2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2020)
Toronto, Canada, October 11-14, 2020

Learning Timescales in Gated and Adaptive Continuous Time Recurrent Neural Networks

Stefan Heinrich*, Tayfun Alpay†, and Yukie Nagai*
*Cognitive Developmental Robotics Lab, IRCN, The University of Tokyo, Tokyo, Japan
†Knowledge Technology, Dept. of Informatics, Universität Hamburg, Hamburg, Germany
heinrich.stefan@ircn.jp, alpay@informatik.uni-hamburg.de, nagai.yukie@mail.u-tokyo.ac.jp

***Abstract*—Recurrent neural networks that can capture temporal characteristics on multiple timescales are a key architecture in machine learning solutions as well as in neurocognitive models. A crucial open question is how these architectures can adopt both multi-term dependencies and systematic fluctuations from the data or from sensory input, similar to the adaptation and abstraction capabilities of the human brain. In this paper, we propose an extension of the classic Continuous Time Recurrent Neural Network (CTRNN) by allowing it to learn to gate its timescale characteristic during activation and thus dynamically change the timescales in processing sequences. This mechanism is simple but bio-plausible as it is motivated by the modulation of oscillation modes between neural populations. We test how the novel Gating Adaptive CTRNNs can solve difficult synthetic sequence prediction problems and explore the development of the timescale characteristics as well as the interplay of multiple timescales. As a particularly interesting finding, we report that timescale distributions emerge, which simultaneously capture systematic patterns as well as spontaneous fluctuations. Our extended architecture is interesting for cognitive models that aim to investigate the development of specific timescale characteristic under temporally complex perception and action, and vice versa.**

***Index Terms*—CTRNN, Recurrent Neural Network, Timescale, Adaptive, Gating, Cognitive Model**

I. Introduction

Recurrent Neural Network (RNN) architectures are an important building block in recent machine learning advances as well as in neurocognitive modelling approaches. In machine learning, key goals with huge potential for assistive systems are learning representations from huge repositories of complex temporal data, such as speech or dialogues, as well as building algorithms that can make use of these representations in understanding, translation, and discourse. Neural network architectures are pursued that are specifically effective in capturing events on multiple and strongly varying timescales and include particular architectural constraints. For instance, recurrent networks with gating mechanisms are used to learn to control the temporal extent and abstraction in order to capture characteristics on multiple timescales in the data [1]–[4]. Here, gating is directly integrated into the neurons' activation and optimised for circumventing the vanishing gradient problem in gradient descent. Distinct hierarchical layer stacking has been researched for many years as a way of constraining universal recurrent neural networks into a structure that is biased towards learning temporally hierarchical dependencies [5], [6].

In computational neuroscience, mechanisms of processing temporal information are studied in order to better understand the most complex recurrent neural network: the human brain. An open question is, which mechanisms are underlying the processing of sensory input with complex temporal dynamics as well as generating precise motor sequences from high-level behavioural intentions in interplay with consistent world models. As crucial mechanistic components, hypotheses from neuroscience suggest neural oscillations, a highly complex interplay of neural populations and local integrations by mode coupling, and multiple timescales in hierarchical processing streams [7]–[9]. However, it is vastly open, *how* these mechanisms are working on a computing and processing level.

In addition, biologically plausible artificial neural network models on cortex level allow an investigation of the development of cognitive functions as well as atypical information processing characteristics in the brain [10], [11]. Specifically, in interdisciplinary research in-between computer science, cognitive psychology, and computational neuroscience, neural models and simulations are used to explain effects in behavioural and brain imaging data [12]. Here, bio-plausible recurrent neural models are adopted and studied in-depth for replicating and extrapolating behavioural observations, for example in studying people with psychiatric symptoms including autism spectrum condition and schizophrenia.

In this paper, we propose a mechanism for RNNs that allows for learning timescale characteristics from the data in order to better capture both short-term fluctuations and long-term dependencies. Specifically, we follow up on a previous approach of adapting the timescale constant in Continuous Time Recurrent Neural Networks (CTRNNs) while learning from input sequences [13]. With a novel gating mechanism, the neurons in this CTRNN can distinctly vary their timescales based on presynaptic input. Since this mechanism is inspired by the brain's general adaptation as well as dynamic tuning to sensorimotor information during learning sequential information, it is a candidate for simulating the interplay of coupled timescale modes and testing for the emergence of individually different timescale structures. We explore our novel mechanism on synthetic as well as behavioural prediction tasks in order to investigate how timescale structures and dynamics form based on the temporal characteristics in the data.

This work was supported by JST CREST "Cognitive Mirroring" (Grant Number: JPMJCR16E2) including AIP challenge program, Japan, and by the World Premier International Research Center Initiative (WPI), MEXT, Japan.

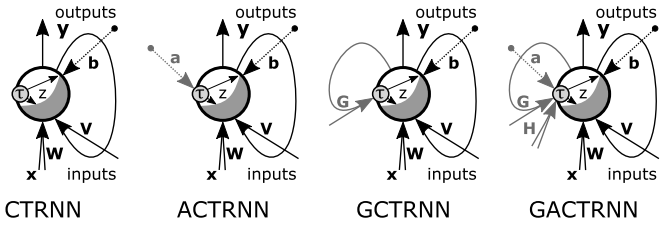

Fig. 1. The neurons' characteristics of the CTRNN variants: A timescale value τ steers how strongly or weakly a neuron is leaking, thus how fast or slow it is forgetting its previous activation. The ACTRNN introduces an adaptive bias to the timescale, while the GCTRNN connects the timescale via gates to the presynaptic input.

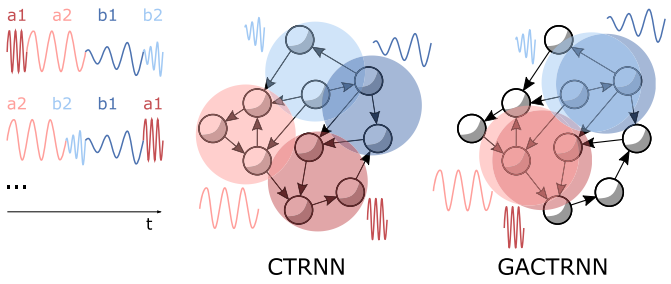

Fig. 2. Effect of GACTRNNs: by changing their timescales during processing, neurons can learn to simultaneously represent temporally different primitives.

II. Continuous Time Recurrent Neural Network Models

One of the models that are seen biologically plausible is the Continuous Time Recurrent Neural Network (CTRNN), which can be derived from the leaky integrate-and-fire model and thus from a simplification of the Hodgkin-Huxley model from 1952. For computational modelling, this network architecture was independently developed by Hopfield and Tank in 1986 as a nonlinear graded-response neural network and by Doya and Yoshizawa in 1989 as an adaptive neural oscillator [14], [15]. The activation $\mathbf{y}$ of CTRNN units is defined as follows:

$$\mathbf{y}_t = f\left(\mathbf{z}_t\right) \quad , \tag{1}$$

$$\mathbf{z}_t = \left(1 - \frac{\Delta t}{\tau}\right)\mathbf{z}_{t-\Delta t} + \frac{\Delta t}{\tau}\left(\mathbf{W}\mathbf{x} + \mathbf{V}\mathbf{y}_{t-\Delta t} + \mathbf{b}\right) \quad , \tag{2}$$

for inputs $\mathbf{x}$, previous internal states $\mathbf{z}_{t-\Delta t}$, weights $\mathbf{W}$ and $\mathbf{V}$, bias $\mathbf{b}$, and an activation function f. The *timescale* parameter τ expresses the leakage within a certain time Δt. Thus in tasks with discrete numbers of time steps, the CTRNN can get employed as a discrete model, e.g. by setting $\Delta t = 1$. In this case, Equation 2 simplifies as follows:

$$\mathbf{z}_t = \left(1 - \frac{1}{\boldsymbol{\tau}_t}\right)\mathbf{z}_{t-1} + \frac{1}{\boldsymbol{\tau}_t}\left(\mathbf{W}\mathbf{x} + \mathbf{V}\mathbf{y}_{t-1} + \mathbf{b}\right) \quad . \tag{3}$$

A. Gated Adaptive CTRNNs

In the original definition of the CTRNN as a computational model, the timescale can be a pre-determined constant parameter τ for all units or a vector $\boldsymbol{\tau}$ of individual constants. On this basis, a range of modifications are possible to directly steer the timescales as an adaptive result of learning or even an adaptive gating mechanism (see Figure 1).

In previous work [16], these individual constants have been replaced by learnable weights $\mathbf{a}$ which work like *adaptive* timescale biases for the neurons:

$$\boldsymbol{\tau}_t = \boldsymbol{\tau}_t^{\mathrm{A}} = 1 + \exp\left(\mathbf{a} + \boldsymbol{\tau}_0\right) \quad . \tag{4}$$

This Adaptive CTRNN (ACTRNN) embeds the learnable $\mathbf{a}$ in an exponential function to ensure that i) the timescales stay in $[1, \infty]$ and ii) the neurons' characteristics remain fully differentiable. The vector $\boldsymbol{\tau}_0$ allows for sensible initial values for the timescales. After training, these adaptive timescale biases lead a fine-grained distribution of timescale values over all neurons.

As a novel extension, the timescales can be steered based on weighted recurrent input by introducing additional recurrent weights $\mathbf{G}$ directly to the timescale parameter:

$$\boldsymbol{\tau}_t = \boldsymbol{\tau}_t^{\mathrm{G}} = 1 + \exp\left(\mathbf{G}\mathbf{y}_{t-1} + \boldsymbol{\tau}_0\right) \quad . \tag{5}$$

In this Gated CTRNN (GCTRNN), these weights operate as *gates* on the neuron's leakage characteristic, effectively controlling during activation whether the neurons should leak strongly or weakly and thus quickly update their activations or conserve their internal states for a longer time. Thus, compared to the adaptive timescale biases, which are different for the individual neurons but constant during activation, the gating allows for arbitrary timescale changes in every time step.

In order to complete the gate characteristic, we can further introduce additional weights $\mathbf{H}$ from the input and also include the timescale biases $\mathbf{a}$:

$$\boldsymbol{\tau}_t = \boldsymbol{\tau}_t^{\mathrm{GA}} = 1 + \exp\left(\mathbf{H}\mathbf{x} + \mathbf{G}\mathbf{y}_{t-1} + \mathbf{a} + \boldsymbol{\tau}_0\right) \quad . \tag{6}$$

Consequently, this Gated and Adaptive CTRNN (GACTRNN) can fully self-organise its leakage characteristic based on the temporal dynamics.

B. Introducing Temporal Constraints

Alternative to defining one arbitrarily large recurrent layer, the CTRNN and the novel GACTRNN neurons can be organised a priori in a constrained horizontal and vertical fashion. For instance, in all proposed CTRNN variants (henceforth called xCTRNN) the neurons can be grouped in horizontal *modules* that are defined with specific fixed timescale constants (in the CTRNN) or roughly initialised with reasonable estimates (in the GACTRNN, for easing the training), where a simple setup can be defined with exponentially increasing values ($\tau = 1, 2, 4, \ldots$). Additionally, these modules can be interconnected recurrently using different connectivity strategies such as *dense* (fully connected), *adjacent* (only connected with the next slower and next faster module), *clocked* (only connected to faster modules [17]), or *partitioned* (no connections between modules). By these means, we can enforce a structure of e.g. slower leaking neurons modulating the activation of faster leaking neurons [16].

Analogously, we can divide the xCTRNN vertically into different layers that are stacked or interconnected with shortcuts and apply the same strategies for initialisation and connectivity. Here, predefined timescales can exponentially increase from the first vertical layer after the input up to the last layer and thereby enforce a structure of hierarchical decomposition or composition, similar to Multiple Timescale Recurrent Neural Networks with context bias or context abstraction (MTRNNs, compare [5], [9], [13]).

Thus, depending on the purpose of the computational model and the task, the GACTRNN can overall be set up to learn both connection weights and timescales from the input data only or get an a priori constraint of a specific structure depending on knowledge about the temporal dynamics in the data. In effect, neurons can adjust their timescales to better represent long-term or short-term dependencies (see Fig. 2).

III. Evaluation

Conceptually similar to the Gated Recurrent Unit (GRU [18]), the novel GACTRNN can learn to forget and conserve internal states but remains closer to the biologically plausible integrate-and-fire models. In order to provide an exploration of the properties of the novel gated and adaptive CTRNN architectures, we evaluated the different variants on several prediction tasks with well-known or well-definable temporal characteristics.

As a general architecture in this study, we defined networks with one hidden layer, consisting of multiple modules. For example, a layer of $h = (m_1, m_2, m_3)$ neurons means it has 3 adjacent modules, where recurrent connections are fully connected to each other (dense connectivity) or only to neighboring modules (adjacent connectivity). All hidden neurons were densely (fully) connected to the input, activate with a TANH function, and connect densely to a linearly activated output layer. The same input and recurrent connectivity patterns were set up for the timescale gates in the cases of the GCTRNN and GACTRNN variants. During training, the loss was calculated using a Mean Squared Error (MSE).

For all following tasks, we optimised hyperparameters such as layer size and initial timescale on the CTRNN in order to reduce bias. We systematically tested networks with 3, 4, and 5 modules of linearly or logarithmically increasing numbers and initial timescales. In a subsequent test, we used the same hyperparameters on the CTRNN, ACTRNN, GCTRNN, GACTRNN, as well as the Simple Recurrent Network (SRN, also known as Elman network), and the GRU. We repeat each run 10 times with a new random initialisation of the network parameters (weights) based on a random seed, which is exactly the same for all model variants.

Our general hypothesis is that the xCTRNN variants perform better than the classical CTRNN because the adaptation leads to a more fine-grained distribution of timescales and the gating allows neurons to represent different temporal characteristics simultanously. The purpose of also comparing them to the SRN and GRU is to get a general intuition of the difficulty in learning the task, where the SRN stands for the naive approach and the GRU for the currently best performing basic architecture with a gating mechanism in machine learning tasks. Our aim for the analyses on the following tasks is to explore emerging differences in timescale characteristics and to inspect how these timescales affect the activation and thus the performance of the networks.

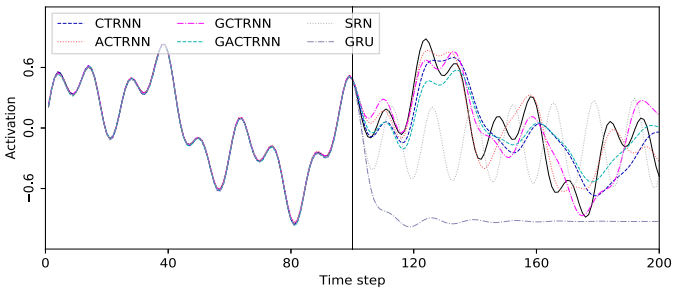

Fig. 3. Prediction of the superposition of three sine waves with periods $100, 30, 12$. Prediction is in open loop from time step 100 onward.

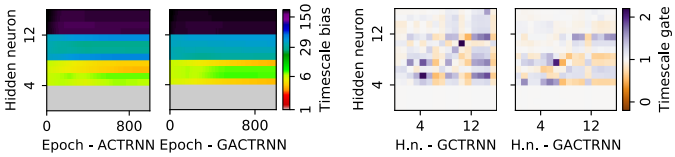

Fig. 4. Emergence of specific adaptive timescale distributions (biases) and recurrent timescale gating matrix from uniform zero initialisation (last epoch).

A. Superposed Sine Wave

To test if the xCTRNNs can capture input patterns on multiple timescales, in the first task we defined a sine wave superposition (SSP) of three sines with amplitudes of $(1.0, 0.5, 0.75)$, periods of $(100, 30, 12)$, and an overall length of 200 steps. The networks' task was to learn this SSP sequence and predict it in an open-loop fashion from time step 100 onward (meaning no ground truth is fed in but only its own output as recurrent input). The best performing CTRNN was found with $(4, 4, 4, 4)$ neurons in a dense connection, timescales of $(1, 6, 36, 216)$, and a learning rate of 0.001 using *RMSprop* for training over 1.000 epochs.

This task is suitable because it is easy for any RNN to memorize the sequence but very hard to capture the underlying characteristic of independent rhythms. Consequently, we confirmed that both the SRN as well as the GRU are generally not able to capture the underlying frequencies well and tend to oscillate with a period close to the weighted mean of our SSP's periods and varying amplitudes. Fig. 3 shows the predictions of the baseline networks, the basic CTRNN, and the novel xCTRNN variants. We found that a good prediction is still depending on the initialisation and the ideal hyper-parameter setting to find the narrow region between underfitting and overfitting. Here, a good timescale setting (in case of the baseline CTRNN) seems to allow the network to roughly capture the characteristics of the individual sines. The xCTRNN variants do not show a significant improvement (or degradation) with the same network sizes.

TABLE I
PERFORMANCE COMPARISON (MAE) IN LISSAJOUS CURVES PREDICTION.

SRN	GRU	CTRNN	ACTRNN	GCTRNN	GACTRNN
0.27281	0.04493	0.02109	0.01982	0.00678	0.00560
±0.00582	±0.00566	±0.00038	±0.00042	±0.00021	±0.00011

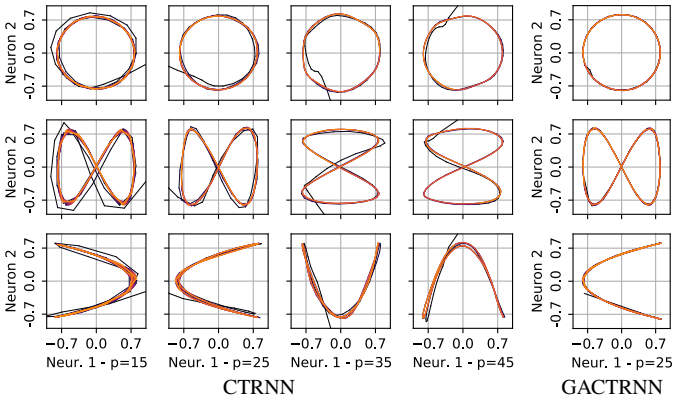

Fig. 5. Prediction of Lissajous Curves with various periods p of 15, 25, 35, and 45, where prediction is plotted from black to orange over time. For the GACTRNN, the results for all periods are visually identical to $p = 25$.

In line with previous results [16], we observed in the analysis that the adaptive CTRNNs are updating the timescale bias during training towards a more diverse and fine-grained timescale distribution, compared to the ideal constant timescales of the CTRNN (see Fig. 4 on the left). In the gating CTRNNs, a timescale gate matrix emerges that shows transitive timescale modulation mostly for adjacent modules, where the differences in the gate matrices for the GCTRNN and the GACTRNN are small. It seems that neurons with a certain timescale bias most strongly modulate the timescales of neurons with not vastly smaller or larger timescale bias.

B. Lissajous Curves with Various Period Lengths

For further investigating how timescales are modified time-step-wise in the gated CTRNNs, we defined a task with systematic temporal differences, thus sequences with patterns that extend differently over time steps. In particular, we generated twelve Lissajous Curves similar to the study in [19] but defined the three different resulting shapes with four different periods p of 15, 25, 35, and 45 over lengths of 200 time. We hypothesise that to better solve the task, the networks must learn the general characteristic of the three shapes as well as the underlying *primitives* of different temporal lengths.

When inspecting the performance in prediction (in this task for all 200 steps in a closed loop) we found clear and significantly higher accuracy of the novel CTRNN variants compared to the basic CTRNN as well as the SRN and GRU baselines. These results were stable over the whole range of investigated meta parameters, including architecture size, shape, and initial timescale setting. For the CTRNN, found the best results for $(16, 8, 4, 2)$ neurons and initial timescales of $(2, 6, 18, 54)$, as compared in Tab. I for the mean absolute error (MAE). In fact, the GCTRNN and GACTRNN seem to

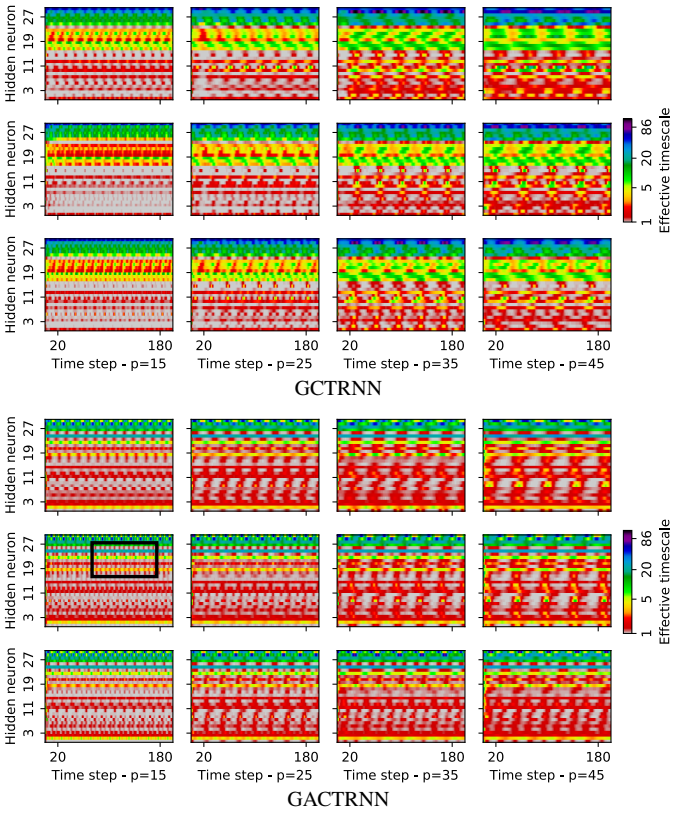

Fig. 6. Effective timescales of the hidden layer neurons during predicting Lissajous Curves with different period lengths. Highlighted box shows doubled frequency of timescale oscillations for the 8-shape (best viewed zoomed-in).

solve the task nearly perfectly, where the baseline CTRNN predicts with small deviations (see Fig. 5). Interestingly, the predictions are mostly off in the beginning, which indicates that the difficulty is related to the vanishing gradients and predicting from the same starting point.

In analysing the effective timescales for the gated CTRNNs, we observed a peculiar behaviour that shows strong differences between processing the different Lissajous Curves. As shown in a representative example in Fig. 6, we can see that the neuron's effective timescales adapt to both the type of the input shape as well as its period length p. This is indicated by the timescale differences emerging from different curve shapes (rows) and the frequency of the timescale oscillations decreasing with increasing period lengths (columns). Additionally, the timescale of many neurons alternate over time frames corresponding to the periods of the respective curves. We can also see that in the 8-shaped curves some neurons alternate the timescales with about half the period, which is not visible for the O-shaped or V-shaped curves. Structurally, the GCTRNN's neurons cluster into four different timescale regions (indicated by the purple, blue, yellow, and red), while the GACTRNN seems to develop a more specialised distribution of the middle region, leading to a stronger manifestation of both small and large timescales. This can be explained by the timescales having a certain limit in their range, leaving the GCTRNN to utilize the fixed initial timescale values a bit differently.

Overall, the observed timescale patterns indicate a tendency of the gated networks to adapt to the specific timescale patterns in the data. Firstly, adaptation occurs in terms of conserving information about reoccurring patterns (as seen by the timescale oscillation frequencies decreasing with increasing period lengths). Secondly, only a few neurons change the timescales slowly while a large number of neurons show similar patterns of leaking strongly or conserving activation to some extend (notably in the red and gray regions). This might indicate that the ability to cover features on different timescales can emerge dynamically.

C. Fluctuating Human Motion Pattern

To study how the xCTRNNs can capture temporally fluctuating patterns that seem random and chaotic but have, in fact, an underlying structure, we used the hand drawing data, recorded by Ahmadi and Tani [20]. In this task, humans were asked to continuously draw eight-shape figures on a tablet by concatenating three distinct prototypical patterns of shape drawings. This way the data includes reoccurring long-term patterns, disturbed by individually-different short-term fluctuations in a random sequence. Our particular aim is to inspect how neurons activate when being set to specific low or high timescales while solving the sequence prediction task with latent temporal fluctuations. In order to inspect the generalisation capabilities of the xCTRNNs, we used the 16 sequences of length 400 for training and tested on prepared independent sequences in an open loop after time step 200. The best CTRNN was identified with $(32, 16, 8, 4)$ module sizes and timescales of $(1, 5, 25, 125)$. Since the eight-shapes were drawn in 2D, the motion patterns appear as pseudo-sinusoidal sequences over two neurons. Overall, the CTRNN variants can roughly capture and predict the motion patterns, but over time clearly lose synchronicity (see Fig. 7 for a comparison). This was observed even more clearly for the SRN and the GRU (not shown), which tend to activate in oscillations that roughly resemble the rhythm of the prototypical patterns with the largest period in the test sequences. Among the xCTRNN, all variants capture both the overall structure of reoccurring block motion patterns and the specific pattern and their fluctuations reasonably well, but show notable deviations, which increase over the course of the prediction.

In analysing the sequence prediction as well as the effective timescales during prediction, we hypothesise that both the principle rhythm of concatenating prototypical patterns and the short-term fluctuations are captured differently by the xCTRNNs. We can, in fact, observe that activation pattern are distinctly different when the timescales can dynamically change during prediction (see Fig. 7 for a comparison). For the ACTRNN, where the timescales are individually different but constant during activation, many neurons activate strongly during the prototypical patterns. Here, the activation graph indicates that few neurons with very high timescales modulate the faster-changing neurons, where many neurons redundantly learn to activate for the patterns. In the GACTNN, only few hidden neurons activate strongly, while others only contribute

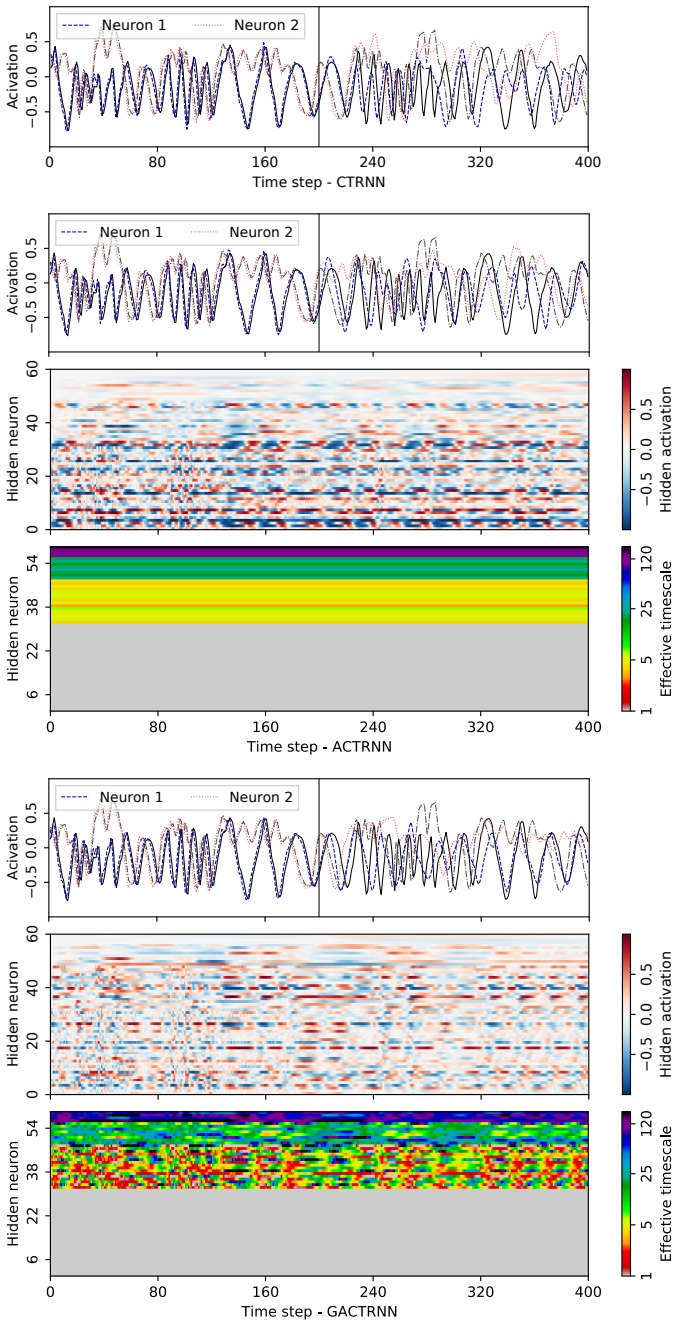

Fig. 7. Predicting a human motion sequence from the test set compared to the simultaneous hidden layer neurons' activation and effective timescales.

weakly (Fig. 7, second last plot). In parallel, correlating with the prediction output, the effective timescales change strongly, where some neurons with larger timescales change to even higher values and thus update activation slower, while many neurons with smaller timescales change to leak even stronger. In effect, the gating CTRNNs seem to learn to synchronise timescales in faster-updating and slowly-updating neuron groups, tied to the output prediction. Overall, some groups seem to specialise in synchronising to the overall motion patterns and some to specific motion curves within a pattern, where fewer neurons are needed to represent patterns.

IV. Discussion

In this paper, we propose to extend[1] the classic CTRNN architecture so that timescales can adapt to a specific timescale distribution (as in [16], [21]) and can change over the course of the neurons' activations by gating input from other presynaptic neurons. This extension seems to allow the architecture to adapt the timescale parameters towards the timescales of temporal dynamics in the data. For instance, a neuron with high timescale (thus low leakage) can get modulated by presynaptic neurons to switch its mode between quickly forgetting and even more strongly maintaining a certain activation.

In the exploratory analysis, we found that the adaptation of timescale biases leads to intrinsic timescales that are more fine-grained and suited for the data. The timescale gating changes the behaviour of the neurons during activation in a way that it adapts to systematically changing timescale characteristics in the data as well as to uncertain fluctuations.

Although we just started analysing the computational properties of the GACTRNN, this network architecture seems promising for capturing characteristics of time series with different and spontaneously varying timescales. More work is needed to fully investigate how the gating and adaptation of the timescales change and specialise to the temporal characteristics of the data. For this, it is necessary to study - and perhaps first of all collect - larger data sets that show distinct and highly complex multi-timescale dependencies. Music is a good candidate, while behavioural data is another.

A. Neurocognitive Modelling

Behavioural data and data from neuro-imagining studies are of particular interest as it reflects the complexities that the human brain is tasked to deal with [10], [11]. In particular, perception data from natural human-environment interactions as well as EEG data, which could be seen as a pre-processed form of behavioural data, is intriguing for two reasons. First, the main purpose of the GACTRNN architectures is to study and better understand candidates of computational mechanisms that are key in the brain's information processing [22]. Second, the architectures can be employed in neurocognitive models that aim to explain the behavioural effects of differences in temporal information processing [23]. For example, cognitive models can be developed to better understand how processing on different intrinsic timescales contribute to difficulties in processing and predicting world models and conforming behaviour. This is particularly interesting in-between typically developed people and people with psychiatric symptoms, including autism spectrum conditions or schizophrenia [12].

B. Conclusion

Overall, adaptive and gating timescales mechanisms show potential as candidates for modelling modulation between neurons. In intrinsically capturing the temporal characteristics in the data, they seem effective for tasks of sequence learning as well as tasks of cognitive modelling with perception or action on long- and short-term dependencies.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] M. Chen, J. Pennington, and S. S. Schoenholz, "Dynamical isometry and a mean field theory of RNNs: Gating enables signal propagation in recurrent neural networks," in *Proc. of the 35th International Conference on Machine Learning (ICML)*, Stockholm, SE, 2018, pp. 873–882.

[3] A. Gu, C. Gulcehre, T. Le Paine, M. Hoffman, and R. Pascanu, "Improving the gating mechanism of recurrent neural networks," *arXiv preprint*, 2019, arXiv:1910.09890.

[4] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, "Time2Vec: Learning a vector representation of time," *arXiv preprint*, 2019, arXiv:1907.05321.

[5] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment," *PLOS Computational Biology*, vol. 4, no. 11, p. e1000220, 2008.

[6] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," in *Proc. International Conference on Learning Representations (ICLR)*, Toulon, FR, 2017.

[7] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, 2004.

[8] A. K. Engel, C. Gerloff, C. C. Hilgetag, and G. Nolte, "Intrinsic coupling modes: multiscale interactions in ongoing brain activity," *Neuron*, vol. 80, no. 4, pp. 867–886, 2013.

[9] K. D. Himberger, H.-Y. Chien, and C. J. Honey, "Principles of temporal processing across the cortical hierarchy," *Neuroscience*, vol. 389, pp. 161–174, 2018.

[10] J. Tani, *Exploring robotic minds: actions, symbols, and consciousness as self-organizing dynamic phenomena*. Oxford University Press, 2016.

[11] A. Cangelosi and M. Schlesinger, *Developmental robotics: From babies to robots*. Cambridge, US: The MIT Press, 2015.

[12] P. Lanillos, D. Oliva, A. Philippsen, Y. Yamashita, Y. Nagai, and G. Cheng, "A review on neural network models of schizophrenia and autism spectrum disorder," *Neural Networks*, vol. 122, pp. 338–363, 2019.

[13] S. Heinrich and S. Wermter, "Interactive natural language acquisition in a multi-modal recurrent neural architecture," *Connection Science*, vol. 30, no. 1, pp. 99–133, 2018.

[14] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986.

[15] K. Doya and S. Yoshizawa, "Adaptive neural oscillator using continuous-time back-propagation learning," *Neural Networks*, vol. 2, no. 5, pp. 375–385, 1989.

[16] S. Heinrich, T. Alpay, and S. Wermter, "Adaptive and variational continuous time recurrent neural networks," in *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Tokyo, JP, 2018, pp. 13–18.

[17] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork RNN," in *Proc. of the 31st International Conference on Machine Learning (ICML)*, Beijing, CN, 2014, pp. 1863–1871.

[18] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN Encoder-Decoder for statistical machine translation," *arXiv preprint*, 2014, arXiv:1406.1078.

[19] S. Murata, J. Namikawa, H. Arie, S. Sugano, and J. Tani, "Learning to reproduce fluctuating time series by inferring their time-dependent stochastic properties: Application in robot learning via tutoring," *IEEE Trans. Auton. Ment. Dev.*, vol. 5, no. 4, pp. 298–310, 2013.

[20] A. Ahmadi and J. Tani, "Bridging the gap between probabilistic and deterministic models: A simulation study on a variational bayes predictive coding recurrent neural network model," in *International Conference on Neural Information Processing (ICONIP 2017)*, Guangzhou, CN, 2017, pp. 760–769.

[21] S. C. Quax, M. D'Asaro, and M. A. J. van Gerven, "Adaptive time scales in recurrent neural networks," *Science Reports*, vol. 10, no. 11360, 2020.

[22] H.-Y. S. Chien and C. J. Honey, "Constructing and forgetting temporal context in the human cerebral cortex," *Neuron*, vol. 106, no. 4, pp. 675–686, 2020.

[23] T. Watanabe, G. Rees, and N. Masuda, "Atypical intrinsic neural timescale in autism," *Elife*, vol. 8, p. e42256, 2019.

[1] Reference implementations of the extended CTRNNs are available on GitHub: https://github.com/heinrichst/GACTRNN.