Beyond Occam's Razor in System Identification: Double-Descent when Modeling Dynamics

Antônio H. Ribeiro[1], **Johannes N. Hendriks**[2], **Adrian G. Wills**[2], **Thomas B. Schön**[1]

[1]Uppsala University, Sweden
[2]The University of Newcastle, Australia

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

UPPSALA UNIVERSITET

Workshop on Nonlinear System Identification Benchmarks
2021

Neural network performance *vs* size

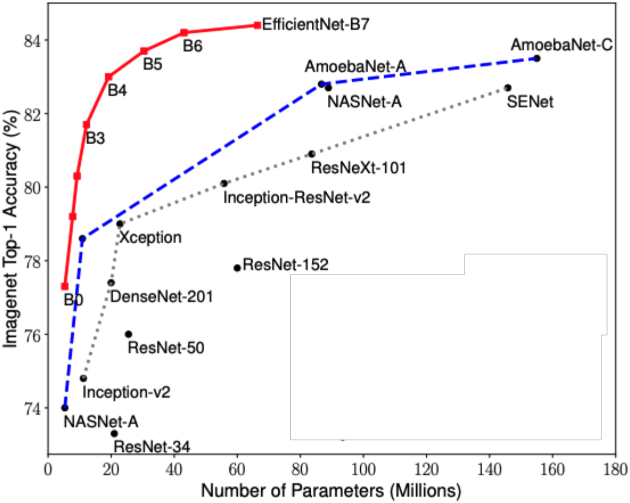
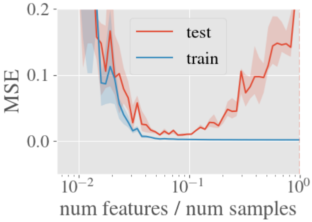


Figure: **Model Size vs. imagenet accuracy.**

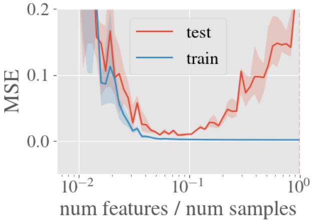
Double-descent



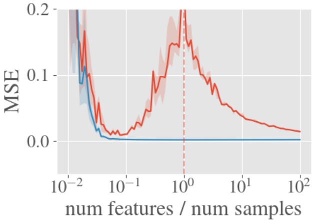
(a) U-shaped MSE

Figure: **Perform in CE8 Benchmark.** We show one-step-ahead prediction error in test and training data for a nonlinear ARX model in the CE8 benchmark.

Double-descent



(a) U-shaped MSE



(b) Double-descent

Figure: **Perform in CE8 Benchmark.** We show one-step-ahead prediction error in test and training data for a nonlinear ARX model in the CE8 benchmark.

Related work and historical development

- Random Fourier features, random forest and shallow networks:

  Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). *Reconciling modern machine-learning practice and the classical bias–variance trade-off.* Proceedings of the National Academy of Sciences, 116(32), 15849–15854.

Related work and historical development

- Random Fourier features, random forest and shallow networks:

  Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). *Reconciling modern machine-learning practice and the classical bias–variance trade-off.* Proceedings of the National Academy of Sciences, 116(32), 15849–15854.

- Transformer and convolutional network model:

  Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B.,and Sutskever, I. (2020). *Deep Double Descent: Where Bigger Models and More Data Hurt.* Proceedings of the 8th International Conference on Learning Representations (ICLR)

Related work and historical development

- Random Fourier features, random forest and shallow networks:

  Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). *Reconciling modern machine-learning practice and the classical bias–variance trade-off.* Proceedings of the National Academy of Sciences, 116(32), 15849–15854.

- Transformer and convolutional network model:

  Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B.,and Sutskever, I. (2020). *Deep Double Descent: Where Bigger Models and More Data Hurt.* Proceedings of the 8th International Conference on Learning Representations (ICLR)

- Linear regression (with theoretical guarantees):

  Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R.J.(2019). *Surprises in High-Dimensional Ridgeless LeastSquares Interpolation.* arXiv:1903.08560.

  Bartlett, P.L., Long, P.M., Lugosi, G., and Tsigler, A.(2020). *Benign overfitting in linear regression.* Proceedings of the National Academy of Sciences

Related work and historical development

- Random Fourier features, random forest and shallow networks:

  Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). *Reconciling modern machine-learning practice and the classical bias–variance trade-off.* Proceedings of the National Academy of Sciences, 116(32), 15849–15854.

- Transformer and convolutional network model:

  Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B.,and Sutskever, I. (2020). *Deep Double Descent: Where Bigger Models and More Data Hurt.* Proceedings of the 8th International Conference on Learning Representations (ICLR)

- Linear regression (with theoretical guarantees):

  Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R.J.(2019). *Surprises in High-Dimensional Ridgeless LeastSquares Interpolation.* arXiv:1903.08560.

  Bartlett, P.L., Long, P.M., Lugosi, G., and Tsigler, A.(2020). *Benign overfitting in linear regression.* Proceedings of the National Academy of Sciences

- Random feature (with theoretical guarantees):

  Mei, S. and Montanari, A. (2019). *The generalization error of random features regression: Precise asymptotics and double descent curve.* arXiv:1908.05355.

Our contribution

Experimentally show the phenomena in the system identification setting: input-output data from a dynamical system.

Motivation example

$$y_t = \left(0.8 - 0.5e^{-{y_{t-1}}^2}\right) y_{t-1} - \left(0.3 + 0.9e^{-{y_{t-1}}^2}\right) y_{t-2}$$
$$+ u_{t-1} + 0.2u_{t-2} + 0.1u_{t-1}u_{t-2} + v_t,$$
$$v_t \sim \mathcal{N}(0, \sigma_v^2)$$

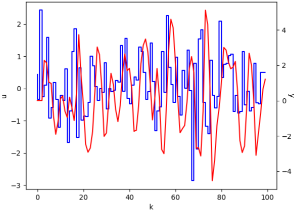


Figure: **System with process noise.** Input in blue and output in red.

Chen, S., Billings, S.A., and Grant, P.M. (1990). *Non-Linear System Identification Using Neural Networks.* International Journal of Control, 51(6), 1191–1214.

Model

Linear-in-the-parameters: Predicted output

$$\hat{y}_t = \theta^{\mathsf{T}} z_t.$$

- $\hat{y}_t \rightsquigarrow$ predicted output

Model

Linear-in-the-parameters: Predicted output

$$\hat{y}_t = \theta^\mathsf{T} z_t.$$

- $\hat{y}_t \rightsquigarrow$ predicted output
- $\theta \rightsquigarrow$ parameters being estimated

Model

Linear-in-the-parameters: Predicted output

$$\hat{y}_t = \theta^\mathsf{T} z_t.$$

- $\hat{y}_t \rightsquigarrow$ predicted output
- $\theta \rightsquigarrow$ parameters being estimated

Nonlinear feature map :

$$z_t = \left(\begin{bmatrix} u_{t-1} \\ u_{t-2} \\ y_{t-1} \\ y_{t-2} \end{bmatrix} \right)$$

Model

Linear-in-the-parameters: Predicted output

$$\hat{y}_t = \theta^\mathsf{T} z_t.$$

- $\hat{y}_t \rightsquigarrow$ predicted output
- $\theta \rightsquigarrow$ parameters being estimated

Nonlinear feature map :

$$z_t = \left(W \begin{bmatrix} u_{t-1} \\ u_{t-2} \\ y_{t-1} \\ y_{t-2} \end{bmatrix} \right)$$

- $W \rightsquigarrow$ Matrix with dimension $m \times 4$

Model

Linear-in-the-parameters: Predicted output

$$\hat{y}_t = \theta^{\mathsf{T}} z_t.$$

- $\hat{y}_t \rightsquigarrow$ predicted output
- $\theta \rightsquigarrow$ parameters being estimated

Nonlinear feature map :

$$z_t = \sigma \left(W \begin{bmatrix} u_{t-1} \\ u_{t-2} \\ y_{t-1} \\ y_{t-2} \end{bmatrix} \right)$$

- $W \rightsquigarrow$ Matrix with dimension $m \times 4$
- $\sigma \rightsquigarrow$ activation function (element-wise)

Model

Random matrix: (set in advance)

$$W = \underbrace{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,3} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m,1} & w_{m,2} & w_{m,3} & w_{m,3} \end{bmatrix}}_{} \left.\vphantom{\begin{bmatrix} w \\ w \\ w \\ \vdots \\ w \end{bmatrix}}\right\} m$$

Model

Random matrix: (set in advance)

$$W = \left.\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,3} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m,1} & w_{m,2} & w_{m,3} & w_{m,3} \end{bmatrix}\right\} m$$

where each entry is i.i.d.:

$$w_{i,j} \sim \mathcal{N}(0, \gamma)$$

Model

Random matrix: (set in advance)

$$W = \underbrace{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,3} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m,1} & w_{m,2} & w_{m,3} & w_{m,3} \end{bmatrix}}_{}\left.\vphantom{\begin{bmatrix} 1\\1\\1\\1\\1 \end{bmatrix}}\right\} m$$

where each entry is i.i.d.:

$$w_{i,j} \sim \mathcal{N}(0, \gamma)$$

Rahimi, A. and Recht, B. (2008). *Random Features for Large-Scale Kernel Machines.* Advances in Neural Information Processing Systems 20, 1177–1184

Model estimation

Estimated parameter: using train dataset (u_t, y_t), $t = 1, \cdots, n$:

Model estimation

Estimated parameter: using train dataset (u_t, y_t), $t = 1, \cdots, n$:

- Underparametrized:

$$\min_{\theta} \sum_t (y_i - \theta^\mathsf{T} z_t)^2$$

Model estimation

Estimated parameter: using train dataset (u_t, y_t), $t = 1, \cdots, n$:

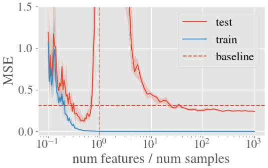
- Underparametrized:

$$\min_{\theta} \sum_t (y_i - \theta^\mathsf{T} z_t)^2$$

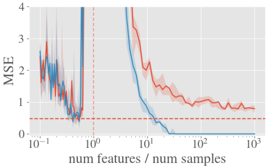
- Overparametrized:

$$\begin{aligned} &\min_{\theta} \|\theta\|_2^2 \\ \text{subject to } & y_t = \theta^\mathsf{T} z_t \\ & \quad \text{for every } t = 1, \cdots, n \end{aligned}$$

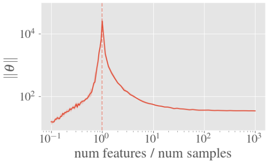
Results



(a) one-step-ahead



(b) free-run simulation



(c) parameter norm

Figure: **Double-descent performance curve.**

Ridge regression

$$\min_{\theta} \sum_{t} (y_i - \theta^\mathsf{T} z_t)^2 + \lambda \|\theta\|^2$$

Ridge regression

$$\min_{\theta} \sum_{t} (y_i - \theta^\mathsf{T} z_t)^2 + \lambda \|\theta\|^2$$

Tends to the minimum-norm solution when $\lambda \to 0+$

Ridge regression

$$\min_{\theta} \sum_t (y_i - \theta^\mathsf{T} z_t)^2 + \lambda \|\theta\|^2$$

Tends to the minimum-norm solution when $\lambda \to 0+$

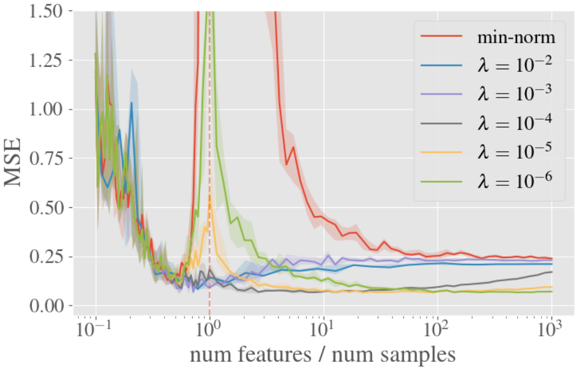


Figure: **Ridge regression with vanishing values of λ.**

Ensembles

- $n \rightsquigarrow \#$ datapoints.

Ensembles

- $n \rightsquigarrow \#$ datapoints.
- $m \rightsquigarrow \#$ features.

Ensembles

- $n \rightsquigarrow$ # datapoints.
- $m \rightsquigarrow$ # features.
- If $m > n$, pick $\mathcal{S} \in \{1, \cdots, m\}$ with n elements and solve the linear system:

$$y_t = \sum_{i \in \mathcal{S}} \theta_i z_{t,i} \text{ for all } i \in \mathcal{S}$$

Ensembles

- $n \rightsquigarrow \#$ datapoints.
- $m \rightsquigarrow \#$ features.
- If $m > n$, pick $\mathcal{S} \in \{1, \cdots, m\}$ with n elements and solve the linear system:

$$y_t = \sum_{i \in \mathcal{S}} \theta_i z_{t,i} \text{ for all } i \in \mathcal{S}$$

- Repeat B times for different sets.

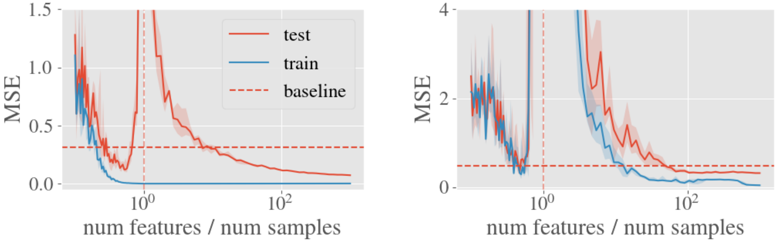
Ensembles

- $n \rightsquigarrow \#$ datapoints.
- $m \rightsquigarrow \#$ features.
- If $m > n$, pick $\mathcal{S} \in \{1, \cdots, m\}$ with n elements and solve the linear system:

$$y_t = \sum_{i \in \mathcal{S}} \theta_i z_{t,i} \text{ for all } i \in \mathcal{S}$$

- Repeat B times for different sets.
- Take the average

Ensembles



(a) one-step-ahead MSE (b) free-run simulation MSE

Figure: **Ensembles after the interpolation threshold**.

Coupled Electric Drives

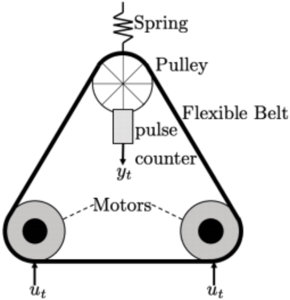
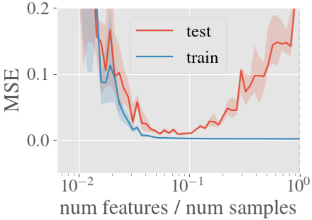


Figure: **Illustration of the CE8 coupled electric drives system**

Wigren, T. and Schoukens, M. (2017). *Coupled electric drives data set and reference models.* Technical Report. Uppsala Universitet, 2017

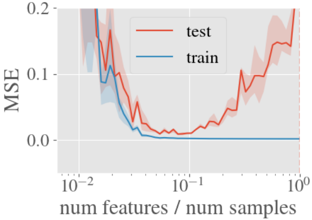
Double-descent in the CE8 benchmarks



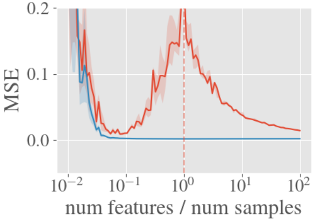
(a) U-shaped MSE

Figure: **Double-descent in CE8 Benchmark.** We show one-step-ahead prediction error in test and training data for a nonlinear ARX model in the CE8 benchmark.

Double-descent in the CE8 benchmarks



(a) U-shaped MSE



(b) Double-descent

Figure: **Double-descent in CE8 Benchmark.** We show one-step-ahead prediction error in test and training data for a nonlinear ARX model in the CE8 benchmark.

Final comments

- Not all nonlinear feature basis necessarily yields this behaviour!

Final comments

- Not all nonlinear feature basis necessarily yields this behaviour!
- For instance, I would not expect it, for instance, for a polynomial basis.

Final comments

- Not all nonlinear feature basis necessarily yields this behaviour!
- For instance, I would not expect it, for instance, for a polynomial basis.
- Additional experiments: Examples with Random Forest and Radial basis function (RBF) network.

Final comments

- Not all nonlinear feature basis necessarily yields this behaviour!
- For instance, I would not expect it, for instance, for a polynomial basis.
- Additional experiments: Examples with Random Forest and Radial basis function (RBF) network.
- Studying double descent for *nonlinear ARMAX*, *output error* and other types of models that can handle more general noise types.

Thank you!

To appear in the 19th IFAC Symposium in System Identification.

arXiv.org **Paper:** https://arxiv.org/abs/2012.06341

Thank you!

To appear in the 19th IFAC Symposium in System Identification.

arXiv.org **Paper:** https://arxiv.org/abs/2012.06341

Code: https://github.com/antonior92/narx-double-descent

Thank you!

To appear in the 19th IFAC Symposium in System Identification.

arXiv.org **Paper:** https://arxiv.org/abs/2012.06341

Code: https://github.com/antonior92/narx-double-descent

Contact: antonio.horta.ribeiro@it.uu.se