Modeling Retinal Ganglion Cells with Neural Differential Equations (Student Abstract)

Kacper Dobek[1], **Daniel Jankowski**[1], **Krzysztof Krawiec**[1]

[1]Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
kdobek@cs.put.poznan.pl, jankowskidaniel06@gmail.com, krawiec@cs.put.poznan.pl

Abstract

This work explores Liquid Time-Constant Networks (LTCs) and Closed-form Continuous-time Networks (CfCs) for modeling retinal ganglion cell activity in tiger salamanders across three datasets. Compared to a convolutional baseline and an LSTM, both architectures achieved lower MAE, faster convergence, smaller model sizes, and favorable query times, though with slightly lower Pearson correlation. Their efficiency and adaptability make them well suited for scenarios with limited data and frequent retraining, such as edge deployments in vision prosthetics.

Code — https://github.com/JankowskiDaniel/Neural-Deep-Retina

Datasets — https://purl.stanford.edu/rk663dm5577

Extended version — https://arxiv.org/abs/2511.18014

Introduction

Neural Ordinary Differential Equations (NODEs) architectures (Chen et al. 2019) promise more precise, fine-grained and plausible modeling of time-dependent phenomena. In this study, we design models based on two recent NODEs, Liquid time-constant neural networks (LTCs) (Hasani et al. 2021) and Closed-form Continuous-time neural networks (CfCs) (Hasani et al. 2022) and apply them to the challenging task of modeling neural activity of retinal ganglion cells (RGCs) of tiger salamanders, following up the work by (Maheswaranathan et al. 2023). Each of the three datasets used therein comprises a monochrome video sequence of natural scenes and synthetic noise images observed by the animal (Fig. 1a), accompanied by time series collected from $n = 9$, 14, or 27 chosen RGCs (Fig. 1b). Each data point in the time series aggregates neural firings in a 10 ms window; despite this aggregation, the time series are highly irregular and change abruptly (Fig. 1b). The training, validation and test set comprise respectively 280,000, 89,802 and 5,996 frames. See Supplementary Material (SM) Sec. A for more details.

To model RCGs' predictions, Maheswaranathan et al. (2023) designed a simple ConvNet comprising a stack of convolutional layers followed by a dense layer. We replicate this architecture and compare its performance with LTCs, CFCs and LSTMs (Hochreiter and Schmidhuber 1997). All architectures are equipped with the same convolution stack processing 40 consecutive frames concatenated into a single 40-channel image. For ConvNet, this is followed by a single dense layer, with units corresponding to the n time series. For the remaining architectures, the conv stack is followed by a 32-dimensional dense layer: in LSTM, this latent vector is then fed into an LSTM cell and two dense layers; in NODEs, it becomes direct input to LTC or CfC, configured to match the output dimensionality n. The models were trained using Adam optimizer (Kingma and Ba 2017) and MSE as the loss function (more details in SM Sec. B).

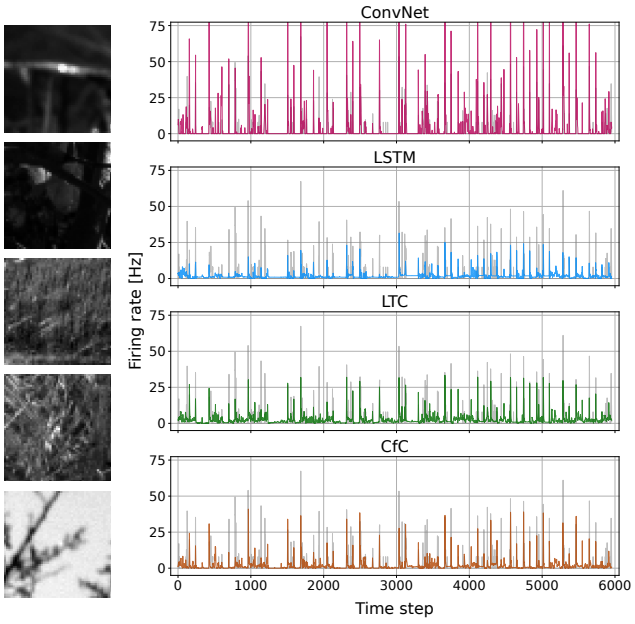

Figure 1: (a) Selected input frames from the dataset, and (b) Test-set predictions for the 6th channel of the rgc9 dataset (ground truth in grayscale).

Results

In test-set evaluation (Table 1), NODEs proved superior to ConvNet and LSTM in terms of MAE on all three datasets.

Dataset	Model	$\rho\uparrow$	95% CI	MAE$\downarrow$	#params	Time [s]	ANOVA p-value
rgc9	*ConvNet*	0.569±0.005	[0.564, 0.574]	4.07±0.90	227,251	2,416±196	5.187×10^{-8}
	LSTM	0.421±0.038	[0.384, 0.457]	3.18±0.11	46,761	4,759±414	
	LTC	0.480±0.012	[0.469, 0.491]	2.73±0.12	49,442	1,823±360	
	CfC	0.474±0.006	[0.468, 0.479]	2.86±0.07	47,496	2,411±387	
rgc14	*ConvNet*	0.586±0.003	[0.583, 0.588]	6.80±2.64	281,346	2,048±326	6.522×10^{-7}
	LSTM	0.484±0.034	[0.451, 0.517]	2.72±0.08	49,670	5,234±24	
	LTC	0.569±0.002	[0.566, 0.571]	2.19±0.04	54,116	1,824±159	
	CfC	0.555±0.009	[0.546, 0.564]	2.35±0.06	50,752	2,673±300	
rgc27	*ConvNet*	0.586±0.015	[0.572, 0.600]	4.72±1.89	421,993	2,395±262	1.633×10^{-12}
	LSTM	0.399±0.017	[0.382, 0.416]	3.62±0.20	53,563	5,019±93	
	LTC	0.529±0.001	[0.519, 0.539]	2.94±0.06	59,830	2,528±474	
	CfC	0.517±0.007	[0.511, 0.524]	3.03±0.13	55,144	3,259±610	

Table 1: The correlation coefficient (ρ) with 95% confidence intervals and MAE (both aggregated over output channels), the number of parameters, the training time, averaged over 5 train-test runs (mean $\pm$ std), and the ANOVA p-value.

However, they yielded to the ConvNet in terms of correlation coefficient. We tested the normality of residuals and homogeneity of variance, and performed ANOVA for each dataset, revealing a significant difference between the performance of the ConvNet and other models (see Fig. 2 for a comparison of *rgc9* results). The reason behind this discrepancy is that NODEs turned out to be better in terms of predicting the *values* of dependent variables, but worse in terms of the *timing* of peaks (Fig. 1b). Notably, NODEs achieve this level of performance with 5x-8x fewer parameters, and require similar training time as the ConvNet, with the LSTM faring worse on this metric.

In the Supplementary Material, we provide the details on the experimental setup (Secs. A-E) and report more results. We found NODEs to converge very quickly compared to the other architectures (Sec. G) and have favorable querying times (Sec. K). When tested on time series perturbed with noise, ConvNet and LSTM proved more robust (Sec. J). We devised also a variant of the architecture equipped with multiscale temporal representation (Sec. H), but it attained worse results than those reported in Table 1.

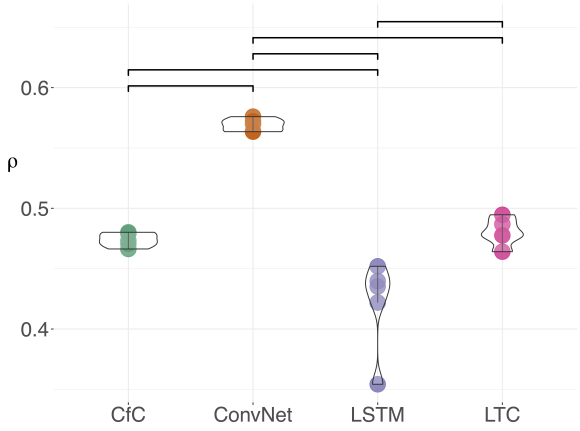

Figure 2: Correlation of the five test-set runs for each model trained on *rgc9* showing significant differences (top).

Conclusion

While NODEs considered in this study proved superior to simpler architectures (ConvNet and LSTM) in terms of quantitative prediction (MAE), training time, and model size (and thus interpretability), they did not excel on all metrics, in particular yielding on the time-wise precision of predictions (ρ). We hypothesize two main causes behind this phenomenon: (i) the highly irregular and discontinuous nature of considered time series, which may be hard to model with ODEs, and (ii) the apparent absence of complex long-term dependencies in the RGC responses, except for rather obvious and relative constant lag of the neural response (Fig. SM 2). Concerning the latter, the constrained, time-agnostic ConvNets might be sufficient to model the essential input-output dependencies present in the data, while being more robust to noise and confounding variables, which form a fair share of the measurement in these observations. NODEs and LSTM, in contrast, are designed to seek more sophisticated temporal dependencies, which may turn out to be spurious and negatively impact generalization in this particular case study. As we showed in SM Sec. F, increasing the input sequence length decreases the performance of NODEs and the LSTM, which may suggest that neuronal responses are characterized by relatively low and and fixed lags, which fit into the adopted 40-frame window and make more intricate modeling unnecessary. The evidence presented in the SM partially supports these claims, but a more in-depth investigation (possibly involving synthetic data with controlled amount of noise and confounders) is necessary to ascertain this proposition.

An interesting advantage of NODEs demonstrated in this study is their capacity to learn and converge quickly (SM Sec. G). This makes them particularly attractive in scenarios where models need to be quickly or/and frequently updated and where training data is scarce, like edge deployments on specialized hardware, which could be of potential value in vision prosthetics.

Acknowledgments

Research supported by the statutory funds of Poznan University of Technology and the Polish Ministry of Science and Higher Education, grant no. 0311/SBAD/0770 and the Research Grant of National Science Center, grant no. 2025/57/B/ST6/03737.

References

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2019. Neural Ordinary Differential Equations. arXiv:1806.07366.

Hasani, R.; Lechner, M.; Amini, A.; Liebenwein, L.; Ray, A.; Tschaikowski, M.; Teschl, G.; and Rus, D. 2022. Closed-form continuous-time neural networks. *Nature Machine Intelligence*, 4(11): 992 – 1003. Cited by: 34; All Open Access, Hybrid Gold Open Access.

Hasani, R.; Lechner, M.; Amini, A.; Rus, D.; and Grosu, R. 2021. Liquid Time-Constant Networks. In *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, volume 9A, 7657 – 7666. Cited by: 84.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735 – 1780. Cited by: 74221.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Maheswaranathan, N.; McIntosh, L. T.; Tanaka, H.; Grant, S.; Kastner, D. B.; Melander, J. B.; Nayebi, A.; Brezovec, L. E.; Wang, J. H.; Ganguli, S.; et al. 2023. Interpreting the retinal neural code for natural scenes: From computations to neurons. *Neuron*, 111(17): 2742–2755.