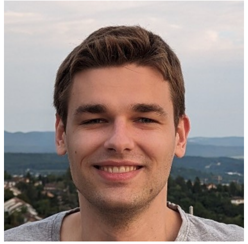
ECML PKDD 2024

Machine Learning for Irregular Time Series ML4ITS2024

Task-Synchronized Recurrent Neural Networks

Mantas Lukoševičius & Arnas Uselis

ktu 1922 kaunas university of technology

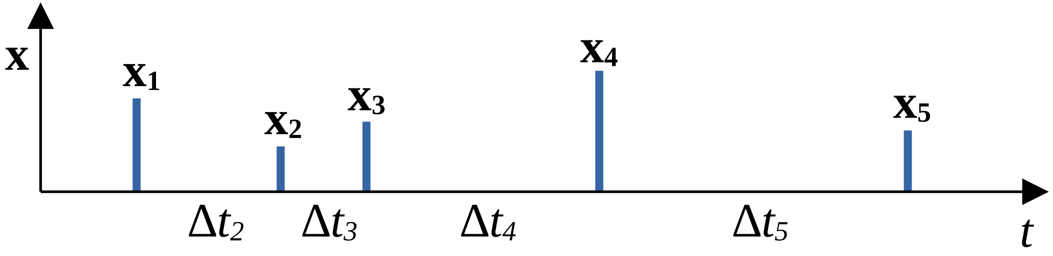


EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Vilnius, 2024.09.13

Irregular Time Series

- Data gathered at irregular intervals
  - Patient examination, test, expedition, etc.
- Data sources that get dated
  - Geological, archaeological sample, historic document, etc.
- Data points when certain events happen
  - Action by a user, accident, natural phenomenon, economic transaction, neural spike, etc.



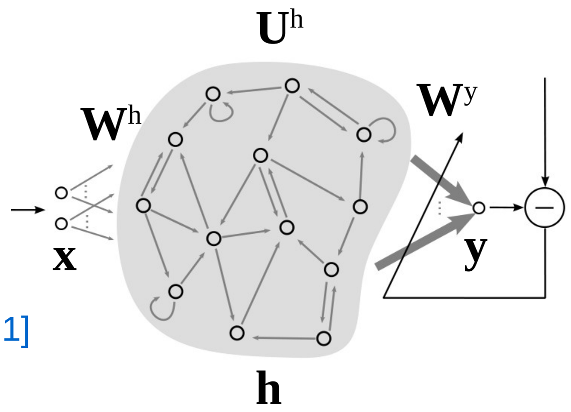
Recurrent Neural Networks (RNNs)

- Recurrences in connections
  - Dynamical systems (universal approximators, Turing complete)
  - Naturally suited for time series
  - Has internal state / memory, no fixed time window
- “Vanilla” (vs. GRU, LSTM)

$$\mathbf{h}_n = (1-\alpha)\mathbf{h}_{n-1} + \alpha\sigma\left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}\mathbf{h}_{n-1}\right)$$

- Reservoir computing: Echo State Network [Jaeger 2001], (vs. fully trained)
  - Investigate different aspects of RNNs in addition to learning
  - Typically only the linear readout is trained
    - Very fast one-shot training
    - Cross-validation can be done almost for “free” [Lukoševičius, Uselis 2021]

$$\mathbf{y}_n = \mathbf{W}^{\mathrm{y}}[\mathbf{x}_n; \mathbf{h}_n]$$

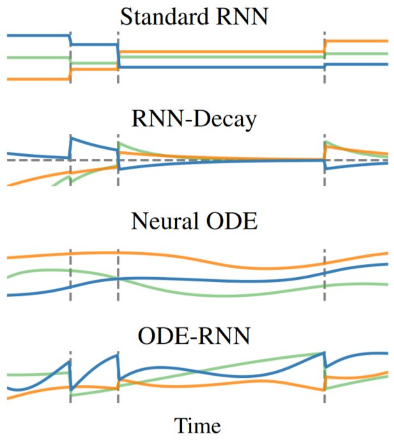


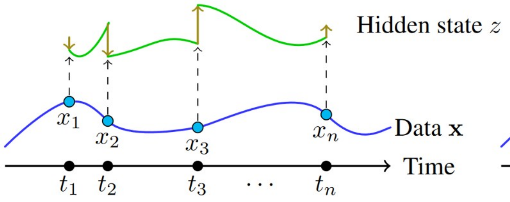
Dealing with Irregular Time with RNNs

- Ignore it’s irregular
- Resample/interpolate the data
- Feed Δt_n as additional (special?) input
- Have specialized data pre-treatment layers
- **“Resample” the RNN to match the time of the data**

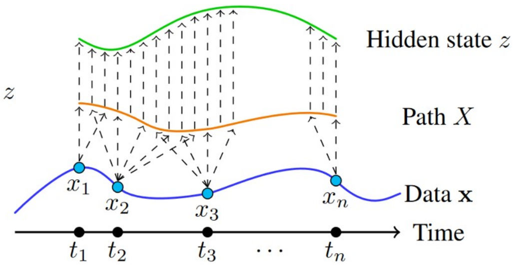
RNN Methods Employing Differential Equation (DE) Solvers

- ODE-RNN [Rubanova et al 2019],
  - Based on Neural ODEs [Chen et al 2018]
  - Simple RNN updates with input
  - ODE evolution in between
  - The two don't mix well
- Neural CDE (Controlled DE) [Kidger et al 2020]
  - Inputs continually affect the RNN state
  - They have to be interpolated
  - Defeats the purpose of having irregular time?
- Both are computationally expensive





ODE-RNN



Neural CDE

Task-Synchronized Echo State Networks

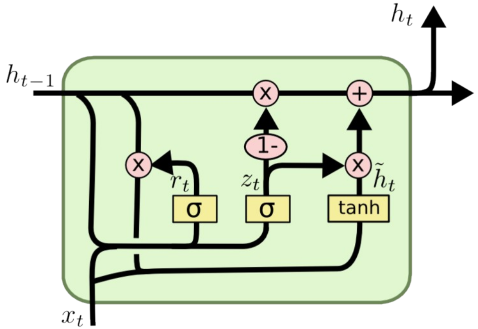
- Continuous-time RNN $$\dot{\mathbf{h}} = \alpha \cdot \left(-\mathbf{h} + \sigma(\mathbf{W}^{\mathrm{h}}\mathbf{x} + \mathbf{U}^{\mathrm{h}}\mathbf{h})\right)$$
- Euler discretization $$\dot{\mathbf{h}} \approx \frac{\mathbf{h}_n - \mathbf{h}_{n-1}}{\Delta t}$$
- Discrete-time RNN $$\mathbf{h}_n = (1 - \alpha \Delta t)\, \mathbf{h}_{n-1} + \alpha \Delta t \sigma \left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}\mathbf{h}_{n-1}\right)$$
- RNN (ESN) with leaky integration, e.g. [Lukoševičius 2012] $$\mathbf{h}_n = (1 - \alpha)\mathbf{h}_{n-1} + \alpha \sigma \left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}\mathbf{h}_{n-1}\right)$$
- **Task-Synchonized ESN** $$\mathbf{h}_n = (1 - \alpha \Delta t_n)\mathbf{h}_{n-1} + \alpha \Delta t_n \sigma \left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}\mathbf{h}_{n-1}\right)$$
- Readout $$\mathbf{y}_n = \mathbf{W}^{\mathrm{y}}[\mathbf{x}_n; \mathbf{h}_n]$$

Task-Synchronized Gated Recurrent Units

- GRUs [Cho et al 2014]
  - Update and reset (forget) gates
  - State update
  - (Update gate ~ like learned unit-wise leaking rate)

$$\mathbf{z}_n = \sigma_g(\mathbf{W}^{\mathrm{z}}\mathbf{x}_n + \mathbf{U}^{\mathrm{z}}\mathbf{h}_{n-1}),$$
$$\mathbf{r}_n = \sigma_g(\mathbf{W}^{\mathrm{r}}\mathbf{x}_n + \mathbf{U}^{\mathrm{r}}\mathbf{h}_{n-1}),$$

$$\mathbf{h}_n = (1-\mathbf{z}_n)\circ\mathbf{h}_{n-1} + \mathbf{z}_n\circ\sigma\left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}(\mathbf{r}_n\circ\mathbf{h}_{n-1})\right)$$



- Continuous-time version

$$\dot{\mathbf{h}} = \mathbf{z}\circ\left(-\mathbf{h} + \sigma(\mathbf{W}^{\mathrm{h}}\mathbf{x} + \mathbf{U}^{\mathrm{h}}(\mathbf{r}\circ\mathbf{h}))\right)$$

- **Task-Synchonized GRU**

$$\mathbf{h}_n = (1-\Delta t_n\mathbf{z}_n)\circ\mathbf{h}_{n-1} + (\Delta t_n\mathbf{z}_n)\circ\sigma\left(\mathbf{W}^{\mathrm{h}}\mathbf{x}_n + \mathbf{U}^{\mathrm{h}}(\mathbf{r}_n\circ\mathbf{h}_{n-1})\right)$$

- Readout $\mathbf{y}_t = \mathbf{W}^{\mathrm{y}}[1;\mathbf{h}_t]$

Nonlinear Time Scaling

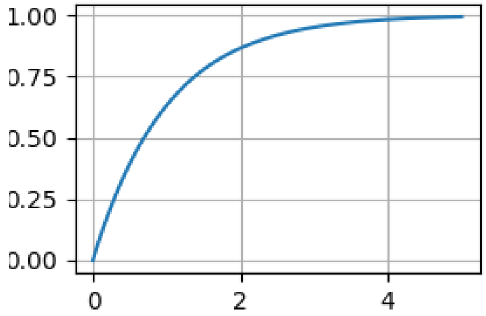
$$\mathbf{h}_n = (1 - \alpha \Delta t_n)\mathbf{h}_{n-1} + \alpha \Delta t_n \sigma \left(\mathbf{W}^{\mathrm{h}} \mathbf{x}_n + \mathbf{U}^{\mathrm{h}} \mathbf{h}_{n-1} \right)$$

- $(1 - \alpha\Delta t_n)$ should not be negative:

  $\Delta t_n \leq 1/\alpha$

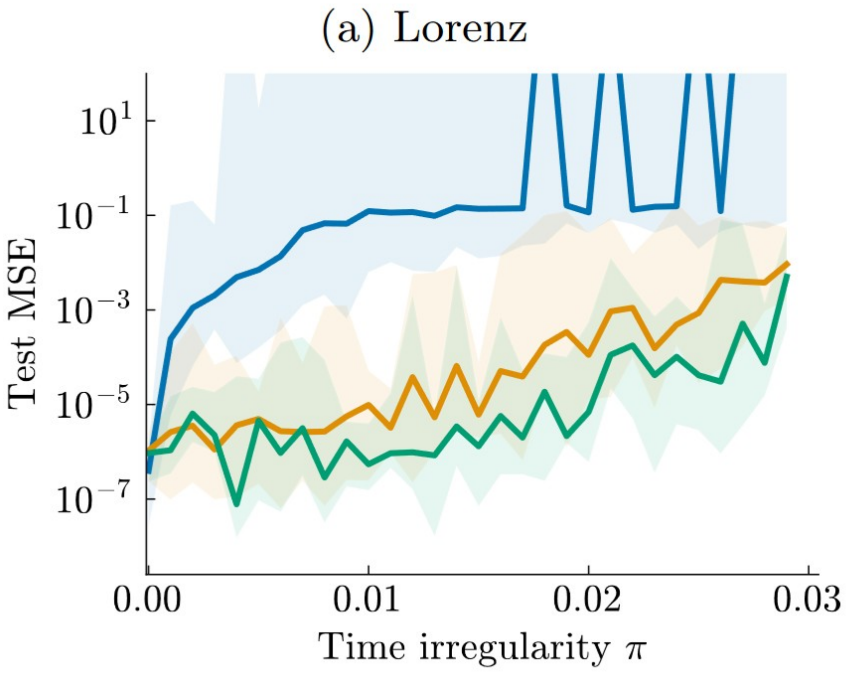
- We can scale Δt_n, but outliers...
- One solution: introduce a version where we replace Δt_n with: ("exp" versions)

$$f(\Delta t_n) = 1 - e^{-\Delta t_n}$$



Synthetic Chaotic Attractor Datasets

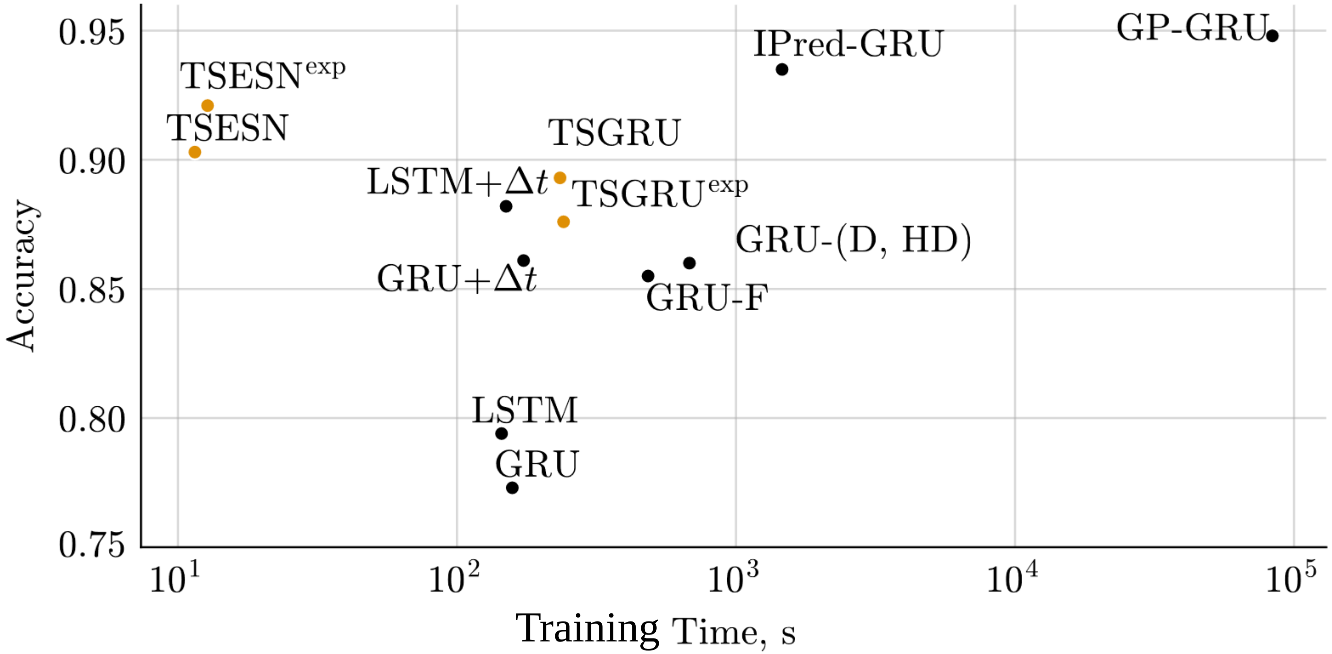
With artificially introduced and controlled levels of (high) time irregularity



UWave Gesture Dataset

Gestures generated from accelerometers

- 8 class classification
- Univariate
- 3582 train and 896 test instances
  - 30% of train is used for validation
  - Each instance is 945 time samples
- 10% of samples is randomly taken to have sparse irregular time series



[Jiayang Liu et al 2009]
https://timeseriesclassification.com
Following [Shukla, Marlin 2019]

Speleothem Dataset

Summer monsoon rainfall
(readings of oxygen isotopes)
over the last two millennia
in a speleothem in a cave in India

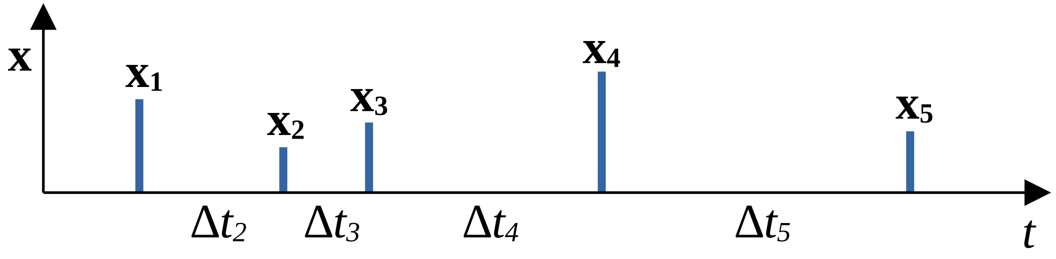
- Prediction
- Univariate
- 1800 samples
  - 1700 for training,
  - 50 for validation,
  - the last 50 for testing
- Very non-stationary (2000 years...)

Model	Valid. RMSE	Test RMSE	Test MAPE	$f(\Delta t_n)$	Valid. type
ESN	0.117 ± 0.009	0.425 ± 0.064	4.311 ± 0.702	linear	standard
ESN+Δt	0.161 ± 0.036	0.348 ± 0.044	3.377 ± 0.456	linear	standard
Interp. ESN	0.139 ± 0.022	0.346 ± 0.093	3.397 ± 0.998	linear	standard
TSESN	0.129 ± 0.008	0.408 ± 0.026	4.125 ± 0.296	exp	standard
TSESN	0.120 ± 0.011	0.490 ± 0.021	4.952 ± 0.237	linear	standard
GRU	0.169 ± 0.010	0.351 ± 0.028	3.401 ± 0.304	linear	standard
LSTM	0.160 ± 0.001	0.405 ± 0.009	3.999 ± 0.103	linear	standard
GRU+Δt	0.164 ± 0.003	0.366 ± 0.017	3.559 ± 0.192	linear	standard
LSTM+Δt	0.162 ± 0.001	0.403 ± 0.023	3.981 ± 0.267	linear	standard
Interp. GRU	0.171 ± 0.010	$\underline{0.334 \pm 0.022}$	$\underline{3.206 \pm 0.236}$	linear	standard
Interp. LSTM	0.157 ± 0.002	0.383 ± 0.010	3.738 ± 0.118	linear	standard
TSGRU	0.187 ± 0.021	0.349 ± 0.053	3.389 ± 0.577	linear	standard
ESN	0.193 ± 0.005	0.185 ± 0.006	1.709 ± 0.057	exp	CV
ESN+Δt	0.201 ± 0.005	0.244 ± 0.001	2.340 ± 0.010	exp	CV
Interp. ESN	0.183 ± 0.004	0.304 ± 0.005	2.917 ± 0.051	exp	CV
TSESN	0.199 ± 0.008	$\underline{0.159 \pm 0.002}$	$\underline{1.478 \pm 0.017}$	exp	CV
TSESN	0.182 ± 0.004	0.346 ± 0.014	3.333 ± 0.150	linear	CV

[Sinha et al 2015]

Summary

- Advantages
  - Very fast and natural: “built-in”
  - No additional learning
  - No inventing of data
  - Effective
  - Should be default?



- Limitations
  - No asynchronous data
  - Huge time gaps are problematic
    - Exp Δt_n
    - Impute?

Future work

- More models
  - Fully-trained TSRNN
  - TSLSTM?
  - In combination with other techniques
- More applications
  - Multivariate
- More comparisons to other models

Questions?

https://arxiv.org/abs/2204.05192

https://github.com/oshapio/task-synchronized-RNNs

https://mantas.info/