Learning Recurrent Models with Temporally Local Rules

Azwar Abdulsalam [1] **Joseph G. Makin** [1]

Abstract

Fitting generative models to sequential data typically involves two recursive computations through time, one forward and one backward. The latter could be a computation of the loss gradient (as in backpropagation through time), or an inference algorithm (as in the RTS/Kalman smoother). The backward pass in particular is computationally expensive (since it is inherently serial and cannot exploit GPUs), and difficult to map onto biological processes. Work-arounds have been proposed; here we explore a very different one: requiring the generative model to learn the joint distribution over current and previous states, rather than merely the transition probabilities. We show on toy datasets that different architectures employing this principle can learn aspects of the data typically requiring the backward pass.

1. Introduction

We consider the general context of trying to fit a model to sequential data, i.e. minimizing the KL divergence between the data distribution $p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_T)$ and the generative-model distribution $\hat{p}(\hat{\boldsymbol{y}}_1, \dots, \hat{\boldsymbol{y}}_T; \boldsymbol{\theta})$. The classic approach is to assume that the temporal correlations in the observed data $\boldsymbol{Y}_t$ can be explained by a latent state, $\hat{\boldsymbol{X}}_t$, that evolves according to a Markov chain. The assumption is not by itself very restrictive, and indeed this framework includes hidden Markov models (HMMs), linear-Gaussian dynamical systems (LGDSs), as well as modern variations like sequential VAEs.

Direct minimization of the marginal KL divergence, or equivalently cross entropy,

$$\mathrm{H}_{p\hat{p}}[\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T] := \langle -\log \hat{p}(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T; \boldsymbol{\theta}) \rangle_{\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T}$$

generates a backward recursion: the gradient of this loss is coupled across time because the observations are. When the dependencies are structured by a neural network, this recursion is called backpropagation through time (BPTT) (Werbos, 1988). This is the approach taken, e.g., in various extensions of the restricted Boltzmann machine (RBM) to temporal data (Sutskever, 2013; Boulanger-Lewandowski et al., 2012; Mittelman et al., 2014). For data arriving in real time, it is not clear how BPTT could be implemented biologically, since it would require a memory trace of the observed data. And the algorithm is perforce serial, and therefore computationally expensive.

An alternative is provided by the expectation-maximization (EM) (Dempster et al., 1977) algorithm, but it only avoids BPTT in certain special cases. It is well known that the marginal cross-entropy is upper-bounded by the *joint* cross entropy,

$$\mathrm{H}_{p\hat{p}\check{p}}\big[\check{\boldsymbol{X}}_1, \dots, \check{\boldsymbol{X}}_T, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T\big] := \big\langle -\log \hat{p}\big(\check{\boldsymbol{X}}_1, \dots, \check{\boldsymbol{X}}_T, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T; \boldsymbol{\theta}\big) \big\rangle_{\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_T, \check{\boldsymbol{X}}_1, \dots, \check{\boldsymbol{X}}_T},$$

where the average is taken under a "recognition model" (Dayan et al., 1995), $\check{p}(\check{\boldsymbol{x}}_1, \dots, \check{\boldsymbol{x}}_T | \boldsymbol{y}_1, \dots, \boldsymbol{y}_T; \boldsymbol{\phi})$, as well as the data distribution. The generative joint distribution ($\hat{p}$) factors across time, due to the assumption that the dependencies in the observations ($\boldsymbol{Y}_t$) can be explained by the latent state ($\hat{\boldsymbol{X}}_t$). This eliminates the need for BPTT in learning the *generative model*. In certain simple cases, like the HMM and LGDS, the generative model can be inverted in closed form with Bayes' rule, and used as the recognition model, in which case BPTT is not necessary at all. However, the expectations under the recognition distribution ($\check{p}$) still need to be computed with a forward and backward pass through the entire data sequence—famously, the forward-backward algorithm (HMMs) and Kalman filtering/RTS smoothing (LGDSs). Similar considerations about computational efficiency and biological plausibility apply. On the other hand, since the joint cross entropy is anyway a bound on the quantity we care about, the marginal cross entropy, the recognition distribution over latent states could simply be assumed (erroneously) to depend only on past observations—at the price of a looser bound. This would amount to (e.g.) using the Kalman filter in place of the RTS smoother.

More critically, any sequential model much more expressive than the HMM or LGDS cannot be inverted with Bayes'

[1]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, U.S.A.. Correspondence to: J.G. Makin <jgmakin@purdue.edu>.

In *ICML Workshop on Localized Learning (LLW)*, Honolulu, Hawaii, USA. 2023.

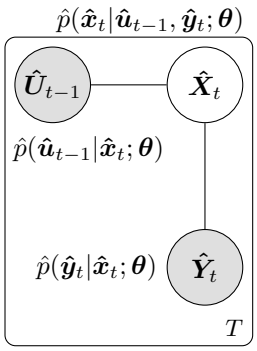

Figure 1. **The rEFH.** All distributions factor completely.

rule. Consequently, if the model is nevertheless to be trained with an EM-like framework—like sequential VAEs (Saxena et al., 2021; Hafner et al., 2019)—the standard approach is to model the recognition distribution independently, with its own set of parameters, and learn them along with the generative model. This again requires BPTT, this time through the recognition model ("encoder" in the language of VAEs).

Here we propose an alternative that still keeps learning temporally local. In place of the standard generative model, a product over transition $\hat{p}(\hat{\boldsymbol{x}}_t|\hat{\boldsymbol{x}}_{t-1}; \boldsymbol{\theta})$ and emission probabilities $\hat{p}(\hat{\boldsymbol{y}}_t|\hat{\boldsymbol{x}}_t; \boldsymbol{\theta})$, we propose to model at each time step the joint distribution of the current and previous states, along with the current observations. More precisely, we model the distribution over *the sufficient statistics for* the previous state, since these contain all the information about the preceding observation sequence (we make this precise below). Intuitively, we require the model *both* to yield good inferences about the hidden state, given the observations and previous state; *and* to be a good generative model *in reverse*. We find that, together, these demands seem to enforce information flows in both directions, obviating the need for BPTT or a backward pass of inference.

To evaluate our approach, we concentrate on a toy dataset for which it is possible to reason about the dynamics, as well as two other simple datasets from the literature.

2. Models

We focus on two architectures: a recurrent RBM (Smolensky, 1986), called the recurrent exponential-family harmonium (rEFH) (Makin et al., 2015); and a recurrent VAE (Rezende et al., 2014; Kingma & Ba, 2014), rVAE. For brevity we derive only the rVAE here, since the rEFH (Fig. 1) has been derived elsewhere (Makin et al., 2016).

Gaussian emissions. Consider the graphical model on the left in Fig. 2, parameterized by the distributions

$$\begin{aligned} \hat{p}(\hat{\boldsymbol{x}}_t; \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{0},\ \mathbf{I}) \\ \hat{p}(\hat{\boldsymbol{u}}_{t-1}|\hat{\boldsymbol{x}}_t; \boldsymbol{\theta}) &= \mathcal{N}\big(\boldsymbol{\mu}_{\hat{u}}(\hat{\boldsymbol{x}}_t, \boldsymbol{\theta}),\ \sigma^2_{\hat{u}}\mathbf{I}\big) \\ \hat{p}(\hat{\boldsymbol{y}}_t|\hat{\boldsymbol{x}}_t; \boldsymbol{\theta}) &= \mathcal{N}\big(\boldsymbol{\mu}_{\hat{y}}(\hat{\boldsymbol{x}}_t, \boldsymbol{\theta}),\ \sigma^2_{\hat{y}}\mathbf{I}\big)\big). \end{aligned} \tag{1}$$

Our goal is to learn the parameters of this generative model for observed data $\boldsymbol{U}_{t-1}, \boldsymbol{Y}_t \sim p(\boldsymbol{u}_{t-1}, \boldsymbol{y}_t)$. Classically, this would be carried out with EM, but for sufficiently complicated neural networks $\boldsymbol{\mu}_{\hat{y}}(\hat{\boldsymbol{x}}_t, \boldsymbol{\theta})$ and $\boldsymbol{\mu}_{\hat{u}}(\hat{\boldsymbol{x}}_t, \boldsymbol{\theta})$, the final equation precludes computation of the posterior with Bayes rule. Therefore to carry out an EM-like algorithm we make use of a *recognition model* (Neal & Hinton, 1998) (see Fig. 2, right),

$$\check{p}(\check{\boldsymbol{x}}_t|\boldsymbol{u}_{t-1}, \boldsymbol{y}_t; \boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\nu}_{\check{x}}(\check{\boldsymbol{u}}_{t-1}, \boldsymbol{y}_t),\ \boldsymbol{\Upsilon}_{\check{x}}(\check{\boldsymbol{u}}_{t-1}, \boldsymbol{y}_t)), \tag{2}$$

that is likewise parameterized with neural networks (but functions of the observations rather than the latent variables), but that might not match the true posterior distribution under the generative model. We further require the covariance to be diagonal:

$$\boldsymbol{\Upsilon}_{\check{x}}(\check{\boldsymbol{u}}_{t-1}, \boldsymbol{y}_t) := \mathrm{diag}\big(\boldsymbol{\upsilon}^2_{\check{x}}(\check{\boldsymbol{u}}_{t-1}, \boldsymbol{y}_t)\big).$$

From the generative and recognition models we can construct the free energy,

$$\begin{aligned} \mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\check{\boldsymbol{X}}_t, \boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\big[\log \check{p}\big(\check{\boldsymbol{X}}_t|\boldsymbol{U}_{t-1}, \boldsymbol{Y}_t; \boldsymbol{\phi}\big)\big] \\ - \mathbb{E}_{\check{\boldsymbol{X}}_t, \boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\big[\log \hat{p}\big(\check{\boldsymbol{X}}_t, \boldsymbol{U}_{t-1}, \boldsymbol{Y}_t; \boldsymbol{\theta}\big)\big], \end{aligned} \tag{3}$$

which is an upper bound on the marginal cross entropy $\mathrm{H}_{\hat{p}\check{p}}\big[\boldsymbol{U}_{t-1}, \boldsymbol{Y}_t; \boldsymbol{\theta}\big]$ (Neal & Hinton, 1998), and has a tractable gradient. (Note that the expectation is under the recognition distribution and the observed data.) When Eq. 3 is minimized by gradient descent in the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using the "reparameterization trick" (Kingma & Ba, 2014) (that is, the pathwise gradient estimator (Mohamed et al., 2020)), the resulting network is called a variational autoencoder.

Eq. 3 can be rearranged to exploit the independence statements asserted by Fig. 2. Substituting in the generative and recognition models from Eqs. 1 and 2, we find that

$$\begin{aligned} \mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\mathbb{E}_{\boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\left[\sum_k^K \log \upsilon^{(k)}_{\check{x}}\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\big[\boldsymbol{\nu}^{\mathrm{T}}_{\check{x}}\boldsymbol{\nu}_{\check{x}} + \boldsymbol{\upsilon}^{\mathrm{T}}_{\check{x}}\boldsymbol{\upsilon}_{\check{x}}\big] \\ + \frac{1}{2}\mathbb{E}_{\check{\boldsymbol{X}}_t, \boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\left[K \log|\sigma^2_{\hat{u}}| + \frac{\|\boldsymbol{U}_{t-1} - \boldsymbol{\mu}_{\hat{u}}\|^2}{\sigma^2_{\hat{u}}}\right] \\ + \frac{1}{2}\mathbb{E}_{\check{\boldsymbol{X}}_t, \boldsymbol{U}_{t-1}, \boldsymbol{Y}_t}\left[M \log|\sigma^2_{\hat{y}}| + \frac{\|\boldsymbol{Y}_t - \boldsymbol{\mu}_{\hat{y}}\|^2}{\sigma^2_{\hat{y}}}\right] + c, \end{aligned}$$

with c a constant term. Since the emission cumulants, $\boldsymbol{\mu}_{\hat{y}}$ and $\boldsymbol{\Sigma}_{\hat{y}}$, are complex (neural-network) functions of the

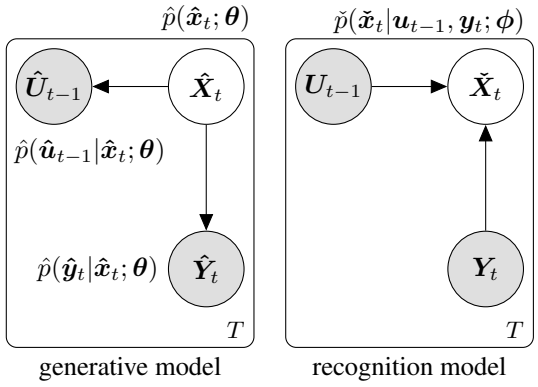

Figure 2. **The rVAE.** Note that the generative and recognition models assert incompatible independencies.

latent variables, exact expectations under the recognition model $\check{p}(\check{\boldsymbol{x}}_t|\boldsymbol{u}_{t-1},\boldsymbol{y}_t;\boldsymbol{\phi})$ are intractable, and must be replaced with sample averages. Likewise, since we have access to the data distribution only via samples, the expectations under $\boldsymbol{U}_{t-1}$ and $\boldsymbol{Y}_t$ must also be replaced with sample averages.

Poisson emissions. We also consider the case where the observations $\boldsymbol{Y}_t$ are Poisson distributed and (still) conditionally independent:

$$\hat{p}(\hat{\boldsymbol{y}}_t|\hat{\boldsymbol{x}}_t;\boldsymbol{\theta}) = \prod_m^M \text{Pois}(\lambda_m(\hat{\boldsymbol{x}}_t)),$$

in which case the third term in the free energy becomes

$$\sum_m^M \mathbb{E}_{\check{\boldsymbol{X}}_t,\boldsymbol{U}_{t-1},\boldsymbol{Y}_t}\Big[\lambda_m(\hat{\boldsymbol{X}}_t) - Y_t^m \log \lambda_m(\hat{\boldsymbol{X}}_t)\Big].$$

Establishing recurrence. So far, despite the notation, the model is static. We now identify the random variables. In particular, we let $\hat{\boldsymbol{X}}_t$ be the latent state, $\hat{\boldsymbol{Y}}_t$ be the observations, and $\hat{\boldsymbol{U}}_t$ be *the sufficient statistics for* $\hat{\boldsymbol{X}}_t$. Recall that the sufficient statistics are any functions of the data—in this case, $\check{\boldsymbol{U}}_{t-1}$ and $\hat{\boldsymbol{Y}}_t$—that throw away no information about the underlying random variable—in this case, $\hat{\boldsymbol{X}}_t$. Thus, if the recognition model matches the true posterior of the generative model, then the mean and variance functions $\boldsymbol{\nu}_{\check{x}}(\check{\boldsymbol{U}}_{t-1},\boldsymbol{Y}_t)$ and $\boldsymbol{\Upsilon}_{\check{x}}(\check{\boldsymbol{U}}_{t-1},\boldsymbol{Y}_t)$ are sufficient for $\hat{\boldsymbol{X}}_t$. Accordingly, we define $\check{\boldsymbol{U}}_t$ to be the concatenation of these two functions.

Intuitively, the sufficient statistics provide a "summary" of $\check{\boldsymbol{U}}_{t-1}$ and $\boldsymbol{Y}_t$, or more precisely of their information about the latent state. Since $\check{\boldsymbol{U}}_{t-1}$ is itself another summary, in this case of $\check{\boldsymbol{U}}_{t-2}$ and $\boldsymbol{Y}_{t-1}$, the argument can be extended recursively to claim that $\check{\boldsymbol{U}}_t$ summarizes *all* of the preceding observations as they pertain to the latent state (Makin et al., 2016), at least up to the capacity of this vector. This makes it a good candidate to be explained by the generative model if we want it to learn how information propagates forward as well as backward in time.

3. Experiments

We consider three experiments, one quantitative and two qualitative.

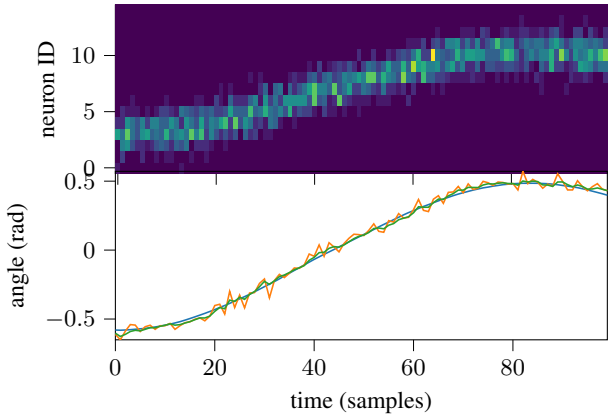

Figure 3. **PPCs experiment (example).** Upper panel: The observed data, consisting of 15 "neurons" responding to the position of an object moving with underdamped, linear, second-order dynamics. Lower panel: Position as a function of time: ground-truth (blue), as decoded from the observations (orange), and as decoded from the reconstructed (denoised) observations (green).

Probabilistic population codes (PPCs). We consider a simple data generator that starts with a one-dimensional, underdamped (oscillatory), second-order dynamical system that is driven by noise. The position of this moving object is then "reported" by a crude model of a population of 15 neurons, with Gaussian-shaped tuning curves that uniformly tile the space of positions (interpreted to be angles). The tuning curves provide the mean to a Poisson distribution from which spike counts are drawn (Ma et al., 2006; Makin et al., 2016). This data set has three nice properties: (1) Since only position is observed, we expect models lacking a backward flow of information to learn only first-order dynamics, despite the fact that first-order systems cannot oscillate. (2) The model is simple, but the relationship between observations and state is still nonlinear. (3) Nevertheless, after applying the appropriate nonlinear transforms to the observations, a closed-form inference procedure is available in the form of a modified Kalman filter (Beck et al., 2011; Makin et al., 2015). This allows us to quantify how close to optimally position information is encoded in (or anyway can be decoded from) the latent state of our models.

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}

Table 1. Mean square 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

Table 2. Mean square errors (MSE) for the bouncing-ball dataset. All cases use a single trajectory for each batch

Bouncing balls. Three balls move at constant speeds and (see Fig. 4) bounce off each other and the perimeter of the frame with complete energy conservation (Sutskever & Hinton, 2007). As with the preceding data set, we expect models that lack a backward flow of information to fail to learn that the velocities are constant (until collisions), since this requires learning a second-order dependency.

MovingMNIST. An extension of MNIST, it introduces dynamic elements with sequences of frames. Digits exhibit diverse motions like translations, rotations, and scaling. Interactions between multiple digits and collisions with the boundaries simulate real-world scenarios. It serves as a benchmark for video analysis, motion prediction, and object tracking. We expect similar results to the bouncing balls.

Details of the VAEs. For our experiments involving the bouncing balls and the MovingMNIST dataset, we employ an architecture inspired by the DCGAN (Radford et al., 2016). In particular, our encoder network (recognition model) and decoder network (generative model) are composed, respectively, of convolutional and deconvolution layers. However, only the images $\boldsymbol{y}_t$ pass through the convolutional layers of the encoder, after which they are concatenated with the (unprocessed) previous sufficient statistics, $\breve{\boldsymbol{u}}_{t-1}$, i.e. the vector of posterior means and variances (see Eq. 2) at time $t-1$. This combined vector is then passed through a fully-connected layer with ReLU (**?**) activations. To obtain the cumulants of the posterior distribution at time t, the output is then passed through two different fully-connected layers, one for the mean and the other for the variance.

For the PPC dataset, both the encoder and decoder are instead composed of two fully connected layers. As in for the bouncing-ball and the MovingMNIST datasets, the PPC observations are passed through this portion of the encoder and then concatenated with the previous sufficient statistics. This combined vector is subsequently passed through a fully connected layer with non-linear activation.

All updates to models were made by stochastic gradient descent with AdaM optimization (Kingma & Welling, 2014).

4. Results

To evaluate models trained on the PPC dataset, we compute the posterior mean under the recognition model at all time steps of a "trajectory." From these we compute the expected value of the observations under a Poisson emission. Critically, this "updated" version of the observations now contains information from the previous sufficient statistics (see again Fig. 2) and therefore can—if the model is good—provide a better estimate of the underlying position being encoded. This estimate is computed with the "center of mass" of the population, i.e weighting each neuron's preferred angle by its number of spikes and normalizing, which is the optimal estimate of the encoded variable (Dayan & Abbott, 2001). Fig. 3 shows a typical example.

To quantify performance, we compare against the optimal inference algorithm, the Kalman filter (see Section 3), when applied to state-space models acquired wih EM. In particular, we acquire both a first- and a second-order model. Table 1 shows that, as expected, models trained according to our procedure (**rVAE** and **rEFH**) come close to the optimal learned Kalman filter (KF-2). Removing the distribution over $\hat{\boldsymbol{U}}_{t-1}$ turns the rEFH into a model introduced by (Sutskever & Hinton, 2007) as the TRBM. Accordingly, we call the corresponding variant of the VAE the TVAE. Although these models are better than decoding from the uncorrected observations alone ("order-0"), they cannot outperform the optimal first-order model (KF-1). Allowing BPTT in the TRBM turns it into the RTRBM (Sutskever et al., 2009; Sutskever, 2013), which restores ability to learn second-order dynamics, as expected.

In the case of bouncing balls, we ask how well each model can predict the next frame. To make predictions, we run the model forward on an input sequence up till time t, and then use clamped Gibbs sampling to establish the next hidden state, $\hat{\boldsymbol{X}}_{t+1}$. Finally, we noiselessy generate a "sample" $\hat{\boldsymbol{y}}_{t+1}$, and compare it to the actual next frame, $\boldsymbol{y}_{t+1}$. Results are reported in Table 2. Again the rEFH performs nearly as well as the RTRBM, despite omitting BPTT; whereas

predictions from the TRBM, trained without BPTT or a distribution over the previous hidden state, are much worse, close to the prediction provided by the previous frame ("order 0"). This is consistent with the PPC results: Despite moving at a fixed, non-zero speed, the balls *average* zero velocity over long trajectories. Consequently, the optimal first-order prediction is the same as the optimal zeroth-order prediction: the previous frame. So it again appears that the TRBM learns first-order dynamics while the rEFH and RTRBM learn something second-order.

In our experiments to date, clamped Gibbs sampling under the rVAE does not converge, so we were unable to generate predictions from this model. However, rVAE (like the rEFH) does not require Gibbs sampling for generation *backwards in time*, so we instead evaluate it qualitatively by examining generated trajectories. Fig. 4 shows frames from one such trajectory. The three balls follow anticipated trajectories before and after colliding with each other and the wall, demonstrating that the model has captured both the constant movement directions in the absence of collisions and the effects of those collisions, which suggests that it has learned something second-order.

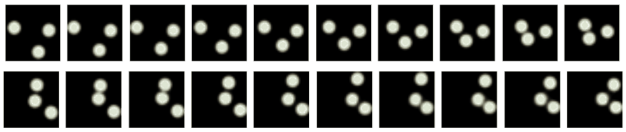

Figure 4. Snapshots of generated trajectory with the Bouncing Ball dataset

For MovingMNIST dataset, trajectories generated by the rVAE correctly bounce of walls (Fig. 5) and pass through one another.

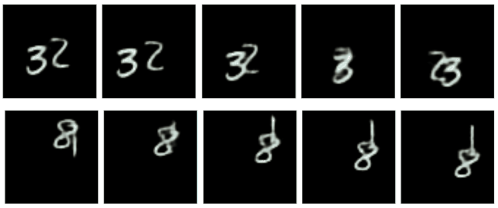

Figure 5. Snapshots of generated trajectory with the MovingMNIST dataset

5. Conclusions

The proposed approach of learning the joint distribution over current and previous states presents a promising alternative to computationally expensive backpropagation through time. This study successfully verified the effectiveness of this principle on a toy dataset, demonstrating its ability to capture essential aspects that otherwise require BPTT. Furthermore, qualitative evaluations with more sophisticated datasets—the bouncing-ball dataset and MovingMNIST—highlight the models' capacity to effectively capture nonlinear second-order dynamics. These findings showcase the potential of the proposed method to overcome computational challenges and open up new possibilities for more biologically inspired modeling of sequential data.

Nevertheless, although intuitively plausible, our procedure lacks a rigorous mathematical justification. It also remains to scale the procedure up to more challenging datasets.

References

Beck, J. M., Latham, P. E., and Pouget, A. Marginalization in Neural Circuits with Divisive Normalization. *The Journal of Neuroscience*, 31(43): 15310–9, oct 2011. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.1706-11.2011. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3230133{&}tool=pmcentrez{&}rendertype=abstract.

Boulanger-Lewandowski, N., Vincent, P., and Bengio, Y. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1159–1166, 2012.

Dayan, P. and Abbott, L. *Theoretical Neuroscience*. MIT Press, 2001.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.5.889.

Dempster, A., Laird, N., and Rubin, D. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977. ISSN 15289036. doi: 10.1115/1.3424485.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4528–4547, 2019.

Kingma, D. P. and Ba, J. L. Adam: A Method for Stochastic Optimization. 2014. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, pp. 1–14, 2014. URL http://arxiv.org/abs/1312.6114.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. Bayesian Inference with Probabilistic Population Codes. *Nature Neuroscience*, 9(11):1423–1438, 2006. doi: 10.1038/nn1790. URL http://www.nature.com/neuro/journal/v9/n11/abs/nn1790.html.

Makin, J., Dichter, B., and Sabes, P. Recurrent Exponential-Family Harmoniums without Backprop-Through-Time. **In revision**, JMLR; preprint available at https://arxiv.org/abs/1605.05799, 2016.

Makin, J. G., Dichter, B. K., and Sabes, P. N. Learning to Estimate Dynamical State with Probabilistic Population Codes. *PLoS Computational Biology*, 11(11):1–28, 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004554.

Mittelman, R., Kuipers, B., Savarese, S., and Lee, H. Structured Recurrent Temporal Restricted Boltzmann Machines. In *ICML*, volume 32, pp. 1647–1655, 2014. ISBN 9781634393973. URL http://jmlr.org/proceedings/papers/v32/mittelman14.pdf.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte Carlo Gradient Estimation in Machine Learning. *Journal of Machine Learning Research*, 21:1–62, 2020.

Neal, R. M. and Hinton, G. E. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 1998. URL http://books.google.com/books?hl=en{&}lr={&}id=7f61BBKdJ4EC{&}oi=fnd{&}pg=PA355{&}dq=A+View+of+the+EM+Algorithm+that+Justifies+Incremental,+Sparse,+and+Other+Variants{&}ots=PUzDK-NGil{&}sig=1WziWdM6K6xzf8j-p0ZYjuzxDqs.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of The 31st ...*, 32:1278–1286, 2014. URL http://jmlr.org/proceedings/papers/v32/rezende14.html{%}5Cnpapers3://publication/uuid/F2747569-7719-4EAC-A5A7-9ECA9D6A8FE6.

Saxena, V., Ba, J. L., and Hafner, D. Clockwork Variational Autoencoders. *Advances in Neural Information Processing Systems*, 35:29246–29257, 2021. ISSN 10495258.

Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, chapter 6, pp. 194–281. MIT Press, 1986. URL http://www-psych.stanford.edu/{~}jlm/papers/PDP/Volume1/Chap6{_}PDP86.pdf.

Sutskever, I. *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, 2013. URL http://www.cs.utoronto.ca/{~}ilya/pubs/ilya{_}sutskever{_}phd{_}thesis.pdf.

Sutskever, I. and Hinton, G. E. Learning Multilevel Distributed Representations for High-Dimensional Sequences. In *AISTATS*, pp. 1–8, 2007.

Sutskever, I., Hinton, G. E., and Taylor, G. W. The Recurrent Temporal Restricted Boltzmann Machine. In *Advances in Neural Information Processing Systems 21: Proceedings of the 2008 Conference*, pp. 1–8, 2009. URL http://papers.nips.cc/paper/3567-the-recurrent-temporal-restricted-boltzmann-

Werbos, P. J. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988. ISSN 08936080. doi: 10.1016/0893-6080(88)90007-X.