Learning Recurrent Models with Temporally Local Rules

Azwar Abdulsalam (abdulsal@purdue.edu)
Joseph G. Makin (jgmakin@purdue.edu)

PURDUE UNIVERSITY

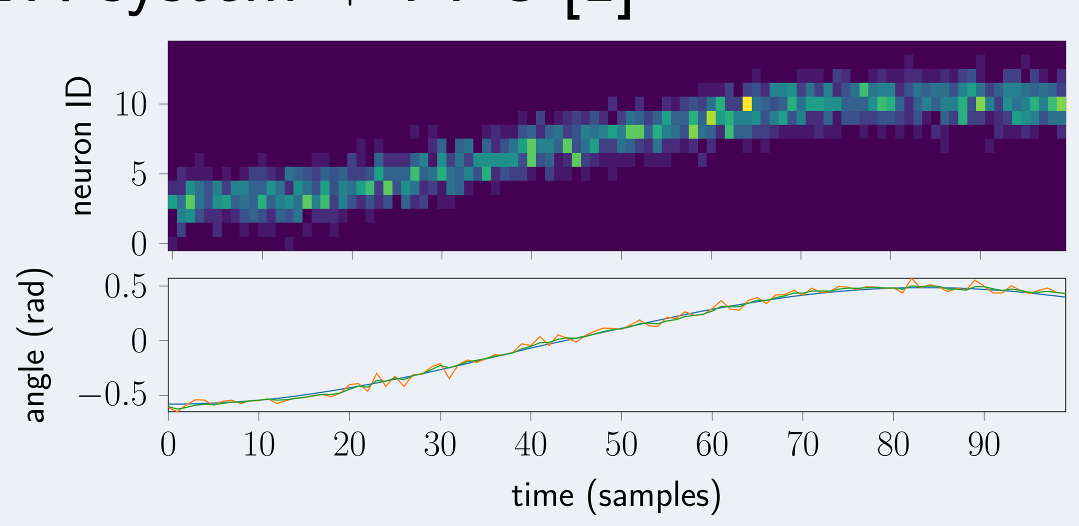
Elmore Family School of Electrical and Computer Engineering, Purdue University

Abstract

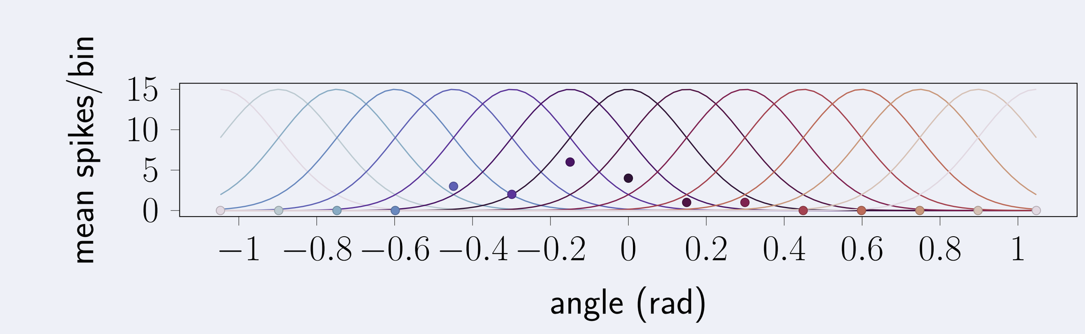
- Fitting **generative models** to sequential data typically requires **backprop-through-time**
- BPTT is biologically implausible and computationally expensive
- We investigate an alternative: **require the generative model to learn the joint distribution over current and previous states, rather than merely the forward-transition probabilities** [2, 3]
- Two architectures: **rEFH**, **rVAE**
- On toy datasets the procedure has the same effect as including BPTT

Experiments

LTI system + PPC [1]



$$p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) = \mathcal{N}(\mathbf{A}\boldsymbol{z}_t,\ \boldsymbol{\Sigma}_{\hat{x}}) \qquad p(\boldsymbol{y}_t|\boldsymbol{z}_t) = \prod_{i=1}^{15} \mathsf{Pois}\big(y_t^i | g_t f_i(z_t^1)\big)$$



- A crude biophysical model of neurons reporting a stimulus.
- Second-order (oscillatory!) dynamics, but only position is "reported" by the neurons.
- Observation model is *nonlinear*, but a closed-form solution is still available (KF). This allows us to determine what order model was learned.
- What algorithms can learn **2nd**-order dynamics?

Bouncing balls [4]

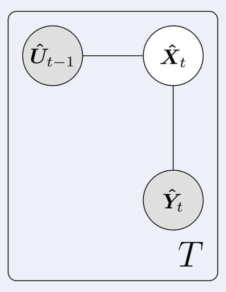
- Video sequences of 3 balls bouncing off each other and walls with complete energy conservation
- constant velocities (2nd-order, linear) + collisions (nonlinear)

Moving MNIST [4]

- Video sequences of moving MNIST digits bouncing off walls and passing through one another
- constant velocities (2nd-order, linear) + overlap (nonlinear) + occlusions (nonlinear)

Models

$$\hat{p}(\hat{\boldsymbol{u}}_{t-1}|\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \mathsf{Bern}\big(\sigma(\mathbf{W}_u^{\mathsf{T}}\hat{\boldsymbol{x}}_t + \boldsymbol{b}_u)\big)$$



$$\hat{p}(\hat{\boldsymbol{x}}_t|\hat{\boldsymbol{u}}_{t-1},\hat{\boldsymbol{y}}_t;\boldsymbol{\theta}) = \mathsf{Bern}(\sigma(\mathbf{W}_u\hat{\boldsymbol{u}}_{t-1} + \mathbf{W}_y\hat{\boldsymbol{y}}_t + \boldsymbol{b}_x))$$

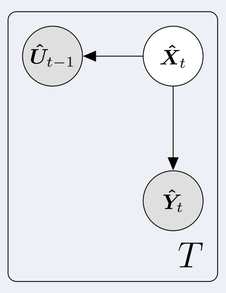
$$\hat{p}(\hat{\boldsymbol{y}}_t|\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \begin{cases}\mathsf{Bern}\big(\sigma(\mathbf{W}_y^{\mathsf{T}}\hat{\boldsymbol{x}}_t + \boldsymbol{b}_y)\big)\\ \mathsf{Pois}\big(\exp(\mathbf{W}_y^{\mathsf{T}}\hat{\boldsymbol{x}}_t + \boldsymbol{b}_y)\big)\end{cases}$$

$$\frac{\mathrm{d}\mathcal{L}_{\mathsf{rEFH}}}{\mathrm{d}\boldsymbol{\theta}} = \sum_{t=1}^{T}\frac{\mathrm{d}\mathsf{H}_{p\hat{p}}[\boldsymbol{U}_{t-1},\boldsymbol{Y}_t;\boldsymbol{\theta}]}{\mathrm{d}\boldsymbol{\theta}} \approx \sum_{t=1}^{T}\mathbb{E}_{\hat{\boldsymbol{X}}_t,\boldsymbol{Y}_t,\boldsymbol{U}_{t-1}}\left[\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}\right] - \mathbb{E}_{\hat{\boldsymbol{X}}_t,\hat{\boldsymbol{Y}}_t,\hat{\boldsymbol{U}}_{t-1}}\left[\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}\right]$$

rEFH

$$\hat{p}(\hat{\boldsymbol{u}}_{t-1}|\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{x}}(\hat{\boldsymbol{x}}_t,\boldsymbol{\theta}),\ \sigma_{\hat{x}}^2\mathbf{I})$$

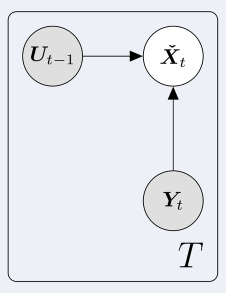
$$\check{p}(\check{\boldsymbol{x}}_t|\boldsymbol{u}_{t-1},\boldsymbol{y}_t;\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\nu}_{\check{x}}(\boldsymbol{u}_{t-1},\boldsymbol{y}_t),\ \boldsymbol{\Upsilon}_{\check{x}}(\boldsymbol{u}_{t-1},\boldsymbol{y}_t))$$



$$\hat{p}(\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0},\ \mathbf{I})$$

$$\hat{p}(\hat{\boldsymbol{y}}_t|\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \mathcal{N}\Big(\boldsymbol{\mu}_{\hat{y}}(\hat{\boldsymbol{x}}_t,\boldsymbol{\theta}),\ \sigma_{\hat{y}}^2\mathbf{I})\Big)$$

generative model



recognition model

$$\frac{\mathrm{d}\mathcal{L}_{\mathsf{rVAE}}}{\mathrm{d}\boldsymbol{\theta}^{\mathsf{T}}} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}^{\mathsf{T}}}\sum_{t=1}^{T}\mathbb{E}_{\check{\boldsymbol{X}}_t,\boldsymbol{U}_{t-1},\boldsymbol{Y}_t}\Big[\log\check{p}\big(\check{\boldsymbol{X}}_t|\boldsymbol{U}_{t-1},\boldsymbol{Y}_t;\boldsymbol{\phi}\big) - \log\hat{p}\big(\check{\boldsymbol{X}}_t,\boldsymbol{U}_{t-1},\boldsymbol{Y}_t;\boldsymbol{\theta}\big)\Big]$$
$$\approx \sum_{t=1}^{T}\mathbb{E}_{\check{\boldsymbol{X}}_t,\boldsymbol{U}_{t-1},\boldsymbol{Y}_t}\left[\frac{\mathrm{d}\log\check{p}\big(\check{\boldsymbol{X}}_t|\boldsymbol{U}_{t-1},\boldsymbol{Y}_t;\boldsymbol{\phi}\big)}{\mathrm{d}\boldsymbol{\theta}^{\mathsf{T}}} - \frac{\mathrm{d}\log\hat{p}\big(\check{\boldsymbol{X}}_t,\boldsymbol{U}_{t-1},\boldsymbol{Y}_t;\boldsymbol{\theta}\big)}{\mathrm{d}\boldsymbol{\theta}^{\mathsf{T}}}\right]$$

rVAE

Letting the derivative pass through the expectations

- ignores the dependence of $\boldsymbol{U}_{t-1}$ on the parameters $\boldsymbol{\theta}$.
- amounts to discarding BPTT

Rationale

Model requirements

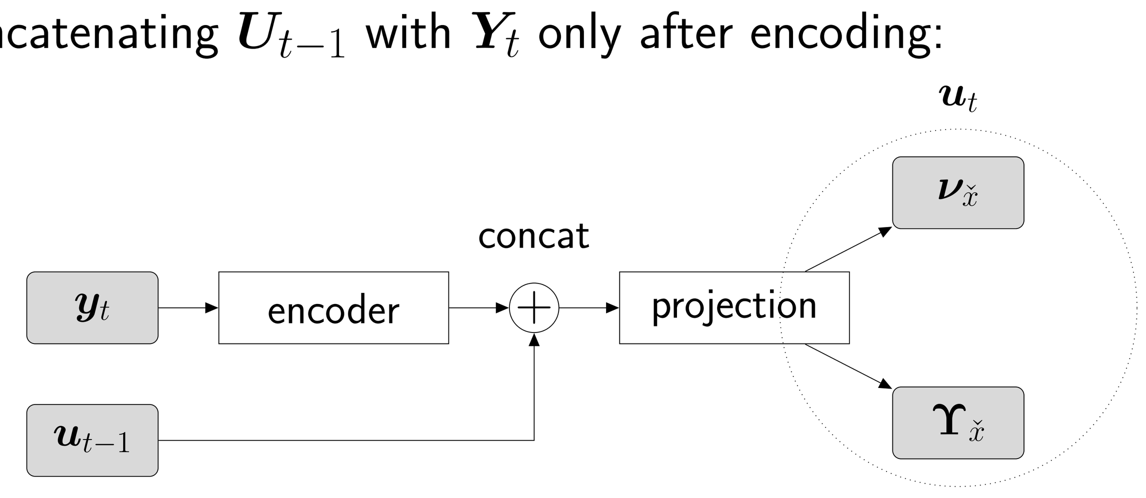
1) Generative model with latent variables, $\hat{\boldsymbol{X}}$
2) Latent variables inferable: $\hat{p}(\hat{\boldsymbol{x}}_t|\boldsymbol{u}_{t-1},\boldsymbol{y}_t;\boldsymbol{\theta})$ or $\check{p}(\check{\boldsymbol{x}}_t|\boldsymbol{u}_{t-1},\boldsymbol{y}_t;\boldsymbol{\phi})$
3) compression: $\dim\Big(\hat{\boldsymbol{X}}\Big) < \dim(\boldsymbol{Y})$

Candidate generative models

× GAN (cannot infer latent variables)
× Energy-Based Models (no latent variables)
× Diffusion $(\dim\Big(\hat{\boldsymbol{X}}\Big) = \dim(\boldsymbol{Y}))$
× Flow $(\dim\Big(\hat{\boldsymbol{X}}\Big) = \dim(\boldsymbol{Y}))$
✓ Variational Auto-Encoder
✓ Exponential-Family Harmonium

Architecture/Training

- The rVAE generative and recognition models make different independence statements. We try to minimize the discrepancy by concatenating $\boldsymbol{U}_{t-1}$ with $\boldsymbol{Y}_t$ only after encoding:



- All models trained with stochastic gradient descent and AdaM optimization. The learning rate was configured to 1e-4, with β_1 and β_2 values set to 0.9 and 0.999, respectively.

Results

Quantitative Results

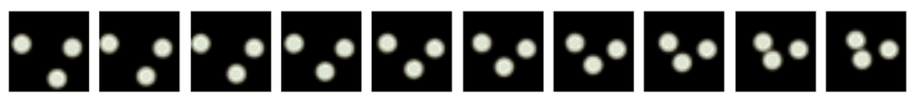
Model	MSE
order 0	12×10^{-4}
TVAE	9.5×10^{-4}
TRBM*	6.0×10^{-4}
KF-1	5.8×10^{-4}
rVAE	5.3×10^{-4}
rEFH	3.3×10^{-4}
RTRBM*	3.1×10^{-4}
KF-2	2.2×10^{-4}

Mean squared errors (MSE) for recovery of position information on the PPC experiment.

Model	MSE
order 0	0.0120
TRBM	0.0124
rEFH	0.0067
RTRBM	0.0059

Mean squared errors (MSE) for bouncing-ball one-step predictions (w/clamped Gibbs sampling)

Qualitative Results



Bouncing-ball frame sequence generated by rVAE.

MovingMNIST frame sequence generated by rVAE.

Conclusions

- Learning joint over current and previous states seems to obviate BPTT
- rVAE learns 2nd-order dynamics from position observations, but worse than rEFH
- The procedure, though intuitive, requires a mathematical basis and scaling to handle more challenging datasets.

References

[1] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget. Bayesian Inference with Probabilistic Population Codes. *Nature Neuroscience*, 9(11):1423–1438, 2006.

[2] J. G. Makin, B. K. Dichter, and P. N. Sabes. Learning to Estimate Dynamical State with Probabilistic Population Codes. *PLoS Computational Biology*, 11(11):1–28, 2015.

[3] J. G. Makin, B. K. Dichter, and P. N. Sabes. Recurrent Exponential-Family Harmoniums without Backprop-Through-Time. https://arxiv.org/abs/1605.05799, 2016.

[4] I. Sutskever. *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, 2013.