE Control Systems Letters*, 4(2):450–455, 2019.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02): 107–116, 1998.

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

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

Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

Nicolas Lanzetti, Ying Z Lian, Andrea Cortinovis, Luis Dominguez, Mehmet Mercangöz, and Colin Jones. Recurrent neural network based MPC for process industries. In *2019 18th European Control Conference (ECC)*, pages 1005–1010. IEEE, 2019.

Oded Maler. Computing reachable sets: An introduction. *Tech. rep. French National Center of Scientific Research*, 2008.

Danilo P Mandic and Jonathon Chambers. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc., 2001.

John Miller and Moritz Hardt. Stable recurrent models. *arXiv preprint arXiv:1805.10369*, 2018.

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

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

Eduardo D Sontag. On the input-to-state stability property. *European Journal of Control*, 1(1): 24–36, 1995.

Dušan M Stipanović et al. Some local stability properties of an autonomous long short-term memory neural network model. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.