Learning nonlinear differentiable models for signals and systems: with application

Antônio Horta Ribeiro

Supervisor: Luis Antonio Aguirre

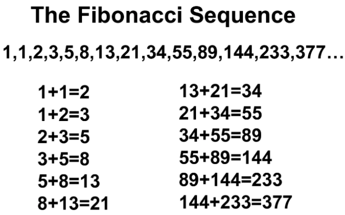
Co-supervisor: Thomas B. Schön

Presented before the committee as requirement for obtaing the doctoral degree in electrical engineering.

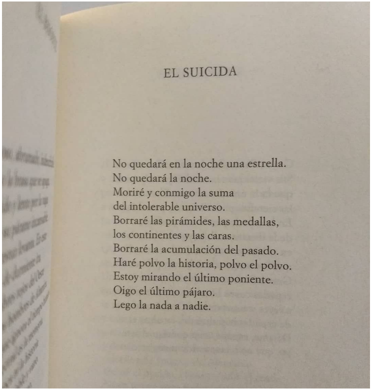
Universidade Federal de Minas Gerais

Brazil, 2020

Signals, systems and sequences



(a) Fibonacci



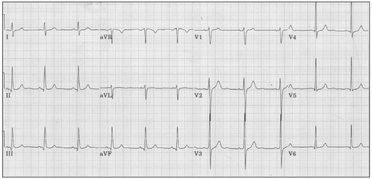
EL SUICIDA

No quedará en la noche una estrella.
No quedará la noche.
Moriré y conmigo la suma
del intolerable universo.
Borraré las pirámides, las medallas,
los continentes y las caras.
Borraré la acumulación del pasado.
Haré polvo la historia, polvo el polvo.
Estoy mirando el último poniente.
Oigo el último pájaro.
Lego la nada a nadie.

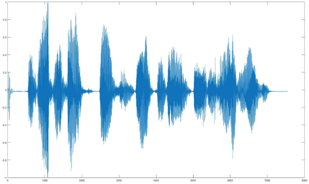
(b) Text

Figure: Sequence

Signals, systems and sequences



(a) Biomedical signal



(b) Speech signal

Figure: Signals

Signals, systems and sequences

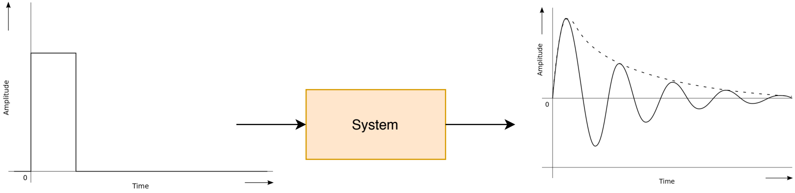


Figure: System

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;
- Predictive model: $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$.

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;
- Predictive model: $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$.
- Cost function:

$$V(\boldsymbol{\theta}) = \sum_{k=1}^{N} l(\mathbf{y}_k, \hat{\mathbf{y}}_k(\boldsymbol{\theta})).$$

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;
- Predictive model: $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$.
- Cost function:

$$V(\boldsymbol{\theta}) = \sum_{k=1}^{N} l(\mathbf{y}_k, \hat{\mathbf{y}}_k(\boldsymbol{\theta})).$$

- Iterative update based. Based on the derivatives.
  E.g. *Gradient descent*:

$$\boldsymbol{\theta}_n \leftarrow \boldsymbol{\theta}_{n-1} - \mu \nabla V(\boldsymbol{\theta}_{n-1})$$

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;
- Predictive model: $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$.
- Cost function:

$$V(\boldsymbol{\theta}) = \sum_{k=1}^{N} l(\mathbf{y}_k, \hat{\mathbf{y}}_k(\boldsymbol{\theta})).$$

- Iterative update based. Based on the derivatives.
  E.g. *Gradient descent*:

$$\boldsymbol{\theta}_n \leftarrow \boldsymbol{\theta}_{n-1} - \mu \nabla V(\boldsymbol{\theta}_{n-1})$$

- Batch vs stochastic

Training models in optimization-based framework

- Observed output: $\mathbf{y}_k$;
- Predictive model: $\hat{\mathbf{y}}_k(\boldsymbol{\theta})$.
- Cost function:

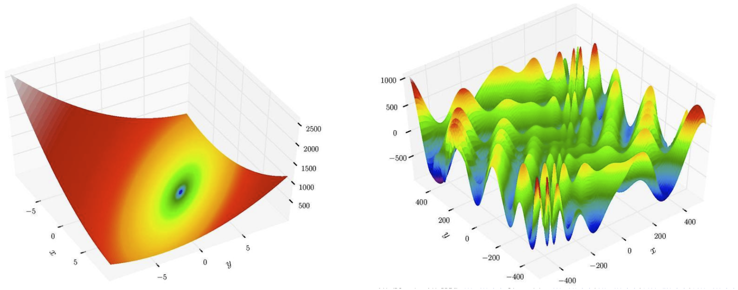
$$V(\boldsymbol{\theta}) = \sum_{k=1}^{N} l(\mathbf{y}_k, \hat{\mathbf{y}}_k(\boldsymbol{\theta})).$$

- Iterative update based. Based on the derivatives.
  E.g. *Gradient descent*:

$$\boldsymbol{\theta}_n \leftarrow \boldsymbol{\theta}_{n-1} - \mu \nabla V(\boldsymbol{\theta}_{n-1})$$

- Batch vs stochastic
- Derivatives computed using automatic differentiation.

Nonlinear systems and non-convex cost function



(a) convex (b) non-convex

Overview

ECG Automatic diagnosis using a deep neural network

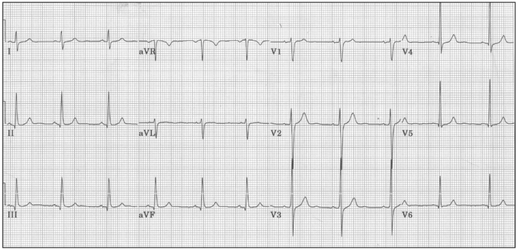
Deep convolutional networks in system identification

Parallel training considered harmful?

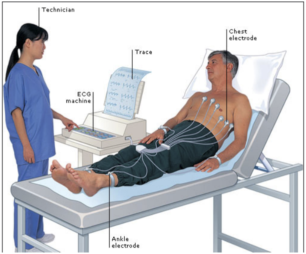
Multiple shooting

Analysing RNN training using attractors and smoothness

The 12-lead electrocardiogram (ECG)



(a) ECG signal



(b) ECG placement

Figure: The 12-lead electrocardiogram exam.

Telehealth network of Minas Gerais

Year	# Municipalities
2006	82
2007	102
2008	97
2009	328
2011	54
2013	106
2015	42
Total	811

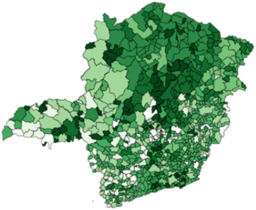
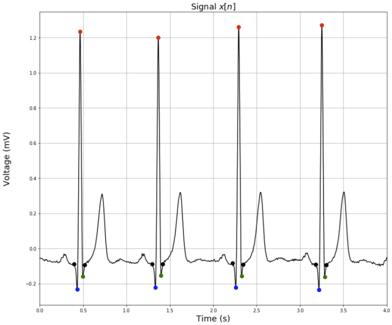


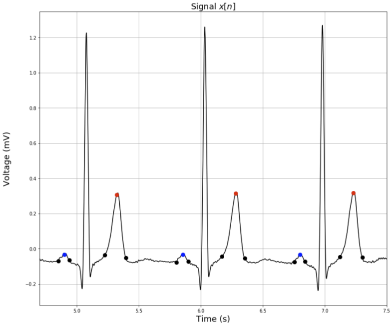
Figure: Telehealth in Minas Gerais

M. B. Alkmim, R. M. Figueira, M. S. Marcolino, C. S. Cardoso, M. Pena de Abreu, L. R. Cunha, D. F. da Cunha, A. P. Antunes, A. G. d. A. Resende, E. S. Resende, and A. L. P. Ribeiro, "Improving patient access to specialized health care: The Telehealth Network of Minas Gerais, Brazil," *Bulletin of the World Health Organization*, vol. 90, no. 5, pp. 373–378, May 2012, ISSN: 1564-0604. DOI: 10/f3x7px.

ECG segmentation



(a) QRS complex



(b) T and P waves

Figure: Two-step automated analysis of the electrocardiogram..

Classical EGG automated analysis

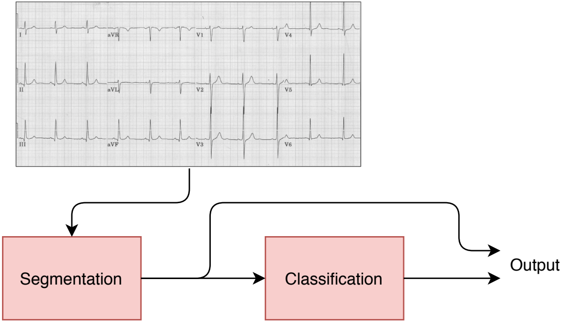


Figure: Peaks and wave lenghts.

P. W. Macfarlane, B. Devine, and E. Clark, "The university of glasgow (Uni-G) ECG analysis program," in *Computers in Cardiology*, 2005, pp. 451–454, ISBN: 0276-6574. DOI: 10.1109/CIC.2005.1588134.

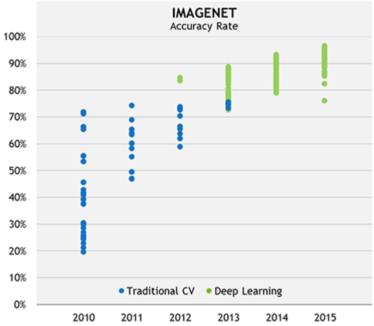
Deep neural networks

Yoshua Bengio, Geoffrey Hinton and Yann LeCun *"for conceptual and engineering breakthroughs that have made* **deep neural networks** *a critical component of computing."*

– Turing award (2018)

Image classification with deep neural networks

(a) Samples



(b) Accuracy

Figure: The imagenet image classification benchmark.

The CODE group

Figure: The CODE (*Clinical outcomes in eletrocardiography*) group was created to conduct clinical studies using storical data from the telehealth network.

Automatic diagnosis of the 12-lead ECG using a deep neural network

Antônio H. Ribeiro, Manoel Horta Ribeiro, Gabriela M.M. Paixão, Derick M. Oliveira, Paulo R. Gomes, Jéssica A. Canazart, Milton P. S. Ferreira, Carl R. Andersson, Peter W. Macfarlane, Wagner Meira Jr., Thomas B. Schön, Antonio Luiz P. Ribeiro
Automatic diagnosis of the 12-lead ECG using a deep neural network
Nature Communication, 2020.

Deep neural network for automatic ECG analysis

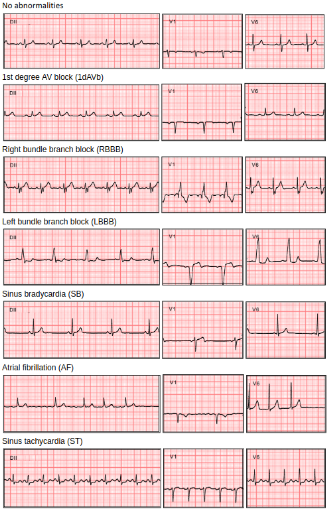


Figure: Abnormalities to classify. We show only 3 representative leads (DII, V1 and V6).

The training dataset

- 2.3 million records;

The training dataset

- 2.3 million records;
- 1.6 million distinct patients;

The training dataset

- 2.3 million records;
- 1.6 million distinct patients;
- Annotated by telehealth center cardiologist;

The training dataset

- 2.3 million records;
- 1.6 million distinct patients;
- Annotated by telehealth center cardiologist;
- Refined by comparing with University of Glasgow software results;

The training dataset

- 2.3 million records;
- 1.6 million distinct patients;
- Annotated by telehealth center cardiologist;
- Refined by comparing with University of Glasgow software results;
- 30 000 exams manually reviewed.

The model

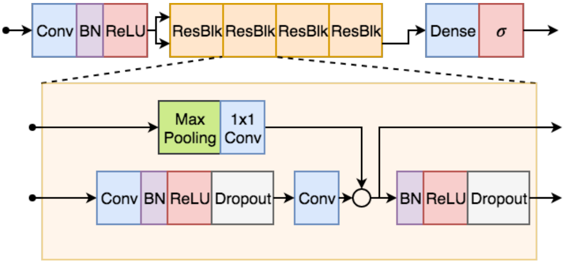
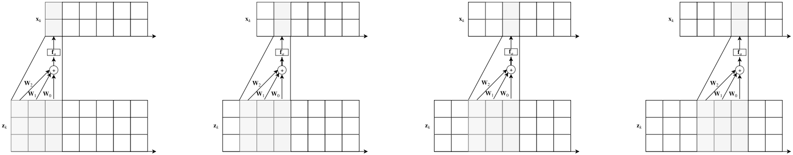


Figure: The uni-dimensional residual neural network architecture used for ECG classification.

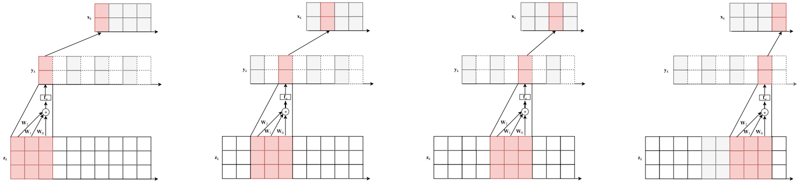
Convolutions



(a) (b) (c) (d)

Figure: Simplified diagram illustrating convolutions.

Subsampling and strides



(a) (b) (c) (d)

Figure: Strides. Convolution followed by subsampling.

Convolutional neural network

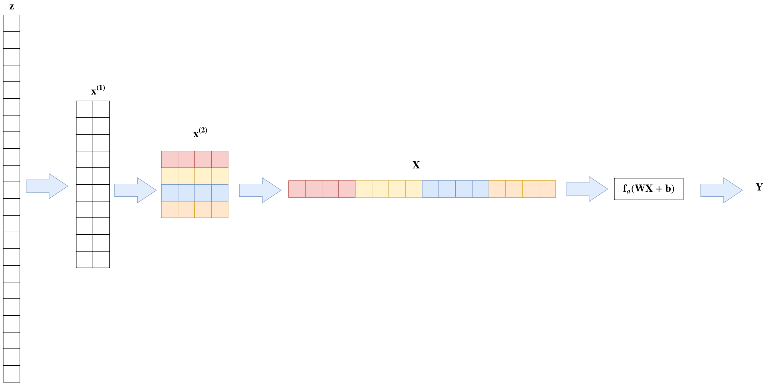


Figure: Simplified convolutional neural network.

Residual neural networks

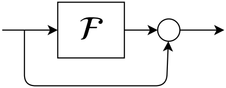


Figure: Residual connection.

K. He, X. Zhang, S. Ren, and J. Sun (2016),
Identity Mappings in Deep Residual Networks,
Computer Vision – ECCV, pp. 630–645, Springer International Publishing, 2016.

The testing dataset

- 827 tracings from distinct patients;
- Annotated by 3 different cardiologists;
- The 2017 physionet challenge has a testset of 3658 ECGs from different patients (4.4 times the size of our dataset).

Results

	F1 Score			
	DNN	cardio.	emerg.	stud.
1dAVb	**0.897**	0.776	0.719	0.732
RBBB	**0.944**	0.917	0.852	0.928
LBBB	**1.000**	0.947	0.912	0.915
SB	**0.882**	**0.882**	0.848	0.750
AF	**0.870**	0.769	0.696	0.706
ST	**0.960**	0.882	0.946	0.873

Table: Performance indexes

Discussion

- Potential to improve tele-health service in short/medium term;
- Screen more important exams;
- Avoid medical mistakes and improve accuracy.

"Why not a recurrent neural network?"

The fall of recurrent neural networks

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).
Attention is All you Need.
Advances in Neural Information Processing Systems 30, pages 5998–6008.

Convolution neural networks for sequence modeling

S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," en, p. 14, 2018.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499 [cs]*, Sep. 2016. arXiv: 1609.03499 [cs].

Deep convolutional networks in system identification

Carl Andersson*, Antonio H. Ribeiro*, Koen Tiels, Niklas Wahlström and Thomas B. Schön (* Equal contribution).
Deep Convolutional Networks in System Identification
To appear in the proceedings of the 58th IEEE Conference on Decision and Control (CDC), 2019.

System identification

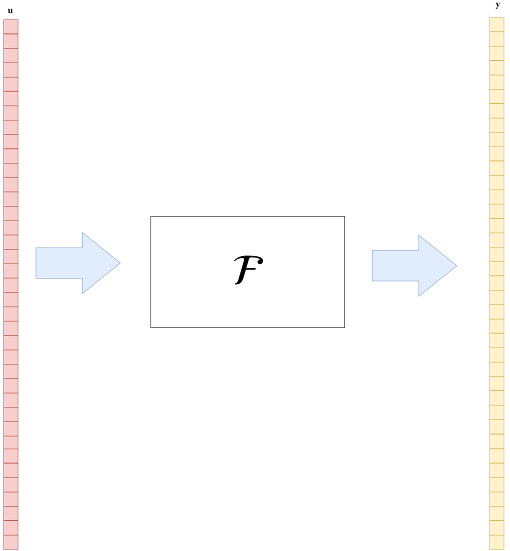


Figure: Learning input-output relations between signals.

Convolutions for capturing input-output relations

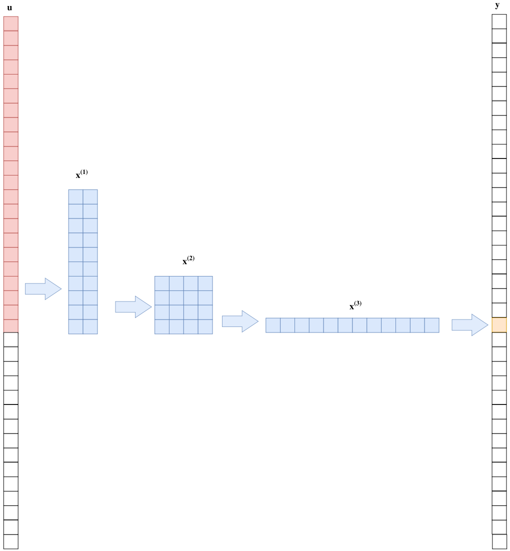


Figure: Learning input-output relations between signals.

Convolutions for capturing input-output relations

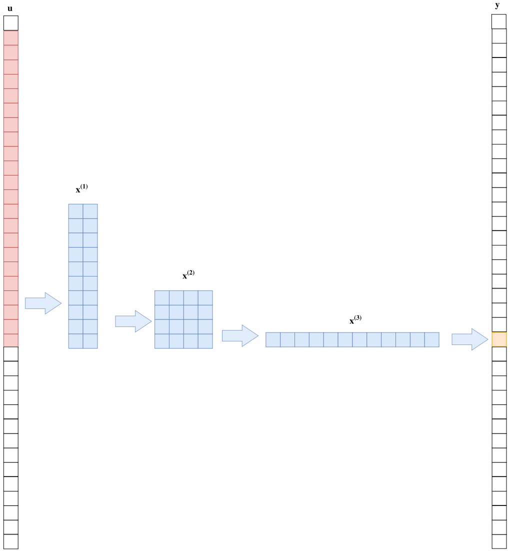


Figure: Learning input-output relations between signals.

Convolutions for capturing input-output relations

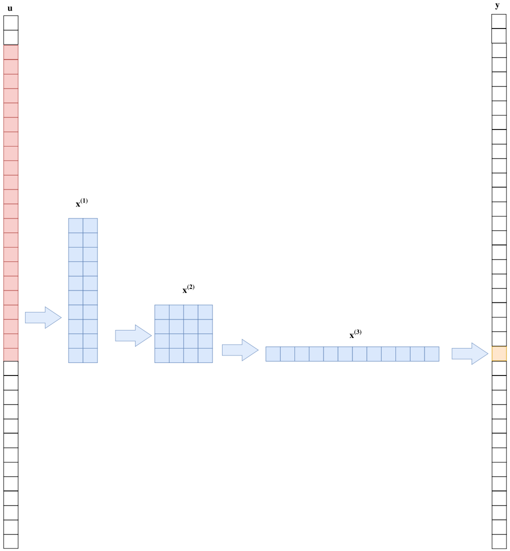


Figure: Learning input-output relations between signals.

Capturing input-output relations using dilations

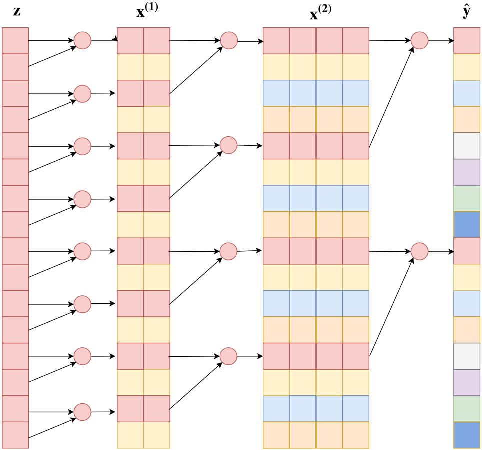


Figure: Convolutional network modeling input-output relations **using dillations**.

Autorregressive terms

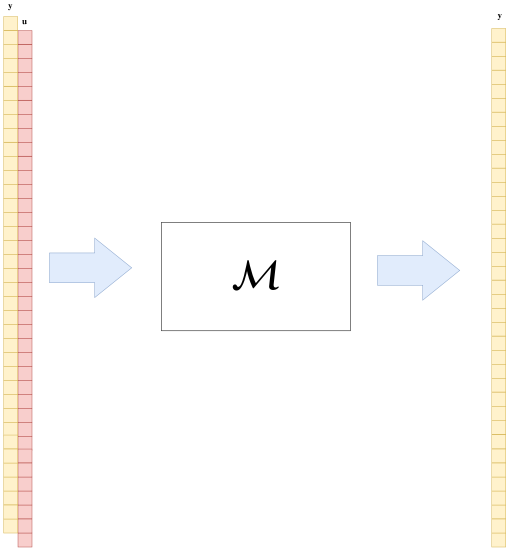


Figure: Learning input-output relations between signals **using autorregressive term**.

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory
- Potential to provide good results in sys. id. (even if this requires us to rethink these models).

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory
- Potential to provide good results in sys. id. (even if this requires us to rethink these models).
- Traditional deep learning tricks did not always improve performance.

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory
- Potential to provide good results in sys. id. (even if this requires us to rethink these models).
- Traditional deep learning tricks did not always improve performance.
  - Dilation (exponential decay of dynamical systems)

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory
- Potential to provide good results in sys. id. (even if this requires us to rethink these models).
- Traditional deep learning tricks did not always improve performance.
  - Dilation (exponential decay of dynamical systems)
  - Dropout

Summary of the results

- 3 examples: 1 toy problem; 2 nonlinear system identification benchmarks (Oscillatory circuit Silverbox, F16 ground vibration test);
- Mixed results: convolutional network often worse than vanilla multilayer perceptron and long-short term memory
- Potential to provide good results in sys. id. (even if this requires us to rethink these models).
- Traditional deep learning tricks did not always improve performance.
  - Dilation (exponential decay of dynamical systems)
  - Dropout
  - Depth

One-step-ahead *vs* free-run simulation

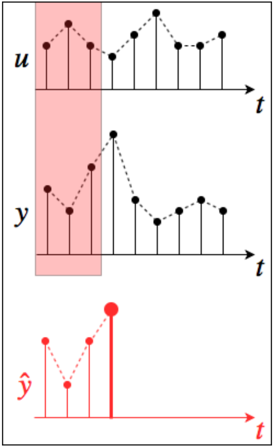


Figure: One-step-ahead prediction

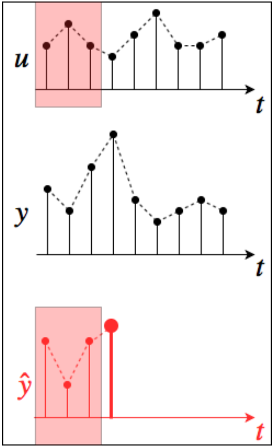


Figure: Free run simulation

One-step-ahead *vs* free-run simulation

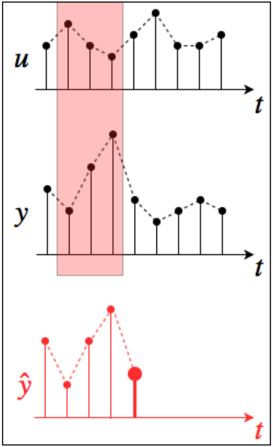


Figure: One-step-ahead prediction

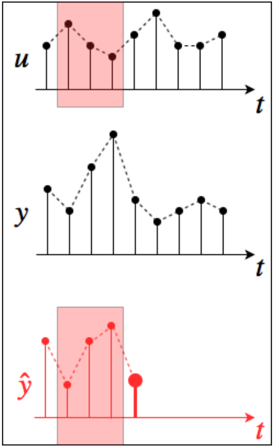


Figure: Free run simulation

One-step-ahead *vs* free-run simulation

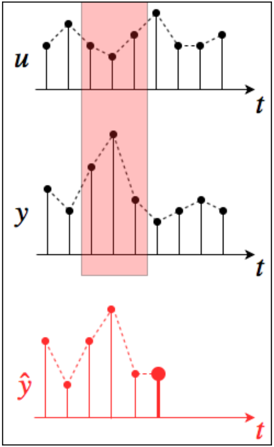


Figure: One-step-ahead prediction

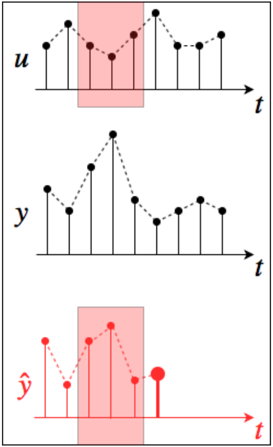


Figure: Free run simulation

One-step-ahead *vs* free-run simulation

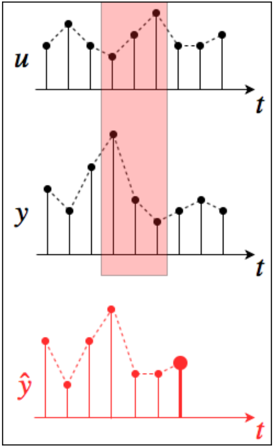


Figure: One-step-ahead prediction

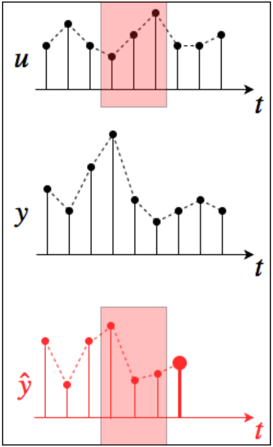


Figure: Free run simulation

One-step-ahead *vs* free-run simulation

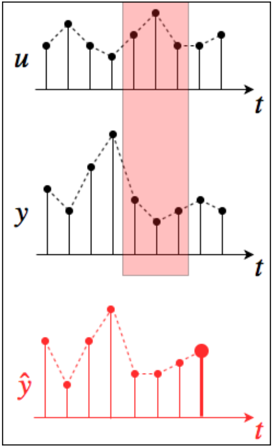


Figure: One-step-ahead prediction

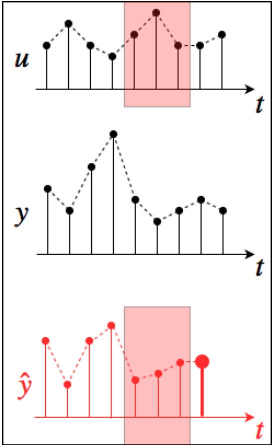


Figure: Free run simulation

One-step-ahead *vs* free-run simulation

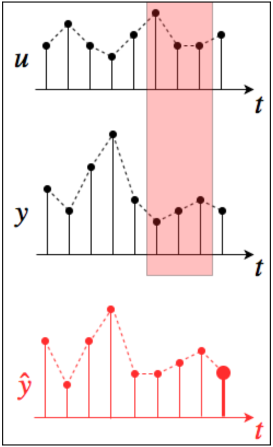


Figure: One-step-ahead prediction

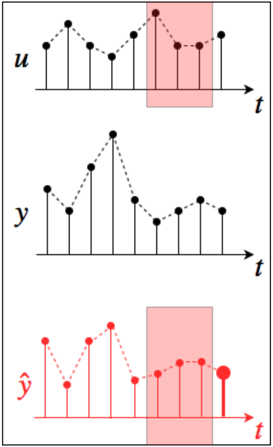


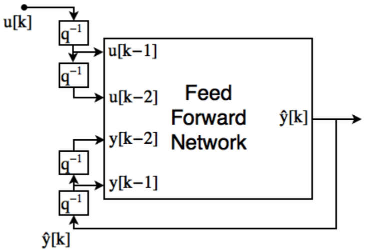
Figure: Free run simulation

"Parallel training considered harmful?"

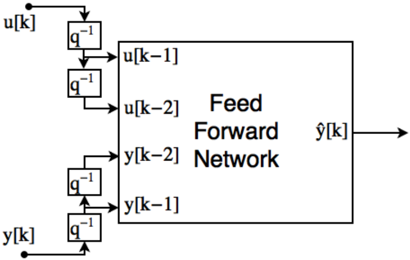
Ribeiro, A. H. and Aguirre, L. A. (2018).
"Parallel Training Considered Harmful?": Comparing Series-Parallel and Parallel Feedforward Network Training.
Neurocomputing, v. 316 (17) pp. 222-231.
doi: 10.1016/j.neucom.2018.07.071

Parallel *vs* Series-Parallel Training

Parallel mode



Series-parallel model



Literature review

Narendra, K. S. and Parthasarathy, K. (1990).
Identification and control of dynamical systems using neural networks.
IEEE Transactions on Neural Networks, 1(1):4–27.

Beale, M. H., Hagan, M. T., and Demuth, H. B. (2017).
Neural network toolbox for use with MATLAB.
Technical report, Mathworks.

Example 1: pilot plant

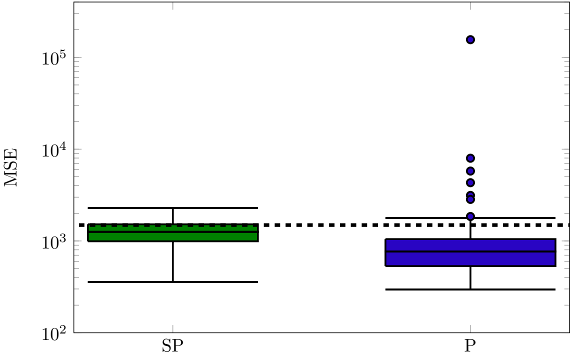


Figure: Boxplots show the distribution of the free-run simulation MSE over the validation window for models trained using series-parallel (SP) and parallel (P) for 100 realizations (changing only the initial parameter guess).

Example 2: toy problem

The nonlinear system:

$$
\begin{aligned}
y^*[k] &= (0.8 - 0.5e^{-y^*[k-1]^2})y^*[k-1] - \\
&\quad (0.3 + 0.9e^{-y^*[k-1]^2})y^*[k-2] + u[k-1] + \\
&\quad 0.2u[k-2] + 0.1u[k-1]u[k-2] + v[k], \\
y[k] &= y^*[k] + w[k],
\end{aligned}
$$

S. Chen, S. A. Billings, and P. M. Grant (1990)
Non-linear system identification using neural networks
International Journal of Control, vol. 51, no. 6, pp. 1191-1214,

Example 2: validation results

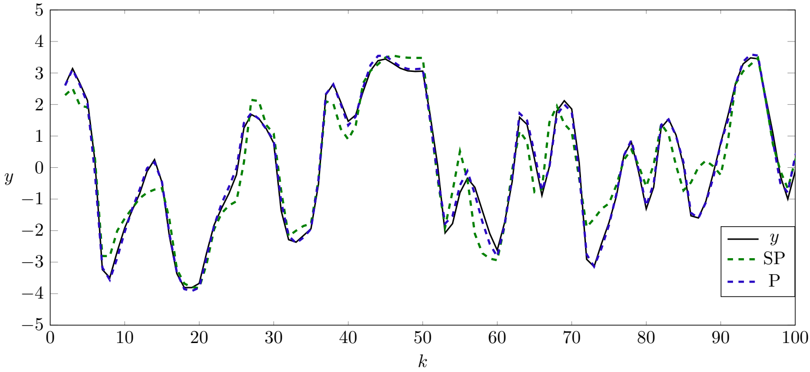
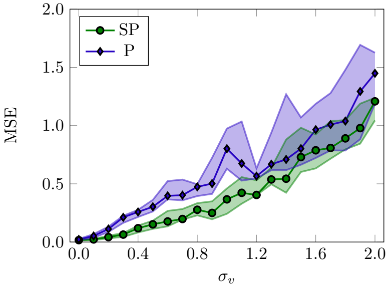
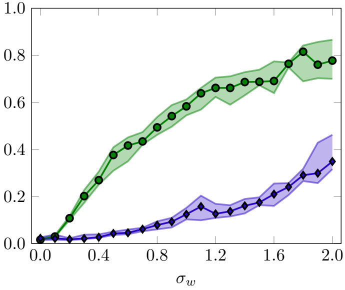


Figure: Displays the first 100 samples of the free-run simulation in the validation window for models trained using series-parallel (SP) and parallel (P) methods. $\mathrm{MSE}_{\mathrm{SP}} = 0.39$; $\mathrm{MSE}_{\mathrm{P}} = 0.06$.

Example 2: white noise effect



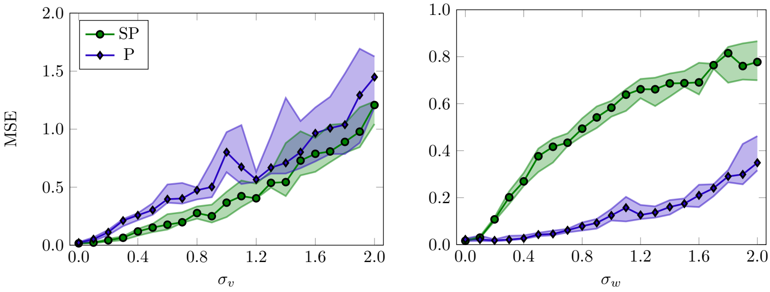
(a) White process error



(b) White output error

Figure: Free-run simulation MSE over the validation window *vs* noise levels.

Example 2: white noise effect



(a) White process error (b) White output error

Figure: Free-run simulation MSE over the validation window *vs* noise levels.

Ljung, L. (1978).
Convergence analysis of parametric identification methods.
IEEE Transactions on Automatic Control, 23(5):770–783.

Parallel *vs* series-parallel

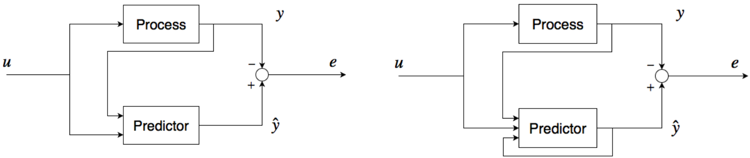
- Series-parallel and parallel training have different properties and are better depending on the noise type;
- Similar computational cost can be attained. However, parallel training is difficult to parallelize;
- Parallel training is more dependent on initial optimization conditions.

On the smoothness of nonlinear system identification

Antônio H. Ribeiro and Luis A. Aguirre (2017).
Shooting Methods for Parameter Estimation of Output Error Models.
IFAC-PapersOnLine, v. 50. p. 13998-14003. In: IFAC World Congress, 2017, Toulouse, France.

Antônio H. Ribeiro, Koen Tiels, Jack Umenberger, Thomas B. Schön, Luis A. Aguirre (2020).
On the Smoothness of Nonlinear System Identification
Automatica, provisionally accepted.

Feedforward *vs* recurrent structures



(a) Feedforward (b) Recurrent

Ljung, L. (1978).
Convergence analysis of parametric identification methods.
IEEE Transactions on Automatic Control, 23(5):770–783.

Single shooting *vs* multiple shooting

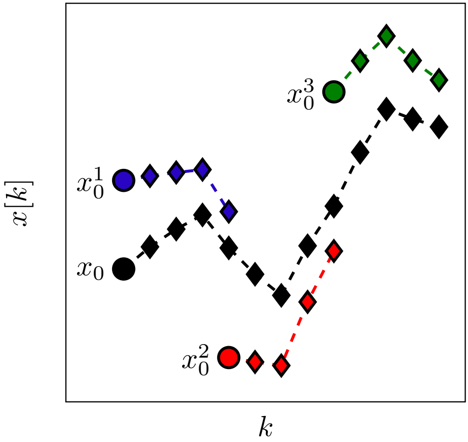
Single Shooting

$$\min V.$$

Multiple Shooting

$$\min \sum_i V_i,$$

$$\text{s.t.: } \mathbf{x}^{i-1}[\mathrm{m}_i] = \mathbf{x}_0^i, \forall i.$$



Single shooting *vs* multiple shooting

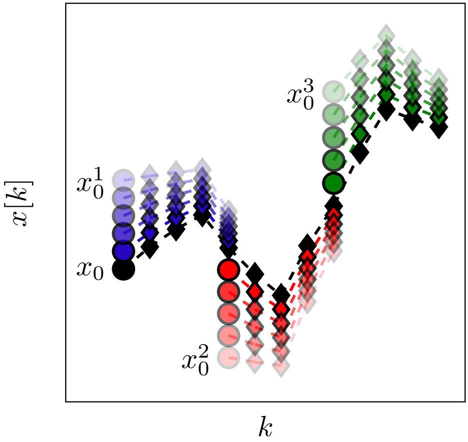
Single Shooting

$$\min V.$$

Multiple Shooting

$$\min \sum_i V_i,$$

$$\text{s.t.: } \mathbf{x}^{i-1}[\mathrm{m}_i] = \mathbf{x}_0^i, \forall i.$$



Single shooting *vs* multiple shooting

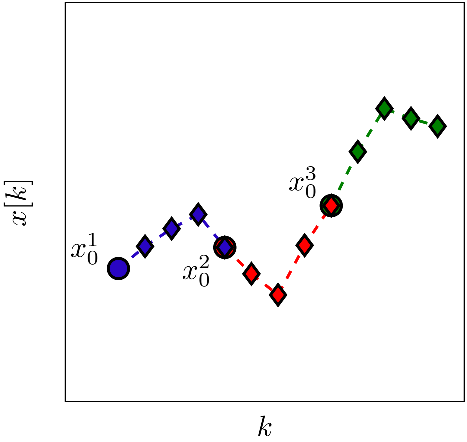
Single Shooting

$$\min V.$$

Multiple Shooting

$$\min \sum_i V_i,$$

$$\text{s.t.: } \mathbf{x}^{i-1}[\mathrm{m}_i] = \mathbf{x}_0^i, \forall i.$$



Example 1: output error model for chaotic system

Logistic map, generated for $\theta = 3.78$:

$$y[k] = \theta y[k-1](1 - y[k-1]).$$

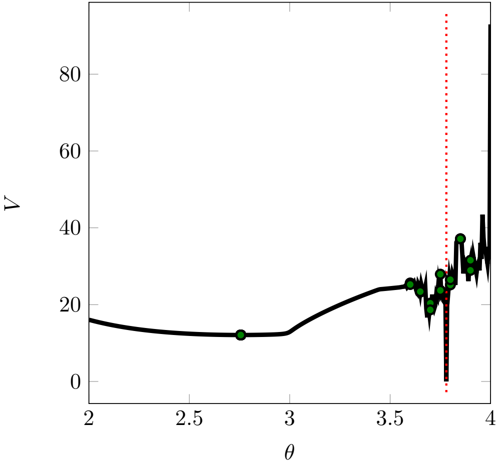


Figure: single shoot

Example 1: output error model for chaotic system

Logistic map, generated for $\theta = 3.78$:

$y[k] = \theta y[k-1](1 - y[k-1])$.

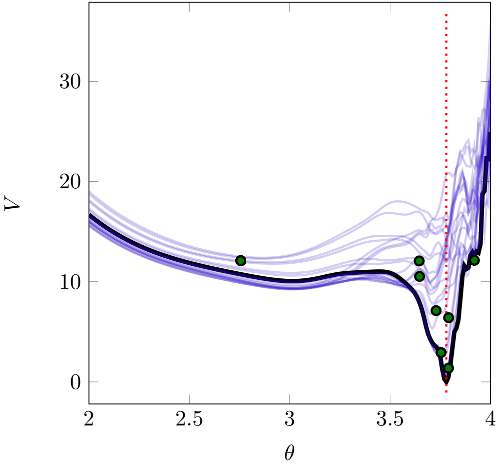


Figure: shoot length = 10

Example 1: output error model for chaotic system

Logistic map, generated for $\theta = 3.78$:

$$y[k] = \theta y[k-1](1 - y[k-1]).$$

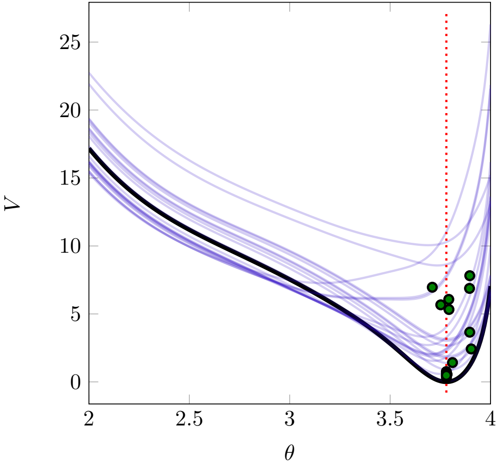


Figure: shoot length = 5

Example 1: output error model for chaotic system

Logistic map, generated for $\theta = 3.78$:

$$y[k] = \theta y[k-1](1 - y[k-1]).$$

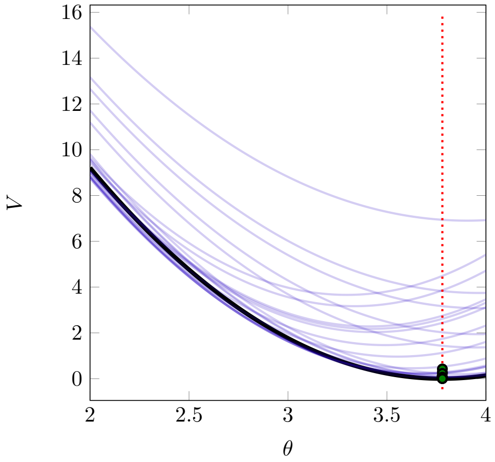


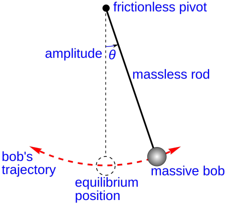
Figure: shoot length $= 2$

Example 1: output error model for chaotic system

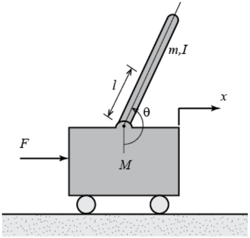
Table: Number of function evaluations (median) for different simulation lengths.

shoot len	f evals
200*	21.5
10	2000*
5	74
2	32

Example 2: pendulum

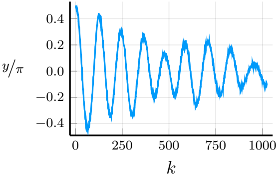


(a) Pendulum

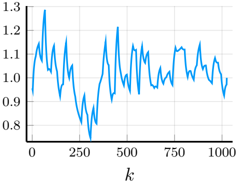


(b) Inverted pendulum

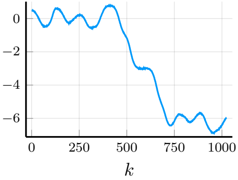
(c) 360 pendulum



(d) Pendulum



(e) Inverted pendulum



(f) 360 pendulum

Figure: The same system in three dynamical regimes

Example 2: pendulum parameter estimation

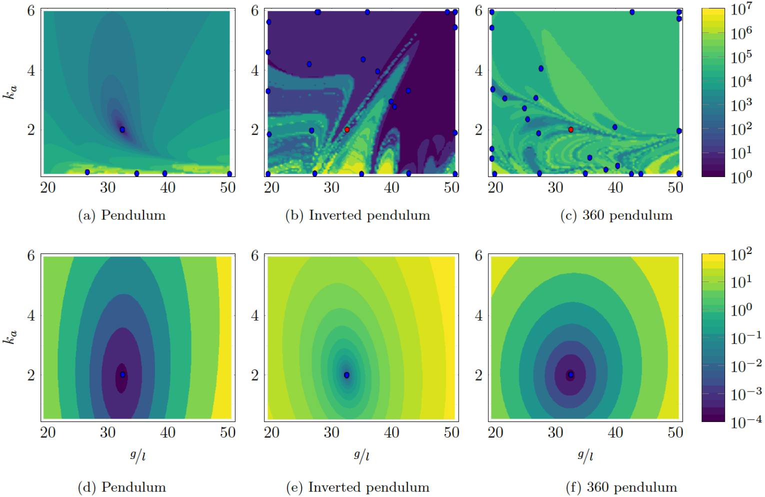


Figure: Contour plot of the cost function.

Beyond exploding and vanishing gradients

Antônio H. Ribeiro, Koen Tiels, Luis A. Aguirre and Thomas B. Schön. (2020)

Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness

To appear in the Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)

Recurrent neural network

RNNs are nonlinear discrete-time dynamical systems, and can be represented by the expression:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{z}_t; \boldsymbol{\theta}), \tag{1}$$

$$\hat{\mathbf{y}}_t = \mathbf{g}(\mathbf{x}_t, \mathbf{z}_t; \boldsymbol{\theta}), \tag{2}$$

za which are sufficiently general to capture vanilla RNNs, LSTM, GRU, and stacked layers of these units.

Exploding gradients

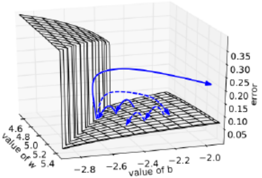


Figure: Wall in the cost function (Pascanu et al., 2013)

Pascanu, R., Mikolov, T., and Bengio, Y. (2013).
On the Difficulty of Training Recurrent Neural Networks.
In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML)*, pages 1310–1318.

Smoothness vs dynamic

Theorem

1. The cost function V is Lipschitz with constant:

$$L_V = \begin{cases} \mathcal{O}(L_f^{2N}) & \text{if } L_f > 1, \\ \mathcal{O}(N) & \text{if } L_f = 1, \\ \mathcal{O}(1) & \text{if } L_f < 1. \end{cases} \tag{3}$$

2. The gradient ∇V is Lipschitz with constant:

$$L'_V = \begin{cases} \mathcal{O}(L_f^{3N}) & \text{if } L_f > 1, \\ \mathcal{O}\left(N^3\right) & \text{if } L_f = 1, \\ \mathcal{O}\left(1\right) & \text{if } L_f < 1. \end{cases} \tag{4}$$

Contractive vs non-contractive

Definition: (Contractive) For some $L < 1$:

$$\|\mathbf{x}_{t+1} - \mathbf{w}_{t+1}\| < L\|\mathbf{x}_t - \mathbf{w}_t\|.$$

All contractive systems have a unique fixed point inside the contractive region $\Omega_{\mathbf{x}}$, and all trajectories converge to such a fixed point

Smoothness analysis

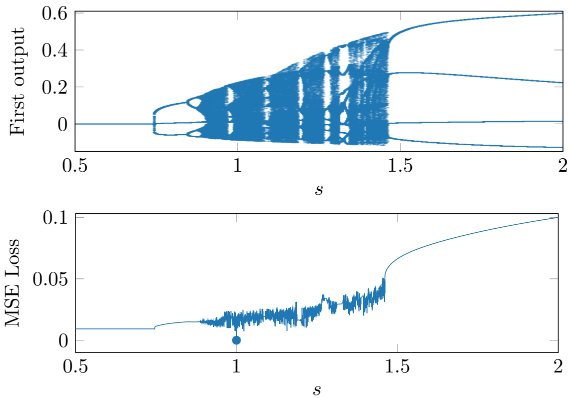


Figure: **Chaotic LSTM.** Display: a) Bifurcation diagram; and b) cost function (mean-square error) for LSTM models with parameter vectors $\theta(s) = s\theta_{\text{true}}$.

Example: classifying sequences based on a few relevant symbols

- Sequence containing categorical values $\{p, q, a, b, c, d\}$;

Example: classifying sequences based on a few relevant symbols

- Sequence containing categorical values $\{p, q, a, b, c, d\}$;
- Distractor symbols $\{a, b, c, d\}$;

Example: classifying sequences based on a few relevant symbols

- Sequence containing categorical values $\{p, q, a, b, c, d\}$;
- Distractor symbols $\{a, b, c, d\}$;
- Output: $\{p, p\}, \{p, q\}, \{q, p\}, \{q, q\}$.

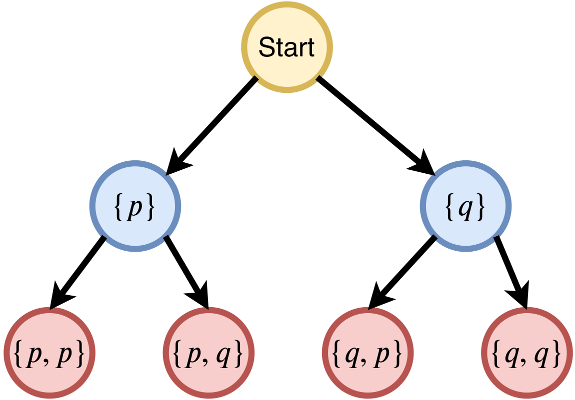
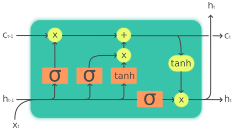
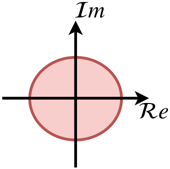


Figure: Finite state machine that needs to be implemeted to solve the problem.

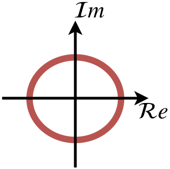
RNN architectures



(a) Long-short term memory (LSTM)



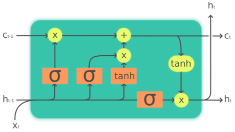
(b) Stable LSTM



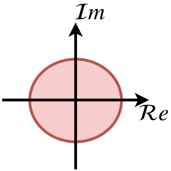
(c) Orthogonal RNN

Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural Computation, 9(8):1735–1780.

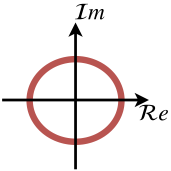
RNN architectures



(a) Long-short term memory (LSTM)



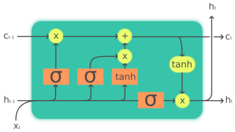
(b) Stable LSTM



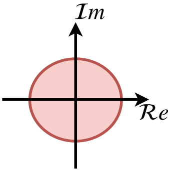
(c) Orthogonal RNN

Miller, J. and Hardt, M. (2019).
Stable Recurrent Models.
In *Proceedings of the 7th International Conference for Learning Representations (ICLR)*.

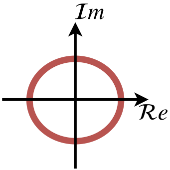
RNN architectures



(a) Long-short term memory (LSTM)



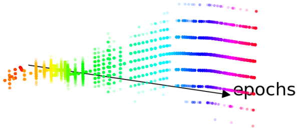
(b) Stable LSTM



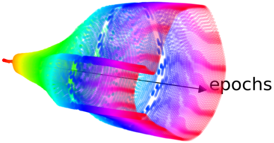
(c) Orthogonal RNN

Lezcano-Casado, M. and Martínez-Rubio, D. (2019).
Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group.
In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3794–3803.

Example: learning attractors

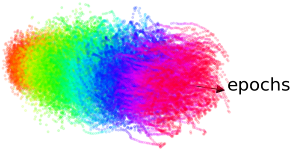


(a) LSTM, $p \to \{p, p\}$



(b) LSTM, $b \to \{p, p\}$

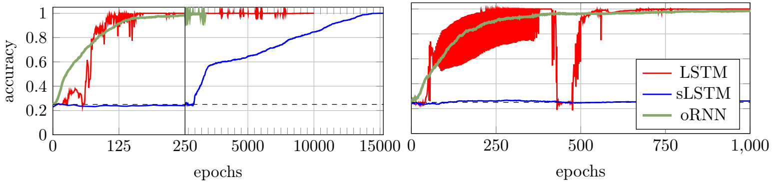
(c) oRNN, $p \to \{p, p\}$



(d) oRNN, $b \to \{p, p\}$

Figure: **Learning to classify sequences.** Bifurcation diagram for the sequence classification task for sequences of length 100. It shows the steady-state of the output y_t and its first difference $y_t - y_{t-1}$.

Example: smoothness of the cost function



(a) Length 100

(b) Length 200

Figure: **Sequence classification training history.** Accuracy on validation data set for the recurrent models trained to perform the same sequence classification task for two different sequence lengths.

Language model

- Given the context, it tries to predict the next word:

Figure: Example of phrase it can try to predict.

- We train a language model on the openly available dataset Wikitext-2 (Merity et al., 2017). This dataset contains 600 Wikipedia articles for training (2,088,628 tokens), 60 articles for validation (217,646 tokens), and 60 articles for testing (245,569 tokens)

Example: training history

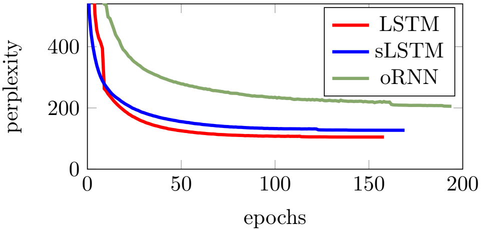
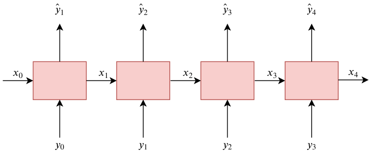
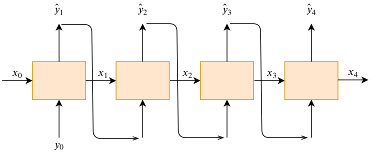


Figure: **Word-level language model training history.** Perplexity on validation data per epoch.

Teacher forcing

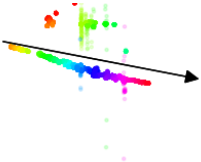


(a) Training

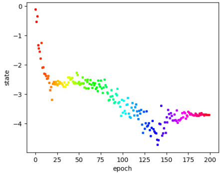


(b) Inference

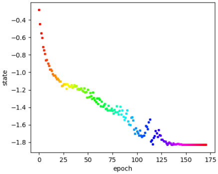
Example: learning attractors



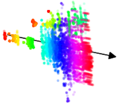
(a) LSTM



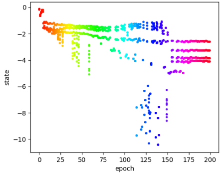
(b) oRNN



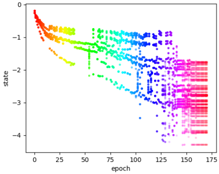
(c) sLSTM



(d) LSTM,
with feedback



(e) oRNN,
with feedback



(f) sLSTM,
with feedback

Figure: The same system in three dynamical regimes

Discussion

- Vanishing gradients *vs* attractors;

Discussion

- Vanishing gradients *vs* attractors;
- Attractor and the non-contractive region;

Discussion

- Vanishing gradients *vs* attractors;
- Attractor and the non-contractive region;
- Teacher forcing;

Discussion

- Vanishing gradients *vs* attractors;
- Attractor and the non-contractive region;
- Teacher forcing;
- Are RNN toy problems well designed?

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.
- It should enable new application and allow us to revisit old ones.

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.
- It should enable new application and allow us to revisit old ones.
- What is the role of recurrence?

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.
- It should enable new application and allow us to revisit old ones.
- What is the role of recurrence?
- Excitement with RNN in 2015 preceded the current wave of interest in feedforward architectures.

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.
- It should enable new application and allow us to revisit old ones.
- What is the role of recurrence?
- Excitement with RNN in 2015 preceded the current wave of interest in feedforward architectures.
  - *"The Unreasonable Effectiveness of Recurrent Neural Networks"* (Andrej Karpathy, now Director of AI Tesla).

Conclusion

- Differentiable nonlinear models (including deep learning) are powerful tools.
- It should enable new application and allow us to revisit old ones.
- What is the role of recurrence?
- Excitement with RNN in 2015 preceded the current wave of interest in feedforward architectures.
  - *"The Unreasonable Effectiveness of Recurrent Neural Networks"* (Andrej Karpathy, now Director of AI Tesla).
- System theory, control theory and signal processing should play a key role in the analysis;