University of Groningen

Dimensions of "Timescales" in Neuromorphic Computing Systems

Jaeger, Herbert; Doorakkers, Dirk; Lawrence, Celestine; Indiveri, Giacomo

Dimensions of "Timescales" in Neuromorphic Computing Systems

Herbert Jaeger
University of Groningen
h.jaeger@rug.nl

Celestine Lawrence
University of Groningen
c.p.lawrence@rug.nl

Dirk Doorakkers
University of Groningen
h.a.doorakkers@rug.nl

Giacomo Indiveri
ETH Zurich and University of Zurich
giacomo@ini.uzh.ch

February 10, 2021

Abstract

This article is a public deliverable of the EU project *Memory technologies with multi-scale time constants for neuromorphicarchitectures* (MeMScales, memscales.eu/, Call ICT-06-2019 Unconventional Nanoelectronics, project number 871371). This arXiv version is a verbatim copy of the deliverable report, with administrative information stripped. It collects a wide and varied assortment of phenomena, models, research themes and algorithmic techniques that are connected with timescale phenomena in the fields of computational neuroscience, mathematics, machine learning and computer science, with a bias toward aspects that are relevant for neuromorphic engineering. It turns out that this theme is very rich indeed and spreads out in many directions which defy a unified treatment. We collected several dozens of sub-themes, each of which has been investigated in specialized settings (in the neurosciences, mathematics, computer science and machine learning) and has been documented in its own body of literature. The more we dived into this diversity, the more it became clear that our first effort to compose a survey must remain sketchy and partial. We conclude with a list of insights distilled from this survey which give general guidelines for the design of future neuromorphic systems.

Contents

1   Introduction and overview

The original topic for this deliverable, agreed more than a year ago and specified in the project proposal, was *Report on literature survey and analysis of STDP and RC* [= spike-time dependent plasticity and reservoir computing, respectively] *guidance for the design of indeterminate hardware,* with the understanding that the guiding focus of this survey would lie on timescale aspects. When we started to carry out this survey we perceived that, first, the restriction to STDP and RC would exclude many technically and algorithmically important phenomena in general analog (spiking) microchips — so we extended our perspective to neuromorphic computing in general. Second, we realized that the phenomenology of "timescales" is very rich, and this word is applied to quite different phenomena in different contexts. Therefore, an important contribution of this deliverable report is to stake out the conceptual dimensions of this scintillating word. Hence our new title, *Dimensions of "timescales" in neuromorphic computing systems.*

The report is structured as follows. In Section 2 we unfold the conceptual dimensions of the timescales concept, by pointing out different uses and subconcepts of this notion in different formal-theoretical, computational and physical contexts. In the following three sections we compile the findings of a literature survey, sorted into the fields of neuroscience (Section 3), mathematics and theoretical physics (Section 4), and computer science / machine learning (Section 5). Section 6 distils a number of take-home messages distilled from the findings in this deliverable which we hope are helpful for informing future research in MemScales and beyond. A final Section 7 gives concrete physical-timescale related guidelines for the design of indeterminate hardware which result from the specific givens in recent developments in STDP and RC research.

2   Talking and thinking about timescales: dimensions of a very rich concept

In this section we give a "travel guide" for the landscape of timescale phenomena, and point out terminologies that are used. We found it not possible (at least, not at present) to develop a unified, comprehensive conceptual framework. Therefore, we present our findings in the form of a collection of objects and places of interest, as in a tourist guide where showplaces are paid passing visits.

A "Speed" and "memory". There are at least two different, but likewise fundamental, understandings of "timescales".

The first one is to speak of fast or slow timescales when a dynamical system evolves faster or slower, as one could for instance mathematically determine by changing *time constants* in ordinary differential equations (ODEs). When the system is "fast", its rates of change in numerical dynamical variables are high — timeseries will exhibit many high absolute first derivatives, have strong components in the high-frequency end of its Fourier spectrum, etc. We note, however, that there is no unique mathematical criterion to measure "speed". For instance, if an ODE-defined dynamical system (DS) is close or in a fixed point attractor, even very small ODE time constants will not translate to large numerical change rates.

The second is to speak of long or short *memory* durations. In this view, a DS is evolving on a slow timescale when it has long "memory spans". This means that some information that is *encoded* in the system state at some time t can be decoded again at (much) later times. Instead of using the word "memory", which is too closely suggestive of neural and cognitive processing, we find it preferable to speak of "preservation of information across time". Exploring the preservation of information across time has been one of the main themes in the theoretical literature on reservoir computing (RC) in the last 20 years.

B State variables, control parameters. In mathematical models of DS, it is customary to distinguish dynamical *state variables* from *control variables*. Both appear as arguments in the defining function of iterated maps and differential equations (and other formalisms), as in the generic ODE $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{a})$ where $\mathbf{x}$ is the state vector and $\mathbf{a}$ denotes the vector of control parameters. The idea is that the latter are "fixed" or "given" and are not affected by the system state update operators. However, this role distribution dynamical vs. fixed is not always clear-cut. Often one considers scenarios where the control parameters are subject to slow changes, for instance induced by top-down regulatory input in hierarchical neural processing architectures.

C Absolute and relative timescales. Some systems, formal or physical, can be "run" faster or slower and the speed is the only thing that changes. Examples are systems defined by differential equations, whose "velocity" can be set by time constants; or digital microprocessor systems whose clock cycle duration can be varied. To describe this "velocity" one needs an absolute reference time which can be formal (as in ODE systems) or physically "real", as in physical microprocessors. Absolute timescales are important for computing systems that are interacting with their input/output environment "in realtime".

Many of the fascinating properties of complex dynamical systems arise not from its absolute "velocity" but from the fact subsets of dynamical variables, or subsystems, evolve faster or slower than other subsystems. Temporal multiscale properties can also be attributed to the dynamics of a single variable: In a single-variable timeseries one may identify a spectrum of short- and longrange "correlations", or "memory traces", or statistical dependencies, etc. *Multiscale dynamics* are a general and maybe essential characteristic of *complex* systems, although this concept has no single, commonly agreed definition.

It is remarkable that there seems to be no good word for the "velocity" of a dynamical system, which is why we put this word in quotes. In speaking about "velocity", one says "the system is slow" or "the fast subsystem", but no-one says "the speed (or velocity) of the system is high".

D Measuring time. For a physicist or signal processing engineer, time is given by nature and invariably denoted by t. Mathematicians do not care about real time and speak of "unit time steps" or "the unit time interval", an arbitrary convention to associate the unit inverval on the real line as a reference to quantify time. When a mathematician talks with a signal processing engineer, the former tends to be puzzled (if not disturbed) by the fact that the latter keeps talking about "seconds", a word that one will not find in mathematical textbooks on dynamical systems.

Theoretical computer scientists ignore the aspect of temporal duration entirely. The formal models of computing automata only know of "state update steps", where the only aspect that is left over from physical time t or unit time $[0, 1]$ is *serial order* and causation: the next state comes after the previous and the latter determines the former. A most interesting challenge with regards to measuring time arises for computational neuroscientists when they want to explain how a brain can "estimate" or "experience" time. What mechanism in neural dynamics can enable a subject to estimate the presentation duration of a stimulus? Proposed answers include the use of neural delay lines, neural reference oscillators that function as clocks, or stimulus-duration-characteristic patterns in high-dimensional neural transients. We find that a general theoretical treatment of how "clocks" or "time-meters" can be defined in dynamical systems would be a rewarding subject of study.

E Collective and derived variables. In statistical physics, neural field models, population dynamics and many other domains where one investigates systems made from large numbers of interacting small subsystems or "particles", one often describes the global dynamics of the "population" or "ensemble" through derived collective variables. Their timescale is typically *slower* than the native, local timescales of the interacting subsystems; and one typically tries to capture the global dynamics with a *small number* of such collective variables. Slowness is here connected with *dimension reduction*, *simplification* or *abstraction*.

F Sub- and supersampling. In discrete-time models of dynamical systems one can create "speedups" by subsampling and "slowdowns" by supersampling / interpolation. This however makes sense only for discrete-time models that can be understood as sampled versions of a continuous-time process. It makes no sense to supersample, for instance, the state sequence of a Turing machine.

G Slowing-down by discretization. When a real-valued timeseries is discretized by binning, or a fine-grained discrete-valued timeseries is further simplified by coarsening, high-frequency detail (which can be regarded as fast-timescale information) gets lost in cases where there are oscillatory fluctuations within bins in the original time-series. Thus, discretization or binning procedures may cut the spectrum of effective timescales.

H Frequency filtering. Applying frequency filters to trajectories deletes dynamical components on the timescales corresponding to the cancelled frequencies. In the special case of low-pass (smoothing) filters, fast timescale information is lost. In the special case of high-pass (baseline normalization) filters the opposite effect is achieved.

I What is a "moment"? We are used to think of a timeline as an ordered sequence of *time points* - let us call them "moments". Even when one leaves out the complications of relativity theory, seeing time as a succession of zero-time moments is not always the most helpful view. Cognitive neuroscientists tell us that the subjective experience of "now" in some ways integrates over several milliseconds. When neuroscientists try to detect or define "synchrony" in neural spike patterns, they must soften the mathematical notion of point-sharp co-temporality to short intervals. Signal processing

engineers would sometimes like to get rid of delays in their equations but can't. Abstractly speaking, in high-dimensional dynamical systems with nonzero-length signal travel pathways, relevant information-carrying "patterns" arise not instantaneously but need some minimial duration to realize themselves. Such observations suggest that in hierachically structured complex dynamical systems, a hierarchy of "nowness-windows" might be an appropriate concept, with short-duration "moments" defined for small, local subsystems and increasingly longer-duration "moments" as one goes up in the subsystem hierarchy.

J Homeostasis, stability, robustness. Biological organisms, brains, non-digital microchips made from unconventional materials, and many other computing or cognizing systems must preserve their functionality in the presence of external perturbations, change of environment, aging, parameter drift and other challenges. They do so through a wide spectrum of stabilization mechanisms which exploit, for example, redundancies, attractor-like phenomena, stabilizing feedback control, adaptation and learning, or robust network topologies for system architectures. A common denominator in this diversity of mechanisms is that they aim to ensure that vital system variables stay within a (narrow) viability window, often by attempting to stabilize them close to an optimal value. This has a twofold aspect of slowness. First, change rates of variables that are being stabilized are slow (when the stabilization is successful). Second, these critical variables must be stabilized through long timespans — vital variables through the entire system lifetime.

K Nonstationarity and mode hierarchies. Computing systems exhibit nonstationary dynamics, be it because they are input-driven or because they "learn" or because they execute a sequence of subprograms. System trajectories (timeseries) resulting from nonstationary dynamics can be qualitatively or quantitatively described through temporal hierarchies of *dynamical modes*. For instance, a neuronal spike train can be characterized on a very short timescale by an interspike interval, on a short timescale by burst modes, on a longer timescale by locally averaged firing frequency, and on a very long timescale by asymptotic measures.

In information processing systems, one may find ways to characterize what the current mode "represents". For instance, the neural activity trajectories in a speech-processing brain might be described as "encoding" or "representing" linguistic phonemes, syllables, words, phrases, sentences, texts.

There exists no unique, general mathematical characterization of modes. Modes of an evolving DS might, for instance, be characterized in terms of frequency spectra, signal shapes, signal energy, attractor structures, degrees of chaoticity, or regions of the system's state space, to name but a few. Describing temporal multiscale dynamics is very much the same task as characterizing modes, and there seems to be an unlimited repertoire of options.

L Hierarchical architectures. The human brain, most autonomous robot control systems, and many multiscale signal processing and control systems are hierarchically structured. The "bottom" layers are in direct contact with incoming signals and generate output signals, while "higher" processing layers carry out increasingly

"cognitive" tasks based on increasingly abstracted and condensed representations of the information contained in the input signals.

We find it a wide-spread, even paradigmatic view that higher levels operate on slower timescales than lower levels. This view is supported by evidence from biological brains, and guides the design of artificial signal processing and control systems that have to cope with temporal multiscale data. It also agrees with the view of classical AI, where *action planning* architectures generate goals and subgoals, plans and sub-plans, procedures and subprocedures in a nested way, where higher nesting levels are taken care of by higher processing layers.

A formidable challenge arises for formal modelers and concrete system developers (in computational neuroscience, machine learning and robotics). It concerns the nature of "top-down" influences: in what sense, and by which concrete mechanisms, do higher layers influence the processing on lower layers? Should this influencing be understood and realized as attention, prediction, context setting, or modulation? Many questions, both conceptual and algorithmic/mathematical, still are open.

M Characterizing multiscale dynamics from left to right and from the side.
In symbolic dynamics and theoretical computer science, a theme related to multi-timescale dynamics is infinite-length symbol sequences. They can be characterized by automata models, where some type of automaton generates the sequence "from left to right" — that is, the sequence is seen as the trajectory of a dynamical system. But such sequences are also described and analyzed as being the fixed points of applying grammar rules. This method of characterizing the structure of an infinite sequence is a-temporal but directly yields a transparent account of its multiscale, hierarchical structure. Research to connect these two views has only recently started. It seems likely (even obvious) that multiscale properties of DS trajectories are related to memory mechanisms that are effective in the generating DS.

In theoretical modeling of timeseries data (in theoretical physics and economics in particular), stochastic dynamics with *long memories* are discussed in terms of the shape of the corresponding power spectrum. One speaks of *fat* or *heavy tailed* or $1/f$ power laws. Such long-memory behavior is associated with *self-induced criticality* or *edge of chaos* conditions, and is often claimed as a characteristic of complex natural processes, for instance in economics, neural dynamics, or speech.

Theoretical computer science offers a canonical repertoire of methods to specify automata with increasing memory capacities (from finite-state autonomata through a variety of stack automata to Turing machines), and how they relate to an equivalent grammar. It would be interesting to investigate how such memory mechanism hierarchies of symbolic automata and their grammars can be transfered into the domain of continuous-time, continuous-value DS and the power spectrum proporties of their trajectories.

N Time warping. In real-life timeseries one frequently finds local speed-ups or slow-downs, for example a speaker stretching out the pronounciation of a vowel for empha-sis. A related effect occurs when different realizations of a signal are originating from slower or faster generators, for example from slower or faster speakers. Biological brains can, within limits, compensate for such *time warping* in inputs. For artificial

temporal pattern recognition systems — often recurrent neural networks (RNNs) — this poses serious challenges.

O Online and real-time processing. Many applications of signal processing and control systems must generate output responses to incoming data streams without or only minimal delay. This is the generic case for control systems, but also for many other applications, for instance speech-to-speech translation, medical cardiographic monitoring. One speaks of *online* or *real-time* processing. One may make a fine distinction (not always observed) between these two concepts.

In online processing, the signal processing system is "entrained" to the driving input stream. Its internal states directly "synchronize" with the input, where "synchronizing" is understood in a generalized way that includes nonlinear transformations and memory effects. The processing dynamical system can appropriately be mathematically regarded as a dynamical system. Analog signal processing devices and RNNs are prototypical examples.

In real-time processing — a natively digital-computing notion — the algorithmics of the system-internal processing is decoupled from the input. The input signal stream is sampled and buffered, processing subtasks are identified and solved by algorithms which must run fast enough to deliver results within predefined time limits. On universal computers this may require the use of an underlying real-time operating system.

P Time complexity classes. In theoretical CS, input-output tasks that can be algorithmically solved (i.e., *computable* tasks) are ordered into a hierarchy of *time complexity*. In theoretical CS, input arguments are always formatted as finite-length symbol strings ("words"). The runtime of an algorithm is measured by the number of machine update steps (concretely, clock cycles) needed from the presentation of the input word until the output word has been generated. To define a complexity class, the runtime is related to the length $|w|$ of the input word w. For example, the class P of polynomially computable tasks comprises all tasks for which some algorithm (or deterministic machine) and some polynomial p exist, such that the algorithm terminates within $p(|w|)$ update steps, for all input words w.

We remark that the concept of time complexity is tied to understanding "computing" as "running a Turing machine from presenting an input word until it terminates with an output word". This concept of time complexity cannot be naturally transfered to online processing tasks.

Q Different names for different timescales. Biological brains exhibit dynamical processes on many timescales, and different processes affect different physical elements in brains in different ways. This leads to a entangled maze of dynamical phenomena in which it is hard to not get lost. A coarse orientation is provided by the conceptual sequence *inference* → *adaptation* → *learning* → *development* → *evolution*. These terms denote denote bundles of dynamical phenomena which manifest themselves on increasingly long timescales. None of them has a precise definition, but all of them are used in computational science, cognitive science and neural-networks based machine learning with more or less similar semantic intuitions:

- *Inference* processes refer to the fast operations of sensor processing, motor control and "reasoning" which do not essentially rely on structural or parametric changes of the neural processing system, using the system "as is". In machine learning one often speaks of "inference" when a ready-trained neural network (or other ML model) is used to process task instances for which it has been trained.

- *Adaptation* is a particularly broad and vague concept. A common denominator of its uses seems to be that adaptation works on slower timescales than inference, and is in principle reversible. It often describes processes when a cognitive / neural system re-calibrates, or re-focusses itself when the environmental context of operation changes. In formal models, adaptation processes often are expressed through changes of control parameters in neural subsystems, induced by "top down" regulatory mechanisms or subsystem-inherent homeostatic self-stabilization mechanisms. While this seems to us the most common intuition connected to the word "adaptation", it is also used in a much more generalized way to denote any change of any sort of system (from a single synapse to a biological population in an ecological niche) that improves the system's "performance" or "viability". In those cases, adaptation is not usually reversible.

- *Learning* refers to processes which expand the functionality of a cognitive system on the basis of experience. Learning processes are usually considered irreversible ("forgetting" are processes in their own right which cannot be understood as time-reversed learning). Learning processes are commonly associated with irreversible changes in system parameters — in neural networks typically "synaptic weights". Structural changes (like deletion of neural connections or adding neurons to a network) may also result from learning, though this aspect seems less central to the "learning" concept than mere parametric change.

- *Development* is a notion which is much more common in the cognitive and neurosciences than in machine learning. It refers to the life-long history of an individual, autonomous cognitive system (animal, human, or generalized "agent"). The development history is often segmented into life periods like pre-natal development, stages of infancy, youth, adolescence, old age which are in turn associated with specific structure-changing processes in the agent's brain. We foresee that developmental change will also become an important theme in neuromorphic computing systems based on non-digital hardware which cannot be "programmed" and whose physical substrate is subject to aging.

- *Evolution* is the longest-timescale item in our list of process categories. It describes the adaptive change of entire populations, across generations, to fit a (possibly changing) environmental "niche".

Mathematical models of cognitive systems describe inference and adaptations processes (typically) through changes in the values of system variables (dynamical state variables and/or control parameters). The system equations do not structurally change. In contrast, models of development and evolutionary processes must account for structural changes in the system equations. Formal tools for *effecting* and *simulating* structural change in system equations exist in the form of genetic / evolutionary algorithms. However, mathematical theories that can be used to *characterize*

and *analyse* structural change *in qualitative terms* are scarce, heuristic, and generally still under-developed. We find that certain tools in mathematical logic ("nonmonotonic logic") come closest. However, these formalisms are not connected yet to dynamical systems modeling.

R Philosophy of time. Time is a fundamental quality of human experience, and philosophical inquiries have approached this theme from many angles. This lies outside our competences and we only list some of the aspects of time that have been investigated by philosophers (gleaned from Callender (2011)).

- *Time and metaphysics.* What are the ontological realities ("presentalist", "possibilist", "eternalist") of the past, the present, and the future? Is time continuous or discrete?

- *The direction of time.* What is the difference between the past and the future? Is the arrow of time inherent in time, an effect of causality, or of thermodynamical laws?

- *Time, ethics and experience.* Themes include: the subjective "now"; memory, anticipation, decisions and free will; development of time concepts in children; benefit and harm in the past and the future.

- *Time in physics.* Are (which) physical laws time reversible? How is time related to space? How is time understood in relativity theory and quantum theory? What are clocks? How is time affected by the uncertainty principle? Is there time at all?

This listing of aspects of time's ways of flowing faster or slower, or of our ways to observe a system for shorter or longer durations, makes it clear that a systematic, unified account of "timescales" is out of reach. In order to give instructive initial input to the MemScales project, the best we can currently do is to compile a "tourist guide"-like collection of concrete empirical findings, mathematical models and theories, and machine learning approaches which have a bearing on some of the listed dimensions of "timescales". We coarsely sort these collection items into three sections **Brains**, **Mathematics** and **Computing**, which is somewhat arbitrary since many lines of research cross-connect these areas. Our survey will be all but complete: firstly because we largely omit entire domains of science (in particular biology, physics, psychology and philosophy), and secondly because even in the three domains that we did explore (the wider neurosciences, mathematics and computer science / machine learning), our bounded expertise and the breadth of the subject put limits on what we could effectively cover. At the end of each section we list themes that we know should be included in future extensions of such a survey.

3   Brains

This section collects perspectives of research, empirical findings and models from the wider neurosciences including (some of) cognitive science.

The title of the first four subsections are the consequence of frameworking towards a theory of neuromorphic signal processing, which we hope to work out more fully and

more systematically in our future work in Mem-Scales. A metaphor to motivate our strategy is that it is incredibly difficult to solve a Rubik's cube by just focusing on one side at a time. In that analogy, it is even undefined which parts of the cube correspond to an open problem. For in some definitions, there already are reasonable solutions to so-called open problems like the binding problem (Skarda, 1999), stability-plasticity dilemma (Shouval et al., 2002), and systems memory consolidation (Van Kesteren et al., 2012). But these solutions do not bind together to a comprehensive theory. The organization of material in our four subsections below arises from distinguishing two axes of discussing neural timescales, an individual (neuron) — recurrent axis and a dynamical processing — plastic adaption axis.

3.1 Single-neuron processing

A biological neuron has multiple timescales of phenomena due to voltage-gated (Doyle et al., 1998) and ligand-gated (Katz, 1971) ion channels (Ranjan et al., 2011), spatiotemporal filtering across dendritic cables (Rall, 2009), hierarchical synaptic-dendritic-membrane-somatic processing (Gao et al., 2018) and biochemical pathways involving multiple chemical compounds (Bray, 1995; Barkai and Leibler, 1997; Bargmann, 2006). Thus, a single neuron has enormous capacity for signal processing, much better than the McCulloch-Pitts and LSTM units in presently widespread artificial neural networks.

A special primitive for spatiotemporal processing at the dendritic level is coincidence detection. It can explain concentration-invariant signal recognition, for example in olfactory (Hopfield, 1995) networks. Chaining of multi-timescale transient units with a coincidence detector results in transient synchrony (Hopfield and Brody, 2001) and can explain uniform time-warping invariant signal recognition.

A pioneering model of temporal processing at the membrane level is due to Hodgkin and Huxley (Hodgkin and Huxley, 1952), which considers the membrane potential and ion channel activation numbers as a coupled system of nonlinear ODEs. A generalization of the Hodgkin-Huxley model with multiple ion channels whose conductances are nonlinear and modulatable at multiple timescales is now the gold standard for modelling the membrane dynamics of a neuron. Note that if the ion channel activation numbers do not have any inter-neuronal immediate effects, then just modelling the membrane potential is sufficient for a complete neurodynamical understanding. For example, Izhikevich (2004) showed that a reduced 2-dimensional threshold-reset ODE system is sufficient to explain a possible set of 20 kinds of temporal processing in cortical neurons including tonic spiking, phasic spiking, spike bursting, spike latency, subthreshold oscillations, rebound, bistability, and spike frequency adaptation.

3.2 Recurrent processing

Here we will focus on the workings of recurrent neural network, ignoring plasticity. Functionally, a worm brain can be understood to operate in sensory-inter-command-motor layers (Gray et al., 2005). The human brain is similar but more complicated (Eliasmith et al., 2012), i.e. the command-layer is split into functional regions performing action selection and motor processing, and the inter-layer is split into functional regions performing information encoding, transform calculation, reward evaluation, working memory and information decoding.

Three noteworthy observations arise from the study of recurrent processing with multiple timescales. Firstly, there often exists a *behavioural hierarchy* (Davis, 1979) resulting in a 'singleness of action' where a long timescale state controls shorter timescales. For example, the mating state of stickleback fish activates a stereotypical dance movement (Tinbergen, 1951), honeybees in a communicative state employ a waggle dance routine (Von Frisch, 1967). Secondly, the behavioural hierarchy can be *deep*, as in a reproductive instinct that activates sub-behaviours such as nest-building or defensive fighting. Experiments have shown that a three level hierarchy explains worm locomotion both behaviorally and in neuroanatomy (Kaplan et al., 2020). Lastly, there need not always be an equivalence between anatomical and behavioral hierarchy, for example chains of neurons can generate birdsongs (Long et al., 2010).

3.3 Neuron-neuron plasticity

Here we will consider the form of synaptic plasticity as postulated by Hebb (Hebb, 1949), where any change in the synaptic weight from one neuron to another neuron, is only based on signals due to the activity of the two neurons i.e. deterministic bi-terminal interactions. Networks with Hebbian plasticity, with or without memory of the neuronal activity (an extreme case is a strict "spike-time" dependence (Caporale and Dan, 2008)), are theoretically capable of signal processing primitives such as principal component analysis (Oja, 1982), self-organizing maps (Kohonen, 1982), and independent component analysis (Jutten and Herault, 1991). So, at a network level, Hebbian learning can be much deeper than the popular maxim of "cells that fire together, wire together". Also, Hebbian-like learning is possible within a single cell (Fernando et al., 2009) if they contain motifs of chemical cycles where the concentration of different chemical species (such as in gene regulatory networks or phosphorylation cycles) can mimic the functionality of synaptic weights (slow-varying control parameters) and action potentials (fast-varying state variables).

Among the gamut of possible Hebbian plasticity rules, the most noteworthy is the Bienenstock-Cooper-Munro (BCM) model (Bienenstock et al., 1982) because it has been experimentally justified (Cooper and Bear, 2012). The BCM model has the rate of change of the synaptic weight equal to a fast timescale times the correlations in the neuron-neuron activity times a saliency factor equal to a mean-deviation of the postsynaptic activity, minus a slow timescale times the synaptic weight. Thus, the BCM model has an increased rate of forgetting on introducing uncorrelated noise, can converge in whitened environments by means of higher-order statistics, can learn direction sensitivity without relying on neuroanatomical asymmetry, and can have a single neuron to be both directionally and orientationally sensitive by learning on video stimuli.

Also noteworthy is that a biophysical model of bidirectional synaptic plasticity (Shouval et al., 2002) can be phenomenologically reduced to a voltage-based STDP (Clopath et al., 2010), which under certain input conditions is equivalent to the BCM model. Experimental measurements of STDP on the visual cortex, somatosensory cortex and hippocampus could be fit to the phenomenological model and distinct timescales were extracted.

3.4 Recurrent plasticity

We can look at recurrent plasticity as having two sides : deterministic effects and non-deterministic effects.

All deterministic effects that are beyond bi-terminal interactions can be subsumed under the banner of neo-Hebbian plasticity (Gerstner et al., 2018), including neuromodulated STDP (Frémaux and Gerstner, 2016) due to some form of reward or punishment (leading to the concept of an eligibility trace and three-factor rules) and heterosynaptic plasticity due to some conservation law (Oh et al., 2015) such as the spatial conservation of the total synaptic weight (Von der Malsburg, 1973) or the normalization of the synaptic weights (Hyvärinen and Oja, 1998) to energetically sustainable levels (Walker and Stickgold, 2006; Kandel et al., 2014). Note that neo-Hebbian plasticity combined with a suitable inter-layer (that is capable of generating rewards internally for congruent or novel information) is sufficient for effective systems memory consolidation (Van Kesteren et al., 2012), but of course in reality non-determinism will also play a supplementary role as discussed below.

All non-deterministic effects (including bi-terminal interactions) can be subsumed under the banner of neural Darwinism (Edelman, 1993), also known as neuroevolution (Stanley et al., 2019). It is plasticity that is based on the principle of selection upon variation, and hence is biased towards generating a hierarchical organization. Experimental evidence supports a hierarchical organization in the basal ganglia to generate action sequences (Jin and Costa, 2015), so it can play an important role in learning procedural memory. Of course, at some level genetics can also enforce a hierarchy (Felleman and Van Essen, 1991), but there is a reason to believe that neural Darwinism plays a major role given that hierarchies in the brain themselves are adaptive. For example, even people with cerebellar agenesis learn to walk (Boyd, 2010) and spoken language perception colonizes the visual cortex in blind children (Bedny et al., 2015; Lane et al., 2015). Also, neural Darwinism can work for hard problems like nonlinear blind source separation for which deterministic and global optimization methods like slow-feature analysis end up failing in high dimensions (Wiskott and Sejnowski, 2002).

3.5 Some topics not covered

We list a number of themes that would warrant a closer inspection but for which we lacked time or expertise on this occasion. Surely there is an endless list of such themes. Nevertheless, with deeper thought or moments of serendipity, we should work towards an ideal where newer themes are assimilated or accommodated into older themes (the Piagetian pun is intended (Piaget, 1952)).

- Neural clock circuits and entrainment of neural dynamics to clock signals.

- Variable binding through theta-wave phase synchronization.

- Hierarchies of memory mechanisms — a large research field in the cognitive and neurosciences which would need a separate, extensive treatment. Surveys are given, for example, by Durstewitz et al. (2000) or Fusi and Wang (2016).

- The role of cerebellar processing in timing fine-control.

- Experimental demonstrations of different time constants in cortical processing (Bernacchia et al., 2011).

- Perception of temporal patterns (Large and Palmer, 2002).

- Stages in ontogenetic development.

4 Mathematics

In this section we collect "pure" mathematical themes and formal modeling methods from dynamical sytems theory and some areas of theoretical physics. Topics in formal logic are treated in Section 5.

4.1 Singular perturbation theory

Arguably the most popular mathematical approach to studying multiple timescale dynamics has been via *singular perturbation theory* (SPT) of systems of ODEs. Intuitively, this theory studies perturbations with small parameters where the dynamics cannot straightforwardly be approximated by the limiting case where the parameters vanish (O'Malley, Jr., 1991; Verhulst, 2005). A geometric approach to singular perturbation theory (GSPT) was first set up by the works of A.N. Tikhonov and those of N. Levinson (Vasil'eva and Volosov, 1967; Kaper, 1999), later worked out in more detail by N. Fenichel (Fenichel, 1979). This geometric approach formalizes interpretations of certain singularly perturbed systems as 'slow-fast' systems, where some variables operate on a relatively fast timescale compared to other slower evolving variables. An enormous amount of research has been done on these slow-fast systems, as they are relevant for the mathematical description of many processes in the life sciences. An extensive modern overview of mathematical theory on slow-fast ODE systems is given by Kuehn (2015). In particular, slow-fast systems have become a big research topic in mathematical neuroscience, see for example Rubin and Terman (2002); Izhikevich (2007); Ermentrout and Terman (2010); Pusuluri et al. (2020). Another notable application of SPT is for control science, where the classical text is Kokotovic et al. (1999).

To illustrate the mathematical approach to slow-fast ODEs, consider the two-dimensional system

$$
\begin{aligned}
\tau_1 \frac{dx}{dT} &= f(x, y), \\
\tau_2 \frac{dy}{dT} &= g(x, y),
\end{aligned}
\tag{1}
$$

with f and g some possibly non-linear functions. We assume $\tau_1, \tau_2 > 0$ to represent the intrinsic timescales of respectively the x and y variables. Now define a new parameter $\epsilon = \tau_1/\tau_2$. Then system (1) can be transformed both into

$$
\begin{aligned}
\epsilon \frac{dx}{ds} &= f(x, y), \\
\frac{dy}{ds} &= g(x, y),
\end{aligned}
\tag{2}
$$

and into

$$\frac{dx}{dt} = f(x, y),$$
$$\frac{dy}{dt} = \epsilon g(x, y), \tag{3}$$

via reparameterizations of time $T = s \cdot \tau_2$ and $T = t \cdot \tau_1$ respectively. As long as $\epsilon > 0$, systems (2) and (3) can be considered equivalent. Suppose that $\tau_1 \ll \tau_2$; intuitively this means the x-variable operates on a much faster timescale than the relatively slow y-variable. In that case $0 < \epsilon \ll 1$, and we might consider how systems (2) and (3) for $\epsilon > 0$ behave when approaching the singular limit $\epsilon \to 0$. We may now also call s the slow timescale, and t the fast timescale. For $\epsilon = 0$, it is important to remark that systems (2) and (3) are not equivalent anymore. The case $\epsilon = 0$ for system (2) is also referred to as the *slow subsystem* or *reduced problem*, while the case $\epsilon = 0$ for system (3) may be called the *fast subsystem* or *layer problem*.

Intuitively, for $0 < \epsilon \ll 1$ the layer problem approximately describes the dynamics of the system on a short timescale, where the slow variable y can be approximated by a constant. Therefore, y can be interpreted as a bifurcation parameter of the fast subsytem. The reduced problem describes the dynamics at $\epsilon = 0$ of the slow variable on a one-dimensional manifold C_0, also called the *critical manifold*, which is given by the zeros of f. Observe that C_0 can alternatively be said to be given by the equilibria of the fast subsystem (3). Close to attracting hyperbolic parts of C_0, the slow subsystem approximately describes the dynamics of the system for $0 < \epsilon \ll 1$ on the longer timescale represented by τ_2. This is formalized by *Fenichel's Theorem*, see for example Fenichel (1979); Kaper (1999) or Chapter 3 of Kuehn (2015).

Orbits starting close to an attracting hyperbolic part of C_0 for $0 < \epsilon \ll 1$ can be predicted to stay close to C_0 by Fenichel's Theorem, approximating the flow of the reduced problem, until nearing a point on C_0 where hyperbolicity is lost. This happens at bifurcations, with respect to y, of the fast subsystem. What happens after reaching such a bifurcation point, requires careful analysis of the full system. One might find jumps between attractors of the fast subsystem (stable equilibria in the two-dimensional example under consideration here), as the system converges towards the vicinity of another attracting hyperbolic part of C_0. While this type of behavior often occurs in slow-fast ODE models, immediate jumps between attractors of the fast subsystem cannot be predicted in general from a decomposition of the full system into fast and slow subsystems at $\epsilon = 0$. Indeed, a peculiar type of behavior might occur where the full system for some time approximately follows a non-attracting hyperbolic part of C_0. This phenomenon is known as a *canard*, and for example plays a role in the analysis of spike adding for models of bursting neurons in mathematical neuroscience (Terman, 1991; Linaro et al., 2012).

The theory of slow-fast ODE systems has been extended to multiscale stochastic differential equations (SDEs) incorporating noise, and more generally to the context of random dynamical systems, see Chapter 15 of Kuehn (2015). Theory on slow-fast SDEs has for example been applied to give a rigorous analysis of certain multiscale synaptic plasticity models for neural networks in Galtier and Wainrib (2012, 2013b). Also, slow-fast systems of maps can be studied with similar techniques, and have been applied

to neuron models as well, see for example Mira and Shilnikov (2005) and Ibarz et al. (2011).

4.2 Delay equations

Multiple timescales can also be introduced in differential equations via delays (Yanchuk and Giacomelli, 2017; Ruschel, 2020). The simplest such delay systems are modelled by

$$\tau_L \dot{x}(t) = -x(t) + F(x(t - \tau_D)), \tag{4}$$

where τ_L is the intrinsic time scale of the system, $F(x)$ is a nonlinear function of x and τ_D is the time delay. When τ_D is large compared to τ_L, it is known that these type of systems can exhibit a host of interesting spatio-temporal dynamical phenomena (Yanchuk and Giacomelli, 2017). Intuitively, a comparatively large delay introduces a slow timescale next to the fast intrinsic dynamics of the system. Such delay systems with large delay have recently been shown to be relevant for approaches to reservoir computing with opto-electronic hardware (Hart et al., 2019). These opto-electronic delay systems can be viewed as an alternative method for implementation of high-dimensional neural networks. Space-time representations allow the dynamics of a delay system with low spatial dimension to be interpreted as spatio-temporal dynamics of spatially extended systems. As such, delay systems have recently also been thought of as useful for the study of complex dynamical behavior in large-scale connected networks. Although delay systems were originally thought of as similar to ring networks of identical neurons, Hart et al. (2019) propose that delay systems can be used to implement networks with arbitrary topologies.

By defining $\epsilon = \tau_L/\tau_D$, equation (4) can be rewritten into the singularly perturbed delay equation

$$\epsilon \dot{x}(t) = -x(t) + F(x(t - 1)).$$

Such type of such systems have been studied for example in Chow and Mallet-Paret (1983) and Ivanov and Sharkovsky (1992).

4.3 Some topics not covered

We list a number of themes that would warrant a closer inspection but for which we lacked time or expertise on this occasion:

- Critical slowdown of dynamics close to bifurcations

- Characterizing multiscale structure in infinite symbol sequences via fixed points of grammar rule applications

- Reaction-diffusion systems

- Variables of multi-dimensional iterative maps can be given differing update frequencies. Little (if any) formal mathematical theory seems to exist on this topic.

- Line attractors and their generalizations.

- Statistical physics modeling of collective phenomena and generalized synchronization, slaving principle (Haken, 1983).

16

5 Computing

This section collects topics, techniques and models from computer science — machine learning and artificial neural networks in particular. The ordering of subsections is arbitrary.

5.1 Neuron models with time constants

In neural network (NN) architectures used in machine learning, a variety (but not a very large variety) of formal/algorithmical neuron models is used. The neuron models used in feedforward NNs always have a-temporal state update equations of the kind $x_i = f_i(\sum_j w_{ij} x_j + b)$, where the x_j are the activations of neurons feeding into neuron i and f_i is a (almost always) monotonically growing "activation function". Time becomes a relevant theme only in recurrent neural networks (RNNs). Besides the a-temporal models $x_i = f_i(\sum_j w_{ij} x_j + b)$, which can also be used in discrete-time RNNs (which then mathematically can be regarded as implementing iterated maps), here we find a diversity of neuron models that either are specified by ordinary differential equations (ODEs) — from the simple leaky integrator neuron $c\dot{x}_i = -x_i + f_i(\sum_j w_{ij} x_j + b)$ with time constant c to LSTM units (Hochreiter and Schmidhuber, 1997) to multi-variable circuit equations for use in analog VLSI neuromorphic microchips (Chicca et al., 2014)) — or by discretized versions of such ODE models, typcially using the elementary Euler approximation; or spiking versions which include a discontinuous neural state reset operation. All of these contain time constants. In complex neuron models (often with a biological motivation), different time constants can be set for different variables. For instance, slow synaptic efficiency adaptation rules ("slow" relative to the soma potential dynamics) are crucial for creating dynamical memory traces in liquid state machines (Maass et al., 2002). When these neurons are "executed" in digital simulations, they can be made to "run" faster or slower over a wide range (limited only be numerical stability conditions) compared to each other or to some reference timescale. In analog neuromorphic hardware realizations however, these time constants are fixed by physical givens and changing them is only possible if the chip design allows one to access and "set" the physical correlates of time constants (for instance voltages), and within limited ranges.

5.2 Time and memory in digital computing

It is easy to *simulate* multiscale temporal dynamics of time-discretized ODE models on digital machines — all one has to do is to set different desired time constants in the various variable equations.

It is not always easy to *realize* multiscale dynamics of time-discretized ODE models on digital machines when the computed dynamics must match physical "real-time" in online signal processing and control applications. The system state update equations must be simple enough, or implemented cleverly enough, or parallelized enough, to ensure that the digital processing needed to compute the next time slice state takes at most as much time as the physical time allotted for a sampling interval. This may become demanding in robot control applications, in particular in compliant robots where

the sampling frequency must be high (order of 1000 Hz) in order to react fast enough to sensor signals signifying effector impact.

In serial-data "cognitive level" AI / machine learning tasks like text processing or visual gesture analysis, the processing algorithm must have a working memory which has (at least) the power of a stack memory. This is needed to process the hierarchically nested temporal structure in "grammatically" organized input sequences. Realizing such a stack memory is of course not a problem for digital computers when they execute symbolic-AI programs. When RNNs that have been set up or trained for such tasks are simulated on digital machines, the hierarchical memory organization cannot be directly mapped to the (easily available) physical stack memory mechanisms of the digital computer. Instead, this memory functionality must become encoded and realized in terms of the RNNs dynamics. One way to do so is to train binary context-level switching neurons which can set the (single) RNN into a temporal hierarchy of dynamical processing modes (Pascanu and Jaeger, 2011).

5.3   Time complexity

In theoretical (symbolic-digital) computer science the concept of *time complexity* refers to upper bounds on the maximal number of processing steps needed by a Turing machine to compute its result when it is started on a (any) input word of length n (Jaeger, 2019) This leads to a classification scheme for the "difficulty" of computing problems. For instance, the time complexity class P is the set of all input-output computing tasks such for a task $T \in P$ that there exists some Turing machine $\mathcal{M}$ and some polynomial function $p : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{M}$ always terminates within $p(n)$ update steps. Some of the deepest unsolved problems in theoretical computer science (and indeed, in mathematics) concern such time complexity classes, in particular the famous $P =? NP$ problem (Cook, 2000).

This standard usage of the term "time complexity" is confined to characterizing the computational demands of evaluating *functions* — a Turing machine (and all other, equivalent mathematical definitions of an *algorithm*, of which there are many) incorporates an input-word to output-word mapping. In the context of neuromorphic computing and recurrent neural networks, a dynamical systems interpretation of "computing" seems more adequate than a function-evaluation interpretation. Some models of "computing" have been proposed in theoretical computer science which account for continual online processing of unbounded-length, symbolic input streams, in particular *interactive Turing machines* (van Leeuwen and Wiedermann, 2001) and more recently *stream automata* (Endrullis et al., 2019). In followup of the traditional questions that are considered in classical complexity theory, this research aims at classifying continual input-output stream processing tasks into complexity classes. The adopted perspective on discussing such complexity classes is however still tightly tied to the classical, function-based concepts of time complexity, in that such automata are designed in a way that upon reading a new input symbol, they can "detach" from the input stream, do a possibly highly complex computation in traditional Turing machine fashion, and after this computation terminates, produce an output symbol (or not). Complexity class hierarchies investigated in such research typically include classes of continual serial input-output tasks which are inaccessible by physical machines — *super-Turing* tasks — in that *oracles* are invoked, that is, external additional input (outside the input data stream) is allowed which provides information that itself is not Turing-computable.

There is also a body of research which aims at transfering concepts and methods from symbolic complexity theory to RNNs with real-valued activations and/or weights (Siegelmann and Sontag, 1994; Sima and Orponen, 2003). One common theme and finding in this line is that a single infinite-precision real numbers allows one to encode an infinite amount of information, which gives rise to super-Turing computing powers (that is, symbolic input-output functions can be computed which no Turing machine can compute). However, there is no evidence that super-Turing performance cannot be physically realized due to noise and limited precision (observability) of physical state variables.

5.4 Logic formalisms for capturing time

A traditional topic in classical (symbolic) AI is "reasoning about action and change", or "reasoning about action and time" (with several conferences and workshops and a wealth of publications that have these expressions in their titles). The objective of this research is to extend the expressive powers of logic-based knowledge representation and inference formalisms to facilitate the representation of, and formal reasoning about, a cluster of themes that includes action, change, planning, intentions, time, events, causation and more. Such formalisms are algorithmically processed with so-called *theorem provers* (also called *inference engines*). These are heuristic, discrete combinatorial search algorithms whose processing steps are not interpreted as temporal steps but as logical arguments. Time, timing, measuring time, comparing durations, ordering events on a timeline etc. are objects that are logically reasoned *about*, in reasoning steps whose ordering is conceived as logical, not temporal. More than four decades of research have produced a rich body of representation formalisms. We can only pinpoint a few examples. Allen (1991) is an early survey. Fundamental figures of reasoning about time are captured by modal operators in *temporal logics* (also known as *tense logics*) (Garson, 2014). A related classical subfield of AI, *qualitative physics* (Forbus, 1988) (closely related: *naive physics*, *qualitative reasoning*) explores logic-based formalisms which capture the everyday reasoning of humans about their mesoscale physical environment. A rather recent development is hybrid logic / dynamical-systems formalisms to reason about physical dynamical sytems (Geuvers et al., 2010) in ways that capture the measurement metrics of "real" continuous time. Such formalisms are intended for formal verification of hybrid physical-computational systems in systems engineering.

5.5 Slow feature analysis

Slow feature analyis (SFA), developed by L. Wiskott (Wiskott and Sejnowski, 2002; Franzius et al., 2008), is a method to extract features from timeseries data (in particular video streams) which are defined by the fact that they change slowly. SFA has been used to explain the functioning of feature detection cells in visual cortex (Berkes and Wiskott, 2003) and hippocampal place cells (Schönfeld and Wiskott, 2015). Interestingly (and possibly, limitingly), the slow features found by SFA are functions of single input frames, not — as one might expect — functions of input episodes that last nonzero time.

5.6 Speed control in RNNs

Biological brains can generate and recognize instances of output patterns which differ from each other only in speed (for instance, generating or recognizing slow and fast handwaving or dance or music pattern generation). For a mathematical model of an RNN written in ODEs with time constants, it would be straightforward to adjust the processing speed in generation or recognition by scaling all time constants with the same factor. But physical neural systems, whether biological or in neuromorphic hardware, cannot scale all physical time constants with a global factor. This leads to a very interesting mathematical and biological (and algorithmical) question: how can the qualitative dynamics of an RNN be speeded up / slowed down without a globel time constant scaling? We are aware of two approaches which both make use of the fact that when a RNN is excited by different-speed versions of the same input pattern, the elicited network states populate different regions of state space. By characterizing the geometry of these different regions with (cheaply computable) variables, and subesquently actively controlling these variables by elementary linear controllers (wyffels et al., 2014) or conceptor filters (Jaeger, 2014), speed variations up to a factor of 10 for pattern generating tasks have been achieved.

5.7 Space to time transformation

Even outside relativistic physics, space and time depend on each other. In particular, travelling solitons and waves need time proportional to travel distance. This may become exploitable for the design of neural mechanisms for variable-speed pattern recognition and generation. The idea is to encode the target pattern spatially on a neural surface and let it be "read" by a travelling wave or soliton whose travel speed is determined by a single or very few variables that can be controlled physically or algorithmically. Neural field theories of cognitive cortical processing, which are based on solitons and waves, are worked out in some detail (Engels and Schöner, 1995; Lins and Schöner, 2014), but as far as we can see, so far not with the aim of explaining speed control.

5.8 Behavior control hierarchies

In robotics and intelligent agent modeling, the cognitive control of action selection and motor control is typically organized in a hierarchy of planning and controlling layers. Higher layers in such hierarchies operate on slower timescales than lower layers. The lowest layers Hierarchical agent "architectures" are so common and have been proposed abundantly since 50 years, such that we can give only a few, exemplary ad hoc pointers. Examples: In classical AI architectures, such hierarchies are explained in terms of plan-subplan hierarchies (Saffiotti et al., 1995). In control engineering, hierarchical control architectures have become an explicit industry standard (Albus, 1993). In the archetype *subsumption architecture* (Brooks, 1989) in behavior-based robotics, higher-level "behaviors" can suppress lower ones. An influential early model in neural networks / machine learning constitutes the control hierarchy in a format of trainable hierarchical *mixture of experts* where higher-level experts can *gate* lower-level experts. A similar structure, based on ODEs where higher-layer behavior-controlling ODEs were run with slower time constants, powered several winners in RoboCup world championships

(Jaeger and Christaller, 1998). When mulit-layered RNNs are trained for robot tasks, timescale-differentiated layers of control emerge automatically (Yamashita and Tani, 2008).

5.9 Eligibility traces

In reinforcement learning (RL), a key submechanism is to represent and compute *eligibility traces* (Sutton and Barto, 1998). This refers to a number of algorithmic methods to maintain a memory trace (with weighted decay) of past actions (of a complete agent or a single neuron, in the latter case the action being spike generation) and input signals (sensor input to an agent or spike input to a neuron), paired with information about the (estimated) utility of the action history to receive reward. The setting of the decay rate determines the memory horizon. Reinforcement learning can be expected to play a large role in neuromorphic training. Recently eligibility traces have also become instrumental in designing neurally plausible (and hence potentially implementable in physical spiking neuromorphic microchips) approximate methods to emulate backpropagation learning in spiking neural networks (Bellec et al., 2019).

5.10 Time (un)warping

In many temporal machine learning tasks, the incoming signal can be sometimes faster, sometimes slower. This can happen when signal sources change (for instance, there are slow and fast speakers), but it can also happen within a single instance of an input signal (for instance, when a speaker gets excited and speaks faster, or when his/her way of pronunciation has temporal ideosyncrasies like stretching vowels longer than average speakers). This is a problem for machine learning algorithms. In brute-force learning paradigms (deep learning in particular), such *time warping* effects are caught by providing exhaustive training samples that cover all sorts of warping effects. A more training-data-economical approach is to send input signals through some time-unwarping preprocessing filter before feeding it to the RNN in training and exploitation, such that the RNN only has to cope with speed-normalized input signals. Another approach that we find the most elegant is to leave the input stream in its original time-warped version and adapt the processing speed of the RNN, speeding it up when the input signal slows down such that each RNN state update step (in discrete-time RNN models) or unit-time state evolution (in continuous-time RNN models) covers the same phenomenological change increment in the input stream (Lukoševičius et al., 2006).

5.11 Continual learning

Continual learning refers to the machine learning challenge to make a neural network (feedforward or recurrent) learn a sequence of tasks, one after the other, such that when the next task is trained into the network, the new weight adaptations do not destroy what the network has previously learnt in other tasks. This *catastrophic forgetting* (or *catastrophic interference*) problem has remained without a convincing solution since it was first acknowledged decades ago (French, 2003). Only recently, a number of novel approaches in deep learning found effective algorithmic ways to de-fuse this problem. This

is today a very active strand of research in deep learning, now called *continual learning*, which has yielded a variety of effective algorithmic paradigms and a differentiated view on variants of the problem statement (Parisi et al., 2019; He et al., 2019a). The continual learning problem is closely related to the theme of *transfer learning* (which concerns the generalization and carry-over of competences learnt on other tasks to a new task), the theme of *federated learning* (which concerns the integration of learning progress made in the peripheral nodes in a network of decentralized local learners (Kairouz et al., 2019)), and the theme of *meta-learning* (which concerns the learning of learning strategies).

Continual learning is connected to timescale and memory topics in several ways. First, in some continual learning algorithms, weight changes in synapses that are deemed important for previously learnt tasks are discouraged, reducing (= slowing down) their adaptation rate. Second, the continual learning problem in its most demanding form poses itself on the longest possible, namely the *lifelong* learning timescale. Third, some continual learning algorithms rely on generative memory replay of previously learnt tasks.

5.12 Dynamical memory in RNNs

The simple linear readout which is typically used for training RNNs in the reservoir computing (RC) field can be used to define natural numerical measures for "how much" memory about previous input is preserved in the current network state. In its most basic format, the *memory capacity* of a discrete-time "echo state" reservoir network is measured by (i) feeding it with white noise input, (ii) training linear readout units y_d by linear regression on the task to recover the input value $u(t-d)$ from d steps before, (iii) adding all correlation coefficients between signals y_d and $u(t-d)$ to get the desired measure (Jaeger, 2002). Note that the training of readout units here is not done to soleve a "useful" task but solely for quantifying an core characteristic of an RNN (or, for that matter, any other dynamical system). Note further that the "memory" which can be determined in this way is a purely dynamical short-term memory and involves no learning inside the RNN. This concept of memory capacity has become the anchor point for a (by now) extensive literature of mathematical analyses which explore memory in RNNs under aspects like the impact of noise (Antonik et al., 2018), continuous time (Hermans and Schrauwen, 2010a), high-dimensional input (Hermans and Schrauwen, 2010b), infinite-dimensional neural networks constructed by kernel methods (Hermans and Schrauwen, 2012), different neuron models (Büsing et al., 2010), or nonlinear readouts Grigoryeva et al. (2016), to name but a few. The literature is by now extensive and a systematic survey would be welcome. Frady et al. (2018) develop a classification scheme for dynamical memory tasks and measures which highlights the richness of phenomena and perspectives associated with dynamical memory in RNNs.

The memory capacity of reservoir networks has become a standard metric to quantify or predict the "goodness" of reservoir networks for cognitive tasks in studies where different network architecures (Strauss et al., 2012), reservoir pre-training methods (Schrauwen et al., 2008), or reservoir control parameter tuning are compared. The latter is often associated with investigations of reservoir performance "close to the edge of chaos" (Legenstein and Maass, 2007) (which in most cases we find an incorrect usage of terminology; correctly it should be "close to the loss of the echo state property").

Measuring delayed-input to trained output correlations is not the only way is not the only way of quantifying the dynamical memory capacity of RNNs or general input-driven dynamical systems. If one adopts a probabilistic perspective, information-theoretic measures like the Fisher memory matrix (Ganguli et al., 2008) are more informative, albeit harder to estimate empirically.

We point out that dynamical memory cast as state-based information carry-over from the past to the present, as discussed above, is not the same as *working memory*. Working memory is a complex concept used in the cognitive and neurosciences for a spectrum of transient recall phenomena in animal and human remembering (Baddeley, 2003; Botvinic and Plaut, 2006; Fusi and Wang, 2016). Working memory phenomena usually entail additional control mechanisms to encode and decode context information and insertion of knowledge stored in long-term memory.

Neither "dynamical memory", "working memory", nor "short-term memory" have generally shared, precise definitions and when one studies the literature one must be careful to appreciate the specific meaning of such terms intended by the author.

5.13   Some topics not covered

We list a number of themes that would warrant a closer inspection but for which we lacked time or expertise on this occasion:

- Generating and detecting timing and rhythm patterns in music, speech or gesture recognition / production (Eck, 2002a,b, 2007).

- Methods for dynamical adapation of learning rates in gradient-descent training of neural networks.

- Timescales in connection with statistical efficiency of neural sampling and Markov-chain Monte Carlo sampling algorithms (Neal, 1993; Jaeger, 2020).

- Interactions between adaptation rates, memory duration, and residual approximation errors in online adaptive signal processing (Farhang-Boroujeny, 1998).

- The role of delays in (neural network based) architectures for motion control.

- Subsampling and supersampling in digital signal processing.

- Attention and working memory mechanisms in deep learning, especially for language processing (Bahdanau et al., 2015).

6   Take-homes

Our meandering journey through the landscape of temporal and timescale phenomena in natural and artificial "cognitive" systems has delivered a large and speckled collection of findings. We could not bind them together in a unifying "story" (we tried this in a first write-up but had to abandon the attempt because there were many themes left that did not fit into the unified picture that we started to draw). But despite the heterogeneity of our findings, there are some lessons learnt that we believe provide useful input to the MemScales consortium at an early time in the project:

Timescales is a multidimensional concept. There are many ways in which "timescale" themes come to the surface when thinking about cognitive systems. One consequence for hardware and computational methods research in MemScales: there is not a single good (or even best) way to design neuromorphic systems with regards to timescales. How multiple timescales have to be physically and algorithmically supported depends on the use scenario of the targeted system.

Timescales cannot be ignored. Our belief that timescales and memory hierarchies are important was a *raison d'être* for launching our project. Our findings substantiate and underline this initial belief and convince us that a dedicated project focus on timescales is a necessary topic of dedicated research in the further development of neuromorphic computing.

More complex cognitive processing needs more timescales. A task's cognitive complexity seems closely linked to the spectrum of memory timescales needed for it. This indicates that for a systematic development of neuromorphic technologies it is helpful to work out a complexity hierarchy of task types and initially not "reach for the stars" but concentrate on tasks of modest complexity that require to integrate information only across a few timescales only (or even a single one).

Relative and absolute timescales. A neuromorphic computing system (hardware plus algorithms) must support a range of timescales that widens as the cognitive task complexity grows. If the system is used in offline mode (for instance, text processing), one only needs to aim for a wide range of relative timescales. If the system is targeting online processing tasks (for instance robot control or cardiac monitoring), in addition one must match the system's absolute timescales to the task data streams. The main challenge here is probably to physically realize slow enough timescales.

Tricks to avoid many physical timescales. It is not easy to realize a wide spectrum of timescales directly in physical effects on a non-digital neuromorphic microchip. There are a number of workarounds that may alleviate this challenge:

- Digital-analog hybrid processors where slow timescales are made possible by digital buffering. Needs a development of dedicated digital-analog algorithmics.
- Large (possibly *very* large) RNNs can encode large amounts of information from past input in the current network state and thus have longer dynamic memory spans. Needs microchip technologies for realizing (very) large RNNs.
- Reservoir transfer methods may have some potential for realizing long memory spans even in modestly sized RNNs if these are explicitly trained for the specific memory functionalities demanded by the target task.
- Designing RNN architectures that include explicit mode switching mechanisms (possibly trainable) may realize temporally nested processing levels. Needs the development of dedicated architectures and learning algorithms, and a clear understanding of the "stack memory" demands of a task.

Delays may make neuromorphic computing difficult. Signal travel delays in un-clocked analog neuromorphic microchips become a problem when delay times are not well separated from the fastest timescales demanded by the processing task (in which

case delays can be ignored). For high-frequency online processing tasks (for instance in future neuromorphic low-energy communication nodes), an explicit modeling and algorithmic compensation for physical delays is needed. For multi-timescale offline tasks, an upper limit for task throughput rates is given by the necessity to separate physical delays from the fastest task timescale.

We note that delays are no mathematical or algorithmical problem in digital computing as long as physical on-chip delay times are much shorter than clock cycle times.

Delays may make neuromorphic computing easy. If one would find a way to physically realize tapped delay lines (by traveling waves or solitons, maybe skyrmionic?), multiple timescale dynamics (with longest scale given by longest signal travel time on the delay line) might become explicitly designable. Needed: mathematics and algorithms embedding tapped delay lines in analog computing architectures.

Life history timescale. If the motto of brain-like computing is taken seriously, the "lifespan" timescale of an individual hardware system becomes relevant. While digital microchips don't age and don't have an individuation history: if they start processing 0's and 1's differently from when they were sold, they are called "broken" and are replaced by an identical twin. Analog neuromorphic microchips will likely be individual from the moment when they leave the fab (due to device mismatch); they will often exhibit slow parameter drift and physical aging; and they cannot be "programmed" in the traditional sense but will likely have to be trained. This will lead to individual lifelong learning and adaptation histories. Needed: novel mathematical tools to describe qualitative change and continual/lifelong learning schemes (algorithms and training schemes) that are appropriate for physically aging systems, which in particular will require a collaboration between learning and homeostatic self-stabilization mechanisms.

7 Timescale requirements for neuromorphic hardware designs for STDP and RC processing

Today, virtually all analog spiking neuromorphic hardware demonstrations are based on either STDP, RC, or sometimes a combination of the two. This also holds true for our research in the predecessor project NeuRAM3 (for instance He et al. (2019b); Yousefzadeh et al. (2018); Cove et al. (2018)). Although our survey has made it clear that biological brains as well as machine learning techniques derive their strength from a much wider range of computational / learning principles than STDP and RC, at the current point in time it makes sense to focus on these two paradigms, identify the current state of development in the theory and the practical uses thereof, and from that derive concrete (minimal) requirements for physical timescales that have to be delivered by analog spiking neuromorphic hardware. We treat STDP and RC in turn, but start with a general summary of timescales and their biological, algorithmical and hardware realizations.

7.1 Timescales: biological, algorithmical, hardware

Research in the neurosciences has identified a plethora of neural adaptation mechanisms. They are based on a wide spectrum of physical and physiological mechanisms and operate on all levels of the brain's hierarchical architecture, from synapses to membranes to entire neural assemblies and projection pathways; and they serve many functions (as far as one can identify them today) in homeostatic regulation, fast and slow adaptation to input characteristics, short-term, working and long-term memory, learning and ontogenesis. This richness is far from being fully understood in the neurosciences, and there exists no unified or comprehensive mathematical model.

Nonetheless, it is instructive to be aware of some core concepts and findings from neuroscience. Table 1 gives a highlevel indicative overview which reflects the ongoing discussions in the consortium. It remains to be explored which physical effects of hardware devices can serve which biologically motivated mechanism. This is an intricate question because it is not the physical device / effect per se that serves a computational/biological mechanism, but a complex interplay of the core physical effect with circuit designs and control schemes, like for instance pulse pattern schemes for setting PCM resistances.

Biological plasticity phenomenon	Timescale	Mechanism	Candidate physical device / effect
Short-term plasticity	1 ms – 10 ms	STDP, SDSP	capacitors
Long-term plasticity	10 ms – 500 ms for weight change; 1 h – years for weight preservation	LTP/LTD	non-volatile memristive devices (for preserving the results of LTP)
Intrinsic plasticity	0.5 s – 10 s	threshold adaptation	volatile ReRAM, TFT, ...
Homeostatic plasticity	1 s – 1 h	synaptic scaling	volatile ReRAM, PCM drift, TFT, ...
Structural plasticity	1 h – lifetime	architecture reorganisation	reconfigurable / extendable architectures

Table 1 Overview of plasticity phenomena

The large number of physiological mechanisms underlying this spectrum of phenomena, as well as the wealth of formal models in theoretical neuroscience that capture these phenomena at different levels of abstraction, as well as physical differences between brain physiology and electronics, make it impossible to copy biological mechanisms 1-1 to electronic microchips. Furthermore, it is not necessarily the most promising engineering strategy to even try to copy brain mechanisms exactly into analog spiking hardware. On the one hand, many biochemical mechanisms will be hard to replicate in electronic

systems, and on the other hand, electronic systems may offer opportunities (especially, faster timescales or very long-term non-volatile memory states) that are not affored in physiological brain substrates. Yet, Table 1 teaches a clear lesson: in order to endow artificial with proxies of the biological inference, adaptation and learning mechanisms, a wide range of timescales must be covered.

How this is concretely done will depend on the available hardware, targeted performance and use-cases, and algorithmic models. In the following subsection we will work this out in an examplary case study.

7.2 Timescale requirements resulting from demands for STDP learning: a worked-out case study

There is not a single, well-defined STDP adaptation rule in biological brains. In fact, it is an experimental challenge to localize, measure and formulate STDP mechanisms in mammalian brains. In the machine learning / computational neuroscience / neuromorphic engineering communities, a broad variety of STDP variants and combinations of them with other neural adaptation mechanism have been explored — Joshi and Triesch (2009); Clopath et al. (2010); Graupner and Brunel (2012); Galtier and Wainrib (2013a); Klampfl and Maass (2013); Roclin et al. (2013); Bengio et al. (2017); Thiele et al. (2018) are but a small selection of approaches that document this variability. The initial specific concept of STDP (as described in the landmark paper by Markram et al. (1997), with many forerunners) does not cover this variability. The term "spike-timing-dependent synaptic plasticity" and the acronym STDP was introduced in Song et al. (2000). The term *Spike-Driven Synaptic Plasticity* (SDSP), apparently introduced by Fusi et al. (2000) in a formal model of an adaptive synapse independent of, but potentially effective in a variety of learning/adaptation mechanisms, should be preferred over the term STDP when one considers spike-driven synaptic plasticity phenomena in a more general setting than the original STDP framing. Since neuromorphic electronic circuits and neural network learning algorithms used in them explore and exploit more general mechansim than STDP proper, we will use the term SDSP in this section.

In order to become concrete, we must however settle on a specific model, and this should not be a repetition of what we already developed in NeuRAM3. Instead, our choice should open doors for the currently most promising line of SDSP exploits, the so-called 3-factor rules. Again, this principle comprises many different variants. Generally speaking, in 3-factor SDSP rules, the adaptation effects determined by the basic two factors (pre- and postsynaptic activations) of SDSP become multiplicatively modulated by a third factor, which represents some kind of global control signal, which can be variously interpreted as a reward signal, a derivate of a supervised target signal, a temporal coordination / synchronization guide, or a mean-field population activity signal for achieving homeostatic regulation of a neuron's average activity level (summarized in Kusmierz et al. (2017)).

Our choice is to opt for the first among the mentioned interpretations, and concretely for the model recently proposed by Bellec et al. (2019, 2020). This model, named the *e-prop* model, imports mechanisms from reinforcement learning and utilizes them to realize an approximation of stochastic gradient descent (SGD) with an SDSP mechanism. SGD is the main enabling learning principle that empowers deep learning techniques,

and is thus of great potential value for neuromorphic technologies since the current state of the art in machine learning is defined through SGD trained neural networks. However, in the deep learning field, one does not use spiking neuron models. Much effort has been spent in the last 10 years to find approximations of SGD that also work in spiking neural environments, with limited success. The model of Bellec et al. has immediately created a strong resonance, building on and transcending previous approaches to apprximate SGD in spiking networks, is mathematically transparent, can be adopted to a variety neuron models, has been explicitly formulated with analog spiking hardware implementations in mind, and furthermore MemScales members (Indiveri, Jaeger) enjoy a long-standing collaboration with the group of Wolfgang Maass where this model originates. Closely related SDSP-realized approximations of SGD are currently being explored in a number of research groups. Nefti and Averbeck (2019) review approaches of transferring neurobiological models of reinforcement learning to artificial neural networks, emphasizing the benefits of neuron models that include sub-mechanisms that operate on different timescales, and report brackets for biological time constants. Payvand et al. (2020) (whose first author is a member of the INI) present an analog circuit for an on-chip realization of (a version of) such learning rules, and demonstrate it in a simulation. Concrete values of effective time constants are unfortunately not provided. The documentation of mathematical formalism in Bellec et al. (2019, 2020) is particularly detailed, which gives us the option to analyse conditions on time constants, which we now proceed to do.

Following Bellec et al. (2020), we first give a brief summary of e-prop for the case of leaky integrate-and-fire (LIF) neurons (formulated in a discrete-time setting, using a unit timestep of $\delta t = 1$ millisecond), the most simple and arguably most popular spiking neuron model in neuromorphic engineering theory. The core of SGD algorithms in supervised learning for the adapation of a synaptic weight w_{ji} from pre-synaptic neuron i to post-synaptic neuron j is the error gradient $\frac{dE}{dw_{ji}}$, which can be factorized as

$$\frac{dE}{dw_{ji}} = \sum_t \frac{dE}{dz_j^t} \cdot \left[\frac{dz_j^t}{dw_{ji}} \right] =: \sum_t L_j^t \, e_{ji}^t, \tag{5}$$

where z_j^t is the postsynaptic spike train (a binary signal), the summation goes over the time points of the learning history, the factor $dE/dz_j^t =: L_j^t$ is the *learning signal*, and the factor $dz_j^t/dw_{ji} =: e_{ji}^t$ is the *eligibility trace*. The elegibility trace depends on pre- and postsynaptic spiking (see below) and are thus a form of SDSP. The learning signal is the "third" factor in the customary terminology when one speaks of 3-factor rules.

Note that this formulation (5) captures the weight change gradient obtained from accumulating information about a whole training sequence or a training batch. For an instantaneous weight adapation in a single model update step from time t to $t+\delta t = t+1$, as needed for adaptive hardware implementations, (5) reduces to the online learning rule

$$\Delta^t w_{ji} = -\eta \, L_j^t \, e_{ji}^t, \tag{6}$$

where η is a learning rate. We now take a closer look first at the elibility trace and the "third factor", the learning signal, in that order. We first give an brief summary account of the formalism in Bellec et al. (2020), which is geared toward discrete-time simulations on a digital computer, and then discuss what conditions on physical time constants in unclocked event-based analog hardware implementations can be derived.

Since spike pre- or postsynaptic spike trains z_j^t are not differentiable, they are replaced by exponentially smoothed filtered versions

$$\bar{z}^t := \mathcal{F}_\alpha(z^t) := \alpha\,\mathcal{F}_\alpha(z^{t-1}) + z^t \tag{7}$$

when needed. Bellec et al. (2020) derive that the eligibility trace e_{ji}^t can then be re-written as

$$e_{ji}^{t+1} = \psi_j^{t+1}\,\bar{z}_i^t, \tag{8}$$

where ψ_j^t is a pseudo-derivative of $\partial z_j^t\,/\,\partial v_j^t$ (used variously in the literature for making spike trains differentiable under consideration of the post-synaptic neuron's j refractory period r; Bellec et al. (2020) refer back to Bellec et al. (2018)), given by

$$\psi_j^t := \begin{cases} 0 & \text{for } t \text{ inside } r \\ \frac{1}{v_{\text{th}}}\,\gamma_{\text{pd}}\,\max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}}}{v_{\text{th}}}\right|\right) & \text{else,} \end{cases} \tag{9}$$

where in turn v_j^t is the membrane potential of neuron j, v_{th} its firing threshold, and γ_{pd} is a heuristic damping parameter that is set to $\gamma_{\text{pd}} = 0.3$ by Bellec et al; the role of this damping parameter is to improve numerical stability of approximated gradient descent in networks that have many layers. The membrane potential v_j^t evolves according to

$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{\text{rec}}\,z_i^t + \sum_i W_{ji}^{\text{in}}\,x_i^{t+1} - z_j^t\,v_{\text{th}}, \tag{10}$$

$$z_j^t = H(v_j^t - v_{\text{th}}), \tag{11}$$

where $W_{ji}^{\text{rec}}, W_{ji}^{\text{in}}$ are the recurrent and input weights to neuron j, x^t is the input signal and H is the Heaviside step function. The decay rate α is can be expressed in terms of an exponential decay function by

$$\alpha = \exp(-\delta t/\tau_m), \tag{12}$$

where τ_m is the membrane time constant. A biologically plausible value is $\tau_m = 20$ ms. With a stepsize $\delta t = 1$ ms (which is used in the simulations in Bellec et al. (2020)), this gives $\alpha \approx 0.95$.

The learning signal L_j^t in (5) measures the deviation between the output signals y_k generated by the network output neurons k (which are not recurrently connected to each other), defined by the leaky integration rule

$$y_k^t = \kappa y_k^{t-1} + \sum_j W_{kj}^{\text{out}}\,z_j^t + b_k^{\text{out}}, \tag{13}$$

and the target outputs $y_k^{*,t}$ by a simple linear combination of the errors

$$L_j^t = \sum_k B_{jk}(y_k^t - y_k^{*,t}). \tag{14}$$

The error backprojection weights B_{jk} are determined in Bellec et al. (2020) according to various heuristics, among them fixing them at random values. While this works satisfactorily in the demonstrations given in Bellec et al. (2020), we see this dependence on

heuristic intuition to define the learning signal as an opportunity for further improvements of this model. Which values of κ were chosen in the demonstrations in Bellec et al. (2020) remained un-documented there. However, it is clear that for tasks where the target outputs $y_k^{*,t}$ are smooth signals, they must be assumed to be high-pass filtered to preclude arbitrary baseline drifts which cannot be learnt by neuronal outputs. To connect our following discussion of synaptic/neuronal time constants with task-specific time constants, we will consider the period length T^* (in milliseconds) of the lowest significant frequency in y^{*,t_k} as the slowest task-relevant timescale. The challenge for online learning with spiking neurons is to be slow enough to be able to integrate task-relevant information on that timescale T^*.

We now consider the question how this model, which is formulated in a discrete-time set-up for simulation on digital computers, translates into requirements for RNN implementations in unclocked, event-based, spiking hardware.

We first consider the second factor $\bar{z}_i^t$ in the eligibility trace (8), which represents pre-synaptic spikes arriving at the synapse w_{ji}. Note that the 1 ms time difference between $t+1$ and t in (8) is due to the discrete-time simulation scenario, where a unit time step is assumed for propagating the information from neuron i to neuron j. For electronic event-based neuromorphic hardware we may assume that the travel delays of electric signals are negligible, hence instead of (8) we will consider

$$e_{ji}^t = \psi_j^t \, \bar{z}_i^t. \tag{15}$$

According to (7) and (12), $\bar{z}_i^t$ is an exponentially smoothed version of z_i^t with an exponential time constant that we will call τ_{pre}. In order to exploit (7) in analog unclocked hardware, a physical variable available at the physical implementation of synapse w_{ji} must represent $\bar{z}_i^t$, that is, a physical leaky integration of the incoming spike train z_i^t with time constant τ_{pre} must be effected somewhere in the circuit — either at the sending neuron i (then this signal must be sent to all receiving neurons j), or at the receiving synapse w_{ji} (then the integration must by physically repeated at all synapses to which i sends out its spike train).

We note that it is not possible to derive general rules for how τ_{pre} should be set for the network to solve its learning task. Whether a specific setting will be successful depends on many design variables, for instance the size of the RNN (in larger RNNs, less precision per synapse is needed), functional specialization of neurons i and j (they might specialize on high-frequency components in the outward task, leading to more relaxed constraints on τ_{pre}), the average firing rate of the feeding neuron i (the higher, the smaller can τ_{pre} be), and importantly, a model of how task-relevant information is encoded in z_i^t. We must make further assumptions to arrive at a well-defined problem. We will proceed under the following assumptions.

1. Task-relevant information is encoded in the network by rate coding.

2. The synapse w_{ij} contributes significantly to the network's functionality with regards to the slowest task-relevant timescale T^*.

The leaky integration of z_i^t should be such that a significant memory trace of spikes that lie T^* in the past is still present in $\bar{z}_i^t$. What "significant" means is subject to an essentially arbitrary commitment. Here we opt for a plausible heuristic and require that

the contribution of $z_i^{t-T^*}$ to $\bar{z}_i^t$ is reduced by a *forgetting factor* F of at most $1/2$ at time t. This leads to the condition

$$\exp(-T^*/\tau_{\mathsf{pre}}) \geq 1/2, \tag{16}$$

that is

$$\tau_{\mathsf{pre}} \geq T^* \frac{-1}{\log(1/2)} \approx 1.4 \cdot T^*. \tag{17}$$

Next we turn to the first factor ψ_j^t in the eligibility trace (15). This factor accounts for the postsynaptic spike timing when interpreted in an SDSP perspective. Inspecting (9) we see that this factor follows the temporal profile of the membrane potential v_j^t of neuron j, which in turn (see (10)) is a leaky integration of recurrent and input spike trains arriving at j. We have to transfer the discrete-time formulation of Bellec et al. to the continuous-time, event-based situation in analog unclocked hardware, that is, we must translate the discrete timestep discount factor α in (10) to an exponential decay rate that we will call (like Bellec et al. do) τ_m. Again we must make additional assumptions to arrive at a specific statement of our problem. Repeating the assumptions and the heuristic that we committed above, we arrive at the same conclusion as in (17):

$$\tau_m \geq T^* \frac{-1}{\log(1/2)} \approx 1.4 \cdot T^*. \tag{18}$$

This suggests that membrane leaking time constants are needed that are in the order of the slowest relevant task time constants. Bellec et al. (2020) used a biologically motivated value of $\tau_m = 20$ ms. They demonstrated their model on a supervised task of phoneme recognition, where T^* is 10 ms, which satisfies our constraints (17) and (18). However, in another experiment, where e-prop was adapted to a reinforcement learning situation, the task-relevant slowest timescale was in the order of $T^* = 2000$ ms, still with $\tau_m = 20$ ms. This is at odds with (18). The solution to this puzzle is an argument that combines the influence of network size with the choice of the forgetting factor $F = 1/2$. If we plug in smaller forgetting factors in (17), (18), we end up with smaller admissible time constants $\tau_{\mathsf{pre}}, \tau_m$. They result in smaller T^*-delayed task-relevant additive components in the signals $\bar{z}_i^t, \psi_j^t$, a source of variation which in turn can however be compensated by the linear combinations effected through $W_{ji}^{\mathrm{rec}}, W_{ji}^{\mathrm{in}}$. This efficacy of this compensation scales with the size N of the RNN and the numerical accuracy of the used computing environment. Bellec et al. used floating-point precision arithmetics and large networks (with 2400 neurons in the phoneme recognition demo, not documented for the reinforcement learning task). In this light, our suggestions (17), (18) should be considered as extremely conservative if not pessimistic, relevant (only) for very small networks with a few neurons and low numerical precision (or with noise).

We summarize our findings:

- For implementing the e-prop algorithm for supervised training or RNNs in analog spiking neuromorphic hardware on the basis of elementary LIF neuron models, two leaky integration mechanisms are needed, one for the membrane potential and one for the smoothing of spike trains arriving at a synapse.

- Lower bounds on the minimally necessary time constants for these two integration mechanisms depend on a number of design variables (in particular network size and realizable numerical accuracy) and task specifics (in particular slowest task-relevant time constant in task signals). Under the most conservative assumptions (very small network, low numerical accuracy) one can reason that the leaky integration time constants for the membrane and synapse integrations must be in the order of the slowest task-specific time constant. As network size and/or available numerical accuracy increases, increasingly faster neuronal/synaptic time constants can be expected to be sufficient for realizing the e-prop algorithm.

- Not all membrane or synapse time constants need to satisfy the conditions described here. In order to enable a RNN architecture to cope with the slowest task-relevant timescales, it is enough if *some* neurons / synapses are capable of the required slow integrations. Specifically, hierarchical network architectures are often designed in a way that "higher" layers operate in slower timescale modes than "lower" layers.

We emphasize that the considerations made above are tied to the specifics of the e-prop algorithm, with its specific version of SDSP and its specific training objective and system architecture proposed in Bellec et al. (2020). There are many other SDSP rules, other training objectives (in particular, unsupervised ones, or tasks based on non-temporal data) and architectures, where other considerations would have to be done. In particular, the necessity of leaky-integrating incoming spike trains at each synapse is a consequence of the specific e-prop mathematics and will not be required in many other SDSP versions, tasks or architectures.

The main lesson to be drawn from this case study is that there should be a mechanism in the neuromorphic system whose time constant matches the slowest timescale T^* of the outward task (where "matching" needs to be qualified, it need not be identity but can mean that the corresponding hardware mechanism has a faster timescale that can be expanded to the task timescale through computational effects). We found the same lesson taught to us in a quite different experimental and algorithmic scenario too, as will be reported in Section 7.3. If that lesson holds true, then a very wide range of task-dictated timescales T^* must be served: ranging from milliseconds in robot/prosthetics control to days or weeks or even years in environmental monitoring, just to name two application tasks that have been proposed as targets for neuromorphic computing technologies.

We emphasize that this case study does not imply a recommendation to for Mem-Scales research to implement this specific model. We chose it as a representative because the article of Bellec et al. (2020) gave a mathematical model in all detail, from which we could develop an exemplary analysis. Other SDSP models have been or are being explored in our consortium, like Yousefzadeh et al. (2018) or Cartiglia et al. (2020). It is impossible to provide a theoretical analysis that covers all options, and it would be inappropriate to try to identify a "best" one.

We finally point out that the e-prop algorithm has recently been employed in a reservoir computing set-up, where it was used to optimize the recurrent weights of a reservoir for an entire class of learning tasks in a "learning to learn" scenario (Subramoney et al., 2021).

7.3 Timescale requirements for RC systems based on analog spiking event-based neuromorphic hardware

Reservoir computing based on physical reservoirs is a flourishing research area. Physical RC systems have been built on the basis of many different non-digital physical substrates. Popular media include optics (Antonik et al., 2018), nano-mechanics (Coulombe et al., 2017), carbon nanotubes (Dale et al., 2016), magnetic skyrmions (Prychynenko et al., 2018), spintronics (Torrejon et al., 2017), or gold nanoparticle thin films (Minnai et al., 2018) (survey in Tanaka and et al (2019)). These studies are mostly experimental. Theoretical analyses, or at least systematic explorations of dynamical phenomena in controlled simulations, are scarce. We are aware only of one work in the optical RC community (Grigoryeva et al., 2016) and another one in memristive electronics based reservoirs (Sheldon et al., 2020), which is however still rather rudimentary.

An inherent obstacle to general theoretical analyses is that every physical systems comes with ideosyncratic dynamical properties that leave their mark on the computational properties of the respective system, and would have to be analysed on a case by case basis. While a large body of analytical research has accumulated over the last two decades for reservoirs that are mathematically defined on the simplest possible rate-based neuron model (the echo state networks), insights made there do not easily carry over to other sorts of reservoirs. Specifically, no theoretical analyses of computational / learning characteristics of reservoirs based on analog spiking continuous-time neural networks are yet available.

A natural starting point for such analyses, with special attention paid to timescale phenomena, would be to study the *memory capacity* (MC) of analog spiking reservoir RNNs. In its original format, which was expressed for discrete-time non-spiking reservoirs of the *echo state network* type, the MC is a measure for how many previous inputs of a one-dimensional white noise signal can be recovered by trained linear readouts, weighted with an accuracy factor. In the work that started this research line (Jaeger, 2002) it was shown that MC is bounded by the number of neurons in the reservoir. This triggered a large number of follow-up studies (a Google Scholar query on "echo state network" "memory capacity" returns more than 600 papers) which extended the original analysis with regards to input signal type, neuron model, noise robustness, input dimension, network architecture, alternative definitions of MC, and more. Contributions came from mathematics, theoretical physics, the neurosciences and machine learning. The broad interest in this question can be explained by the fundamental nature of the question of information transport in dynamical systems in general, the relevance for machine learning tasks (see Dambre et al. (2012) for the intimate connection between memory properties and general computational capacities), and the relevance for understanding dynamical short-term memory in biological brains. We are however not aware of mathematical analyses of MC in spiking RNNs, although it has been experimentally measured in a number of simulation studies.

In the group of Jaeger at the University of Groningen, the PhD student Dirk Doorakkers, whose position is funded through MemScales and who is a mathematician with a specialization in dynamical systems theory (and who authored Section 4 of this deliverable report), will carry out a dissertation project that centrally addresses the question of information transport in spiking RNNs. His project, with the working title *Double transients in multi-timescale systems provide a geometric description*

for dynamic coding with activity-silent working memory, plans a rigorous analysis of mechanisms in spiking RNNs where

1. an input signal is initially *encoded* in a temporal activation pattern of the RNN, which

2. propagates in time through the RNN for a delay ("memory") period d, undergoing a sequence of transformations, until

3. upon a cue signal a desired output transform of the input signal is recovered by a *decoding* ("readout") mechanism.

This analysis will be done with the tools of contemporary dynamical systems theory, in particular slow-fast systems (singular perturbation methods) and bifurcation theory, aiming for a characterization of such memory mechanisms in terms of generic geometrical dynamical systems concepts, which to a certain degree would render the analyses transferable to general classes of multi-timescale hardware reservoirs. This will constitute a substantial contribution to task T 1.4, *Toward a general model of unconventional computing.*

For the time being, the best that we can offer is a summary of findings that we collected in the NeuRAM3 forerunner project to MemScales. Jaeger's group was charged to realize an online heartbeat anomaly classifier on the Dynap-se, a spiking analog neuromorphic microchip developed at the Institute of Neuroinformatics in Zurich (Moradi et al., 2018). The challenge was that the natural time constant of human heartbeats is 1 sec, while the slowest available time constants for spike train integration on the Dynap-se were much faster. Our findings and methods are reported in He et al. (2019b). Here we give a summary account, which agrees well with our observations in Section 7.2:

- In earlier simulations (not reported in He et al. (2019b)) we found that the learning task was possible with spike train integration time constants that matched the task time constants.

- The unavailability of physical time constants that were as slow as the 1 sec time constant of the heartbeat data led to failures in "direct-attack" attempts to train a Dynap-se based reservoir.

- The task became solvable on the Dynap-se when a novel *reservoir transfer* method was employed to pre-configure the synaptic weights in the hardware reservoir in a way that allowed linear combinations of spike trains arriving at a receiving neuron to compensate for the small forgetting factors (see Section 7.2) inherent in the Dynap-se physics. The reservoir comprised about 750 neurons.

References

J. S. Albus. A reference model architecture for intelligent systems design. In P. J. Antsaklis and K. M. Passino, editors, *An Introduction to Intelligent and Autonomous Control*, chapter 2, pages 27–56. Kluwer Academic Publishers, 1993.

J. F. Allen. Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.

P. Antonik, M. Hermans, M. Haelterman, and S. Massar. Random pattern and frequency generation using a photonic reservoir computer with output feedback. *Neural Processing Letters*, 47(3):1041–1054, 2018.

A. Baddeley. Working memory: looking back and looking forward. *Nature Reviews: Neuroscience*, 4(10):829–839, 2003.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015. URL http://arxiv.org/abs/1409.0473v6.

Cornelia I Bargmann. Chemosensation in c. elegans. In *WormBook: The Online Review of C. elegans Biology [Internet]*. WormBook, 2006.

Naama Barkai and Stan Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–917, 1997.

Marina Bedny, Hilary Richardson, and Rebecca Saxe. "visual" cortex responds to spoken language in blind children. *Journal of Neuroscience*, 35(33):11674–11681, 2015.

G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. arxiv manuscript, 2018. URL https://arxiv.org/abs/2006.12484.

G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass. Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. arxiv manuscript, 2019. URL https://arxiv.org/abs/1901.09049.

G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):1–15, 2020.

T. Bengio, Y. abd Mesnard, S. Fischer, A. abd Zhang, and Y. Wu. Stdp-compatible approximation of backpropagation in an energy-based model. *Neural Computation*, 29(3):555–577, 2017.

P. Berkes and L. Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Cognitive Sciences EPrint Archives (CogPrints)*, 2804, 2003. URL http://cogprints.org/2804/.

A. Bernacchia, H. Seo, D. Lee, and X.-J. Wang. A reservoir of time constants for memory traces in cortical neurons. *Nature Neuroscience*, 14(3):366–372, 2011.

Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

M. M. Botvinic and D. C. Plaut. Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, 113(2):201–233, 2006.

CAR Boyd. Cerebellar agenesis revisited. *Brain*, 133(3):941–944, 2010.

Dennis Bray. Protein molecules as computational elements in living cells. *Nature*, 376 (6538):307–312, 1995.

R.A. Brooks. The whole iguana. In M. Brady, editor, *Robotics Science*, pages 432–456. MIT Press, Cambridge, Mass., 1989.

L. Büsing, B. Schrauwen, and R. Legenstein. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural Computation*, 22(5): 1272–1311, 2010.

C. Callender. Introduction. In C. Callender, editor, *The Oxford Handbook of Philosophy of Time*. Oxford University Press, 2011.

Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a hebbian learning rule. *Annu. Rev. Neurosci.*, 31:25–46, 2008.

M. Cartiglia, G. Haessig, and G. Indiveri. An error-propagation spiking neural network compatible with neuromorphic processors. In *Proc. 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 84–88, 2020.

E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. of the IEEE*, 102(9):1367–1388, 2014.

S.-N. Chow and J. Mallet-Paret. Singularly perturbed delay differential equations. In J. Chandra and A.C. Scott, editors, *Coupled Nonlinear Oscillators*, pages 7–12. North-Holland Publishing Company, 1983.

Claudia Clopath, Lars Büsing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: a model of voltage-based stdp with homeostasis. *Nature neuroscience*, 13(3):344, 2010.

S. Cook. The P versus NP problem. Official problem description of the fourth millenium problem, Clay Mathematics Institute, 2000. http://www.claymath.org/sites/default/files/pvsnp.pdf.

Leon N Cooper and Mark F Bear. The bcm theory of synapse modification at 30: interaction of theory with experiment. *Nature Reviews Neuroscience*, 13(11):798–810, 2012.

J. C. Coulombe, M. C. A. York, and J. Sylvestre. Computing with networks of nonlinear mechanical oscillators. *PLOS ONE*, 12(6), 2017. URL Https://doi.org/10.1371/journal.pone.0178663.

E. Cove, R. George, J. Frascaroli, S. Brivio, C. Mayr, H. Mostafa, G. Indiveri, and S. Spiga. Spike-driven threshold-based learning with memristive synapses and neuromorphic silicon neurons. *Journal of Physics D: Applied Physics*, 51:344003, 2018.

M. Dale, S. Miller, J. F. anbd Stepney, and M. A. Trefzer. Evolving carbon nanotube reservoir computers. In *Proc. Int. Conf. on Unconventional Computation and Natural Computation*, pages 49–61, 2016.

J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar. Information processing capacity of dynamical systems. *Nature Scientific Reports*, 2:id 514, 2012.

William J Davis. Behavioural hierarchies. *Trends in Neurosciences*, 2:5–7, 1979.

Declan A Doyle, Joao Morais Cabral, Richard A Pfuetzner, Anling Kuo, Jacqueline M Gulbis, Steven L Cohen, Brian T Chait, and Roderick MacKinnon. The structure of the potassium channel: molecular basis of k+ conduction and selectivity. *science*, 280 (5360):69–77, 1998.

D. Durstewitz, J. K. Seamans, and T. J. Sejnowski. Neurocomputational models of working memory. *Nature Neuroscience*, 3:1184–91, 2000.

D. Eck. Real-time musical beat induction with spiking neurons. Technical Report IDSIA-22-02, IDSIA, Instituto Dalle Molle di studi sull' intelligenza artificiale, Galleria 2, CH-6900 Manno, Switzerland, 2002a. ftp://ftp.idsia.ch/pub/techrep/IDSIA-22-02.ps.gz.

D. Eck. Finding downbeats with a relaxation oscillator. *Psychological Research*, 66(1): 18–25, 2002b. http://www.iro.umontreal.ca/~eckdoug/papers/2002_psyres.pdf.

D. Eck. Identifying metrical and temporal structure with an autocorrelation phase matrix. *Music Perception*, 2007. http://www.iro.umontreal.ca/~eckdoug/papers/2006_rppw_draft.pdf.

Gerald M Edelman. Neural darwinism: selection and reentrant signaling in higher brain function. *Neuron*, 10(2):115–125, 1993.

Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *science*, 338(6111):1202–1205, 2012.

J. Endrullis, J. W. Klop, and R. Bakhshi. Transducer degrees: atoms, infima and suprema. *Acta Informatica*, 57(3-5):727–758, 2019.

Ch. Engels and G. Schöner. Dynamic fields endow behavior-based robots with representations. *Robotics & Autonomous Systems*, 14:55–77, 1995.

G.B. Ermentrout and D. Terman. *Mathematical Foundations of Neuroscience*, volume **35** of *Interdisciplinary Applied Mathematics*. Springer, New York NY, 2010.

B. Farhang-Boroujeny. *Adaptive Filters: Theory and Applications*. Wiley, 1998.

Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. In *Cereb cortex*. Citeseer, 1991.

N. Fenichel. Geometric singular perturbation theory for ordinary differential equations. *Journal of Differential Equations*, **31**:53–98, 1979.

Chrisantha T Fernando, Anthony ML Liekens, Lewis EH Bingle, Christian Beck, Thorsten Lenser, Dov J Stekel, and Jonathan E Rowe. Molecular circuits for associative learning in single-celled organisms. *Journal of the Royal Society Interface*, 6 (34):463–469, 2009.

K. D. Forbus. Qualitative physics: past, present and future. In *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*, pages 239–296. Morgan Kaufmann, 1988.

E.P. Frady, D. Kleyko, and F.T. Sommer. A theory of sequence indexing and working memory in recurrent neural networks. *Neural Computation*, 30(6):1449–1513, 2018.

M. Franzius, B. Wilbert, and L. Wiskott. Invariant object recognition with slow feature analysis. In *Proc. of ICANN 2008*, number 5163 in Lecture Notes in Computer Science, pages 961–970. Springer Verlag Berlin, 2008. DOI: 10.1007/978-3-540-87536-9_98.

Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

R. M. French. Catastrophic interference in connectionist networks. In L. Nadel, editor, *Encyclopedia of Cognitive Science*, volume 1, pages 431–435. Nature Publishing Group, 2003.

S. Fusi and X.-J. Wang. Short-term, long-term, and working memory. In M. Arbib and J. Bonaiuto, editors, *From Neuron to Cognition via Computational Neuroscience*, pages 319–344. MIT Press, 2016.

S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. J. Amit. Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation. *Neural Computation*, 12(10): 2227–2258, 2000.

M. Galtier and G. Wainrib. Multiscale analysis of slow-fast neuronal learning models with noise. *Journal of Mathematical Neuroscience*, **2**(13), 2012. doi: `10.1186/2190-8567-2-13`.

M. Galtier and G. Wainrib. A biological gradient descent for prediction through a combination of STDP and homeostatic plasticity. *Neural Computation*, 25(11):2815–2832, 2013a. URL `http://de.arxiv.org/abs/1206.4812`.

M.N. Galtier and G. Wainrib. A biological gradient descent for prediction through a combination of STDP and homeostatic plasticity. *Neural Computation*, **25**:2815–2832, 2013b.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *PNAS*, 105(48):18970–18975, 2008.

Shangce Gao, MengChu Zhou, Yirui Wang, Jiujun Cheng, Hanaki Yachi, and Jiahai Wang. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, 30(2):601–614, 2018.

James Garson. Modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2014 edition, 2014.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

H. Geuvers, A. Koprowski, D. Synek, and E. van der Weegen. Automated machine-checked hybrid system safety proofs. In *Proc. Int. Conf. on Interactive Theorem Proving*, pages 259–274. Springer, 2010.

M. Graupner and N. Brunel. Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *PNAS*, 109(10):3991–3996, 2012.

Jesse M Gray, Joseph J Hill, and Cornelia I Bargmann. A circuit for navigation in caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 102(9): 3184–3191, 2005.

L. Grigoryeva, J. Henriques, L. Larger, and J.P. Ortega. Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals. *Neural Computation*, 28(7):1411–1451, 2016.

H. Haken. *Advanced Synergetics - Instability Hierarchies of Self-Organizing Systems and Devices*, volume 20 of *Springer Series in Synergetics*. Springer, Berlin/Heidelberg, 1983.

J.D. Hart, L. Larger, T.E. Murphy, and R. Roy. Delayed dynamical systems: networks, chimeras and reservoir computing. *Philosophical Transactions of the Royal Society A*, **377**:20180123, 2019.

X. He, Sygnowski J., A. Galashov, A. A. Rusu, Y. W. Teh, and R. Pascanu. Task agnostic continual learning via meta learning. In *Proc. neurIPS 2019*, 2019a. https://arxiv.org/abs/1906.05201.

X. He, T. Liu, F. Hadaeghi, and H. Jaeger. Reservoir transfer on analog neuromorphic hardware. In *Proc. 9th Int. IEEE/EMBS Conf. on Neural Engineering*, pages 1234–1238, 2019b.

Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949.

M. Hermans and B. Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural Networks*, 23(3):341–355, 2010a.

M. Hermans and B. Schrauwen. Memory in reservoirs for high dimensional input. In *Proc. WCCI 2010 (IEEE World Congress on Computational Intelligence)*, pages 2662–2668, 2010b.

M. Hermans and B. Schrauwen. Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation*, 24(1):104–133, 2012.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

John J Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.

John J Hopfield and Carlos D Brody. What is a moment? transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences*, 98(3):1282–1287, 2001.

Aapo Hyvärinen and Erkki Oja. Independent component analysis by general nonlinear hebbian-like learning rules. *signal processing*, 64(3):301–313, 1998.

B. Ibarz, J.M. Casado, and M.A.F. Sanjuan. Map-based models in neuronal dynamics. *Physics Reports*, **501**:1–74, 2011.

A.F. Ivanov and A.N. Sharkovsky. Oscillations in singularly perturbed delay equations. In C.K.R.T. Jones, U. Kirchgraber, and H.O. Walther, editors, *Dynamics Reported: Expositions in Dynamical Systems*, volume New series: **1**, pages 164–224. Springer-Verlag, 1992.

E.M. Izhikevich. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. MIT Press, Cambridge MA, 2007.

Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

H. Jaeger. Short term memory in echo state networks. GMD-Report 152, GMD - German National Research Institute for Computer Science, 2002. URL `https://www.ai.rug.nl/minds/uploads/STMEchoStatesTechRep.pdf`.

H. Jaeger. Controlling recurrent neural networks by conceptors. Technical Report 31, Jacobs University Bremen, 2014. arXiv:1403.3369.

H. Jaeger. Computability and complexity. Lecture notes of theoretical computer science ii, Jacobs University Bremen, 2019. `https://www.ai.rug.nl/minds/uploads/scriptCC.pdf`.

H. Jaeger. Exploring the landscapes of "computing": digital, neuromorphic, unconventional — and beyond. manuscript arXiv:2011.12013, 2020. URL `http://arxiv.org/abs/2011.12013`.

H. Jaeger and Th. Christaller. Dual dynamics: Designing behavior systems for autonomous robots. *Artificial Life and Robotics*, 2:108–112, 1998.

Xin Jin and Rui M Costa. Shaping action sequences in basal ganglia circuits. *Current opinion in neurobiology*, 33:188–196, 2015.

P. Joshi and J. Triesch. Optimizing generic neural microcircuits through reward modulated STDP. In *Proc. ICANN 2009*, pages 239–248, 2009.

Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, K. A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G.L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adria Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečny, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramer, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. arxiv report, 2019. URL https://arxiv.org/abs/1912.04977.

Eric R Kandel, Yadin Dudai, and Mark R Mayford. The molecular and systems biology of memory. *Cell*, 157(1):163–186, 2014.

T.J. Kaper. An introduction to geometric methods and dynamical systems theory for singular perturbation problems. *Proceedings of Symposia in Applied Mathematics*, **56**: 85–131, 1999.

Harris S Kaplan, Oriana Salazar Thula, Niklas Khoss, and Manuel Zimmer. Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. *Neuron*, 105 (3):562–576, 2020.

B Katz. Quantal mechanism of neural transmitter release. *Science*, 173(3992):123–126, 1971.

S. Klampfl and W. Maass. Emergence of dynamic memory traces in cortical microcircuit models through STDP. *J. of Neuroscience*, 33(28):11515, 2013.

Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

P. Kokotovic, H.K. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control Analysis and Design*, volume **25** of *Classics in Applied Mathematics*. SIAM, Philadelphia PA, 1999.

C. Kuehn. *Multiple Time Scale Dynamics*. Springer, Cham, Switzerland, 1st edition, 2015.

L. Kusmierz, T. Isomura, and T. Toyoizumi. Learning with three factors: modulating hebbian plasticity with errors. *Current Opinion in Neurobiology*, 46:170–177, 2017.

Connor Lane, Shipra Kanjlia, Akira Omaki, and Marina Bedny. "visual" cortex of congenitally blind adults responds to syntactic movement. *Journal of Neuroscience*, 35(37):12859–12868, 2015.

E. W. Large and C. Palmer. Perceiving temporal regularity in music. *Cognitive Science*, 26:1–37, 2002.

R. Legenstein and W. Maass. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3):323–334, 2007.

D. Linaro, A. Champneys, M. Desroches, and M. Storace. Codimension-two homoclinic bifurcations underlying spike adding in the Hindmarsh-Rose burster. *SIAM Journal on Applied Dynamical Systems*, **11**(3):939–962, 2012.

J. Lins and G. Schöner. Neural fields. In S. Coombes, P. beim Graben, R. Potthast, and J. Wright, editors, *Neural fields: theory and applications*, pages 319–339. Springer Verlag, 2014.

Michael A Long, Dezhe Z Jin, and Michale S Fee. Support for a synaptic chain model of neuronal sequence generation. *Nature*, 468(7322):394–399, 2010.

M. Lukoševičius, D. Popovici, H. Jaeger, and U. Siewert. Time warping invariant echo state networks. IUB Technical Report 2, International University Bremen, 2006. URL `https://www.ai.rug.nl/minds/uploads/techreport2.pdf`.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002. http://www.cis.tugraz.at/igi/maass/psfiles/LSM-v106.pdf.

H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.

C. Minnai, M. Mirigliano, S. A. Brown, and P. Milani. The nanocoherer: an electrically and mechanically resettable resistive switching device based on gold clusters assembled on paper. *Nano Futures*, 2:011002, 2018.

C. Mira and A. Shilnikov. Slow-fast dynamics generated by noninvertible plane maps. *International Journal of Bifurcation and Chaos*, **115**(11):3509–3534, 2005.

S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Transactions on Biomedical Circuits and Systems*, 12(1):106–122, 2018.

R.M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dpt. of Computer Science, University of Toronto, 1993.

E. O. Nefti and B. B. Averbeck. Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, 1(3):133–143, 2019.

Won Chan Oh, Laxmi Kumar Parajuli, and Karen Zito. Heterosynaptic structural plasticity on local dendritic segments of hippocampal ca1 neurons. *Cell reports*, 10 (2):162–169, 2015.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

R.E. O'Malley, Jr. *Singular Perturbation Methods for Ordinary Differential Equations*, volume **89** of *Applied Mathematical Sciences*. Springer-Verlag, New York NY, 1991.

G. I. Parisi, R Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

R. Pascanu and H. Jaeger. A neurodynamical model for working memory. *Neural Networks*, 24(2):199–207, 2011. DOI: 10.1016/j.neunet.2010.10.003.

M.E. Payvand, M. abnd Fouda, F. Kurdahi, A. Eltawil, and E.O. Neftci. Error-triggered three-factor learning dynamics for crossbar arrays. In *Proc. 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 218–222, 2020.

Jean Piaget. *The origins of intelligence in children*. International Universities Press New York, 1952.

D. Prychynenko, M. Sitte, K. Litzius, B. Krüger, G. Bourianoff, M. Kläui, J. Sinova, and K. Everschor-Sitte. Magnetic skyrmion as a nonlinear resistive element: A potential building block for reservoir computing. *Phys. Rev. Applied*, 9:014034, 2018.

K. Pusuluri, H. Ju, and A.L. Shilnikov. Chaotic dynamics in neural systems. In R.A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, page : to appear. Springer Science, 2020.

Wilfrid Rall. Rall model. *Scholarpedia*, 4(4):1369, 2009.

Rajnish Ranjan, Georges Khazen, Luca Gambazzi, Srikanth Ramaswamy, Sean L Hill, Felix Schürmann, and Henry Markram. Channelpedia: an integrative and interactive database for ion channels. *Frontiers in neuroinformatics*, 5:36, 2011.

D. Roclin, O. Bichler, C. Gamrat, S. J. Thorpe, and J. O. Klein. Design study of efficient digital order-based STDP neuron implementations for extracting temporal features. In *Proc. of The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2013.

J.E. Rubin and D. Terman. Chapter 3: Geometric singular perturbation analysis of neuronal dynamics. In B. Fiedler, editor, *Handbook of Dynamical Systems*, volume **2**, pages 93–146. Elsevier, 2002.

S. Ruschel. *Multiple Time-Scale Delay Systems in Mathematical Biology and Laser Dynamics*. PhD thesis, Technical University of Berlin, 2020.

A. Saffiotti, K. Konolige, and E.H. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76:481–526, 1995.

F. Schönfeld and L. Wiskott. Modeling place field activity with hierarchical slow feature analysis. *Frontiers in Computational Neuroscience*, 9:51, 2015.

Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J. Steil, and Dirk Stroobandt. Improving reservoirs using intrinsic plasticity. *Neurocomputing*, 71: 1159–1171, 1 2008.

F. C. Sheldon, A. Kolchinsky, and F. Caravelli. The computational capacity of memristor reservoirs. Arxiv manuscript, 2020. URL `https://arxiv.org/abs/2009.00112v2`.

Harel Z Shouval, Mark F Bear, and Leon N Cooper. A unified model of nmda receptor-dependent bidirectional synaptic plasticity. *Proceedings of the National Academy of Sciences*, 99(16):10831–10836, 2002.

H.T. Siegelmann and E.D. Sontag. Analog computation via neural network. *Theoretical Computer Science*, 131(2):331–360, 1994.

J. Sima and P. Orponen. General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation*, 15(12):2727–2778, 2003.

Christine A Skarda. The perceptual form of life. *Journal of Consciousness Studies*, 6 (11-12):79–93, 1999.

S. Song, K. D. Miller, and L. F. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, 2000.

Kenneth O Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, 2019.

T. Strauss, W. Wustlich, and R. Labahn. Design strategies for weight matrices of echo state networks. *Neural Computation*, 24(12):3246–3276, 2012.

A. Subramoney, F. Scherr, and W. Maass. Reservoirs learn to learn. In K. Nakajima and I. Fischer, editors, *Reservoir Computing: Theory, Physical Implementations and Applications*, chapter 1.3. Springer, 2021. Preprint arXiv:1909.07486.

R. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. Cambridge: MIT press, 1998. online version at `http://incompleteideas.net/book/ebook/the-book.html`.

G. Tanaka and et al. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019. preprint in `https://arxiv.org/abs/1808.04962`.

D. Terman. Chaotic spikes arising from a model of bursting in excitable cell membranes. *SIAM Journal of Applied Mathematics*, **51**(5):1418–1450, 1991.

J. C. Thiele, O. Bichler, and A. Dupret. Event-based, timescale invariant unsupervised online deep learning with STDP. *Frontiers in Computational Neuroscience*, 12:article 46, 2018.

Niko Tinbergen. *The study of instinct.* Clarendon Press/Oxford University Press, 1951.

J. Torrejon, F. A. Riou, M. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(27 July):428–431, 2017.

Marlieke TR Van Kesteren, Dirk J Ruiter, Guillén Fernández, and Richard N Henson. How schema and novelty augment memory formation. *Trends in neurosciences*, 35 (4):211–219, 2012.

J. van Leeuwen and J. Wiedermann. Beyond the turing limit: Evolving interactive systems. In *International Conference on Current Trends in Theory and Practice of Computer Science*, number 2234 in LNCS, pages 90–109. Springer, 2001.

A.B. Vasil'eva and V.M. Volosov. The work of Tikhonov and his pupils on ordinary differential equations containing a small parameter. *Russian Mathematical Surveys*, **22**:124–142, 1967.

F. Verhulst. *Methods and Applications of Singular Perturbations: Boundary Layers and Multiple Timescale Dynamics*, volume **50** of *Texts in Applied Mathematics*. Springer, New York NY, 2005.

Chr Von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100, 1973.

Karl Von Frisch. *The dance language and orientation of bees.* Harvard University Press, 1967.

Matthew P Walker and Robert Stickgold. Sleep, memory, and plasticity. *Annu. Rev. Psychol.*, 57:139–166, 2006.

L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

F. wyffels, J. Li, T. Waegeman, B. Schrauwen, and H. Jaeger. Frequency modulation of large oscillatory neural networks. *Biological Cybernetics*, 108:145–157, 2014.

Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLOS Computational Biology*, 4(11):e1000220, 2008.

S. Yanchuk and G. Giacomelli. Spatio-temporal phenomena in complex systems with time delays. *Journal of Physics A: Mathematical and Theoretical*, **50**:103001, 2017.

A. Yousefzadeh, E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco. On practical issues for stochastic STDP hardware with 1-bit synaptic weights. *Frontiers in Neuroscience*, 12(October):Article Nr 665, 2018.