A Computing the derivatives

Let the Jacobian matrices of \( f(x, u; \theta) \) with respect to \( x \) and to \( \theta \) evaluated at the point \( (x_t, u_t; \theta) \) be denoted, respectively, as \( A_t \) and \( B_t \). Similarly, the Jacobian matrices of \( g(x_t, u_t; \theta) \) are denoted as \( C_t \) and \( F_t \). A direct application of the chain rule to \( (1) \) gives a recursive formula for computing the derivatives of the predicted output in relation to the parameters in the interval \( 1 \leq t \leq N \):

\[
D_{t+1} = A_t D_t + B_t \quad \text{for} \quad D_0 = 0
\]

\[
J_t = C_t D_t + F_t, \quad \text{(S1)}
\]

where we denote the Jacobian matrices of \( \hat{y}_t \) and \( x_t \) with respect to \( \theta \) respectively as \( J_t \) and \( D_t \).

For the cost function \( V \) defined as in Eq. (2), its gradient \( \nabla V \) is given by:

\[
\nabla V = \frac{1}{N} \sum_{t=1}^{N} J_t \ell'(#hat{y}_t, y_t), \quad \text{(S2)}
\]

where \( \ell'(#hat{y}_t, y_t) \) denotes the gradient of the loss function with respect to its first argument, evaluated at \((#hat{y}_t, y_t)\).

B Proofs

B.1 Entropy lower bound

Theorem 2. Let \( f(\cdot, u; \theta) \) in Eq. (1) be a one-to-one continuous differentiable map, and let \( f(x, u; \theta) \) be Lipschitz in \((x, \theta)\) with constant \( L_f \) on a compact and convex set \( \Omega = (\Omega_x, \Omega_u, \Omega_\theta) \). Then the entropy \( H_t \) in Eq. (3) with \( x_t \in \mathbb{R}^{N_x} \) satisfies:

\[
H_t + N_x \log L_f \leq H_{t+1}. \quad \text{(S3)}
\]

Proof. Under the assumption that \( f(\cdot, u; \theta) \) is a 1-1 continuous differentiable map (cf. Theorems 3-13 and 3-14 by Spivak (1998)), applying the change of variable \( x_{t+1} = f(x_t, u_t; \theta) \) we get:

\[
H_t = -\int_{f(x_t, u_t; \theta)} p_t(x_{t+1}) \log \left( \frac{p_t(x_{t+1})}{\det \frac{\partial x_{t+1}}{\partial x_t}} \right) dx_{t+1}
\]

\[
= H_{t+1} - \int_{f(x_t, u_t; \theta)} p_t(x_{t+1}) \log \left( \det \frac{\partial x_{t+1}}{\partial x_t} \right) dx_{t+1},
\]

where \( \frac{\partial x_{t+1}}{\partial x_t} \) is the Jacobian matrix of \( f(\cdot, u; \theta) \). Using Hadamard’s inequality:

\[
\log \left| \det \frac{\partial x_{t+1}}{\partial x_t} \right| \leq \sum_{i=1}^{N_x} \log \| v_i \|_2, \quad \text{(S4)}
\]

where \( v_i \) is the \( i \)-th column of the Jacobian matrix and \( \log \| v_i \|_2 \leq \log \left( \frac{\partial x_{t+1}}{\partial x_t} \right) \| v_i \|_2 \leq \log L_f \). Hence we have that:

\[
H_t + N_x \log L_f \leq H_{t+1}. \quad \text{(S5)}
\]
For \( l(y, \hat{y}) \) defined as in (b), it follows from Eq. (S6), Eq. (S7), and the fact that \( y \) contains only values in the set \( \{0, 1\} \), that:
\[
|l(\hat{y}, y) - l(\hat{z}, y)| \leq \|\hat{y} - \hat{z}\|_1 \leq \sqrt{N_y} \|\hat{y} - \hat{z}\|_2
\]  
(S8)
where \( \|\cdot\|_1 \) and \( \|\cdot\|_2 \) denote \( l_1 \) and \( l_2 \) norm of a vector and \( N_y \) is the number of outputs.

\[\Box\]

### B.3 Proof of Theorem 1 (a)

Assume two different trajectories resulting from simulating the system in Eq (1) with parameters and initial conditions \((x_0, \theta)\) and \((w_0, \phi)\), respectively. We denote the corresponding trajectories by \( x \) and \( w \). Let us call:
\[
\|\Delta \hat{y}\| = \|g(x_t, u_t; \theta) - g(w_t, u_t; \phi)\|. 
\]  
(S9)
Since \( f \) and \( g \) are Lipschitz in \((x, \theta)\) we have:
\[
\|f(x, u, \theta) - f(w, u, \phi)\| \leq L_f^2 \|x - w\|^2 + \|\theta - \phi\|^2,
\]
\[
\|g(x, u, \theta) - g(w, u, \phi)\| \leq L_g^2 \|x - w\|^2 + \|\theta - \phi\|^2,
\]
for all \((x, u, \theta)\) and \((w, u, \phi)\) in \((f, \Omega, \Omega_u, \Omega_w)\). Applying these relations recursively we get that:
\[
\|\Delta \hat{y}\|^2 \leq L_f^2 L_g^2 \ell_0 \|x_0 - w_0\|^2 + L_g \left( \sum_{t=0}^{\ell} L_f^2 \right) \|\theta - \phi\|^2.
\]
Since \( L_f \) is positive, the constant multiplying the second term in the above equation is always larger than the constant multiplying the first term. Hence, taking the square root on both sides of the above inequality and after simple manipulations, we obtain:
\[
\|\Delta \hat{y}\| \leq L_0 S(t) \|\theta, x_0\|^T - [\phi, w_0]^T \|, 
\]  
(S10)
where
\[
S(t) = \sum_{t=0}^t L_f^2 = \sqrt{t + 1} \quad \text{if } L_f = 1, 
\]
\[
\sum_{t=0}^t L_f^2 = \frac{L_f^{t+1} - 1}{L_f - 1} \quad \text{if } L_f \neq 1.
\]  
(S11)
Since \( \Omega \) is compact and \( \hat{y}_t \) is a (Lipschitz) continuous function of the parameters and initial conditions, then \( \hat{y}_t \) is bounded in \( \Omega \), i.e. \( \|\hat{y}_t\| \leq M(\ell) \). Furthermore, it follows from Eq. (S10) and from the existence of an invariant set\(^4\) in \( \Omega \) that \( M(t) = \mathcal{O}(S(t)) \).

The following inequality follows from Eq. (2), and from property 1 in Lemma 2:
\[
|V(\theta, x_0) - V(\phi, w_0)| \leq \frac{1}{N} \sum_{t=1}^N (K_1 L_y + K_2 M(t)) \|\Delta \hat{y}\|, 
\]  
(S12)
where \( L_y = \max_{1 \leq t \leq N} \|y_t\| \). Now, assembling Eq. (S12) and Eq. (S10) we obtain:
\[
|V(\theta, x_0) - V(\phi, w_0)| \leq L_{V} \|\theta, x_0\|^T - \phi, w_0\|^T \|, 
\]
for \( L_{V} = \left( \frac{N}{N} \sum_{t=1}^N (K_1 L_y + K_2 M(t)) S(t) \right) \). The asymptotic analysis of this expression with respect to \( N \) yields Eq. (6).

\(^4\)There are multiple ways to guarantee the invariant set premise will hold, but a very simple way is to just choose \( f \) such that \( f(0, u; 0) = 0 \). In this case, \( \{0\} \) is an invariant set and if \( \Omega \) contains this point the premise is satisfied. For this specific case, one can just choose \( [\phi, w_0] = 0 \) and it follows from Eq. (S10) that \( \|\hat{y}_t\| \leq L_0 S(t) \|\theta, x_0\| = \mathcal{O}(S(t)) \). The more general case, for any invariant set, follows from a similar deduction.

### B.4 Proof of Theorem 1 (b)

It follows from Eq. (S2), and from property 2 in Lemma 2, that:
\[
\|\nabla V(\theta, x_0) - \nabla V(\phi, w_0)\|
\]
\[
\leq \frac{1}{N} \sum_{t=1}^N K_3 L_y \|\Delta J_t\| + \|\Delta J_t \Psi(\hat{y}_t)\|, 
\]  
(S13)
where we have used the notation \( \Delta J_t \) to denote the difference between \( J_t \) evaluated at \( (\theta, x_0) \) and \( (\phi, w_0) \). Analogously, \( \Delta J_t \Psi(\hat{y}_t) \) denotes the difference between \( J_t \Psi(\hat{y}_t) \) evaluated at the two distinct points, where \( \Psi \) is the Lipschitz continuous function with constant \( K_4 \) defined in Lemma 2.

From Eq. (S1) it follows that:
\[
J_t = C_t \sum_{t=1}^T \prod_{j=1}^{t-\ell} A_{t-j+1} B_t + F_t; 
\]  
(S14)
Since the Jacobian of \( f \) is Lipschitz with Lipschitz constant \( L_f \), it follows that:
\[
\|\Delta A_j\| \leq L_{J_f} \|s(j)\| \|\theta, x_0\| - [\phi, w_0]\|^T, 
\]  
(S15)
Using a procedure analogous to the one used to obtain Eq. (S10), it follows that:
\[
\|\Delta A_j\| \leq L_{J_f} \ell_0 \|s(j)\| \|\theta, x_0\|^T - [\phi, w_0]\|^T, 
\]  
(S16)
where \( s(j) \) is defined as in Eq. (S11). An identical formula holds for \( B_j \) and a similar formula, replacing \( L_{J_f} \) with \( L_{J_g} \), holds for \( C_j \).

Since \( f \) and \( g \) are Lipschitz with Lipschitz constants \( L_f \) and \( L_g \) it follows that \( \|A_j\| \leq L_f, \|B_j\| \leq L_f, \|C_j\| \leq L_g \) and \( \|F_j\| \leq L_g \). Hence, it follows from Eq. (S10), Eq. (S14), Eq. (S16) and the repetitive application of Lemma 1 that \( \|\Delta J_t\| \) and \( \|\Delta J_t \Psi(\hat{y}_t)\| \) are upper bounded by \( \|\theta, x_0\|^T - [\phi, w_0]\|^T \) multiplied by the following constants:
\[
L_{J_{f,t}} = \sum_{t=1}^{T} P(t, t) + L_{J_f} S(t),
\]
\[
L_{J_{g,t}} = \sum_{t=1}^{T} Q(t, t) + T(t) S(t),
\]
where \( T(t) = K_4 L_y^T M(t) + L_g^T \) and:
\[
P(t, t) = L_{J_f}^{T-t} \left( L_{J_f} L_y S(j) + L_{J_f} L_g^T S(j) \right),
\]
\[
Q(t, t) = L_{J_g}^{T-t} \left( K_4 M(t) L_y L_{J_g}^T + L_{J_g} L_g S(j) + L_{J_g} T(t) S(t) \right).
\]
Hence:
\[
\|\nabla V(\theta, x_0) - \nabla V(\phi, w_0)\| \leq L_{V} \|\theta, x_0\|^T - [\phi, w_0]\|^T,
\]
where:
\[
L_{V} = \frac{1}{N} \sum_{t=1}^N (K_3 L_y L_{J_{f,t}} + L_{J_{g,t}}).
\]
Combining the above, the asymptotic analysis of \( L_{V} \) results in Eq. (7).
C Chaotic LSTM example

An LSTM with zero input and without bias terms is considered:

\[
c_t = \sigma (W_{hf} h_{t-1}) + c_{t-1} + \sigma (W_{hi} h_{t-1}) \ast \tanh (W_{hg} h_{t-1})
\]

\[
h_t = \sigma (W_{ho} h_{t-1}) \ast \tanh(c_t).
\]

(S17)

(S18)

The sigmoids \(\sigma(\cdot)\) and hyperbolic tangents \(\tanh(\cdot)\) operate element-wise. The symbol \(\ast\) indicates an element-wise product. The hidden and cell state have initial conditions \(h_0 = c_0 = [0.5 \ 0.5]^T\). The hidden state \(h_t\) is also the output of the model. The weight matrices are put equal to

\[
W_{hi} = \begin{bmatrix} -1 & 4 \\ -3 & -2 \end{bmatrix}, \quad W_{hf} = \begin{bmatrix} -2 & 6 \\ 0 & -6 \end{bmatrix}, \quad W_{hg} = \begin{bmatrix} -1 & -6 \\ 6 & -9 \end{bmatrix},
\]

and \(W_{ho} = \begin{bmatrix} 4 & 1 \\ 9 & 7 \end{bmatrix}\) to generate the data. The values in the weight matrices are stacked on top of each other in a parameter vector \(\theta_{true}\).

Figure 1 shows the mean-square loss evaluated on data generated by the same LSTM model with the same initial conditions, but with different parameter values. A two-dimensional grid of parameter values \(\theta(s_1, s_2)\) is used with values interpolated (and extrapolated) between \(\theta_{true}\), zero parameter values, and a randomly chosen parameter vector \(\theta_{random}\), i.e. in this case, \(\theta(s_1, s_2) = s_1\theta_{true} + s_2\theta_{random}\). Again, the region in the parameter space around \(\theta_{true}\) is intricate. There is an entire region where the cost function is intricate and has many neighboring local minima.
D Additional Experiments

Supplementary Figure 1: Chaotic LSTM. Mean-square cost function (2) for LSTM models with parameter vectors \( \theta(s_1, s_2) = s_1 \theta_{\text{true}} + s_2 \theta_{\text{random}} \).

Supplementary Figure 2: The effect of the initial conditions. Accuracy on training and validation data on the symbol classification task for the LSTM model in identical scenarios but with different initial random parameter initialization (from different random seeds). In (a), the optimization procedure abruptly finds a solution that has good accuracy on both training and validation; while, in (b), the convergence is slow and steady towards a solution that has good accuracy on the training set (above 90%) but is no better than random guessing on the validation set.
Supplementary Figure 3: The LSTM learns to classify sequences. Bifurcation diagram for the LSTM model in symbol classification task for sequences of length 100. It shows the steady-state of the output $y_t$ and its first difference $y_t - y_{t-1}$. The arrows point towards the evolution of the number of epochs, that vary from 0 to 400.
Supplementary Figure 4: The oRNN learns to classify sequences. Bifurcation diagram for the oRNN model in symbol classification task for sequences of length 100. It shows the steady-state of the output $y_t$ and its first difference $y_t - y_{t-1}$. The arrows point towards the evolution of the number of epochs, that vary from 0 to 400.
Supplementary Figure 5: **The LSTM mechanism for learning a language model.** Bifurcation diagram for the LSTM world-level language model. For each epoch, the plot shows values visited by the projections of the internal state \( p(x_t) \) and its first difference \( p(x_t) - p(x_{t-1}) \) after a burnout period of 1500 samples. This burnout period is used to remove the transient response and yields a visualization of the system attractors, per epoch. The arrow point towards the evolution of the number of epochs, that varies from 0 to 150. In (a) and (b), we have two different realizations of the bifurcation diagram obtained from constant inputs. In (c) and (d), the diagram is generated using as input the word predicted with the highest probability at the previous time instant, and using as first input to the sequence the same input as in (a) and (b), respectively. The subplots (a) to (d) use the average of internal states as projections, i.e. \( p(x_t) = \bar{x}_t \). The second row, (e) to (h), and third row, (i) to (l), show the exact same experiments but for the projections in the direction of the tokens “is” and “Valkyria”, respectively.
Supplementary Figure 6: The sLSTM mechanism for learning a language model. Bifurcation diagram for the stable LSTM world-level language model. For each epoch, the plot shows values visited by the projections of the internal state $p(x_t)$ after a burnout period of 1500 samples. This burnout period is used to remove the transient response and yields a visualization of the system attractors, per epoch. In the displays (a) to (d), the diagram is generated for the same constant input. In (e) to (h), the diagram is generated using as input the word predicted with the highest probability at the previous time instant, and using as first input to the sequence the same input as in the first row. The projections are: the average of internal states as projections, i.e. $p(x_t) = \bar{x}_t$; and, projections into the direction of the tokens “is”, “Valkyria” and “<unk>”.

Supplementary Figure 7: The oRNN mechanism for learning a language model. Bifurcation diagram for the orthogonal RNN world-level language model. For each epoch, the plot shows values visited by the projections of the internal state $p(x_t)$ after a burnout period of 1500 samples. This burnout period is used to remove the transient response and yields a visualization of the system attractors, per epoch. In the displays (a) to (d), the diagram is generated for the same constant input. In (e) to (h), the diagram is generated using as input the word predicted with the highest probability at the previous time instant, and using as first input to the sequence the same input as in the first row. The projections are: the average of internal states as projections, i.e. $p(x_t) = \bar{x}_t$; and, projections into the direction of the tokens “is”, “Valkyria” and “<unk>”.