Efficient online learning with low-precision synaptic variables

Marcus K. Benna and Stefano Fusi
Center for Theoretical Neuroscience, Columbia University, New York, USA
mkb2162@columbia.edu, sf2237@columbia.edu

***Abstract*—Current neuromorphic devices suffer from major limitations in their ability to perform on-chip online learning. These limitations often derive from their poor memory capacity, which is due to the low precision of the variables representing the synaptic weights. Here we present simple constructions of synaptic models with low-precision dynamical variables that can continually store and preserve a large number of memories, which grows almost linearly with the number of synapses per neuron. In addition, the initial memory strength, which is related to the generalization ability of the network, is also high in these models, and scales approximately like the square root of the number of synapses. These favorable properties are obtained by orchestrating multiple interacting processes that operate on different timescales, to ensure the memory strength decays as slowly as the inverse square root of the age of the corresponding synaptic modification. This decay curve achieves an optimal compromise between large memory strengths and long lifetimes. We discuss digital implementations of such synapses suitable for neuromorphic hardware. They are efficient in the sense of requiring only a small number of bits per synapse, and respond robustly to auto-correlated sequences of synaptic modifications.**

Keywords—synaptic plasticity, online learning, synaptic memory consolidation, neuromorphic hardware

I. Introduction

In the biological brain, synaptic memory consolidation following one-shot learning relies on a complex network of highly diverse biochemical processes. In contrast to this complex structure of real synapses, the synapses used in machine learning applications are typically represented by a single high-precision variable for each weight. For hardware implementations of neural networks, however, energy efficiency considerations and design constraints typically limit the precision of the synaptic weights, which often are binary (see e.g. [1,2]). This limited precision can somewhat reduce the inference (prediction) performance of neural networks, but it constitutes a much more severe problem for online learning in such networks. Here we focus on the latter issue, and discuss strategies to circumvent it by designing more complex synapses that augment the low-precision weight by additional, internal states. We hope that this note will provide some useful guidance on engineering plastic synapses suitable for neuromorphic hardware that supports local online learning.

This work was supported by NSF's NeuroNex program award DBI-1707398, the Simons Foundation, the Gatsby Charitable Foundation, the Swartz Foundation and the Kavli Foundation.

A. Problem setting: local, one-shot learning

The problem of online learning with low-precision weights is most easily illustrated by rephrasing it as a memory performance limitation of a set of N synapses exposed to an ongoing sequence of ever new memories (input patterns), each of which can induce plasticity events. Each input pattern is a vector of potential synaptic modifications (computed from the neural activity according to some learning rule), which would optimally store the corresponding memory. As an example, we can consider one-shot learning of a sequence of training examples each of which modifies the synapses of a perceptron architecture in a classification task, which requires hetero-associative memory. To quantify the memory performance we compute a signal to noise ratio (SNR) for the retrieval of a learned pattern as a function of the number of memories that have been stored since the one in question [3], which is closely related to the generalization ability of the network around the learned patterns (see [4] for details). The relevant quantities to consider are the initial SNR, which tells us how strongly new inputs are encoded, and the memory capacity, which is the number of subsequently stored memories before the SNR drops below the retrieval threshold (of order one), and the memory can no longer be recalled (or correctly classified). We are particularly interested in the scaling of these quantities with the number N of synapses connected to our output unit, since we want to ensure good scalability to large system sizes.

Memory capacities of e.g. the classical perceptron or auto-associative networks are known to grow linearly with N, but the derivations of these results assume (effectively) unbounded synapses (with a number of distinguishable states at least of order $\sqrt{N}$). Furthermore, these capacities refer to the transient memory performance after starting out the system from a special initial state of zero (or small) weights. In contrast to this, we focus on synaptic models for which memories can be overwritten an arbitrary number of times, as in continual learning. The memory capacity we are interested in refers to the average number of patterns that can be recalled in the steady state that is reached when a very large number of memories have been stored. In this state the weight distribution no longer changes (for constant input statistics), and the oldest memories are gradually forgotten to make room for new ones (palimpsest property). The steady state capacity is typically lower than the transient one starting from the tabula rasa initial state. Crucially, these steady state models don't suffer from a blackout catastrophe that would wipe out all memories at once.

In the simplest case of binary synapses, it was recognized long ago [3] that for online learning using local update rules, the capacity would grow at best as fast as $\sqrt{N}$ for very rigid synapses (that have a small SNR), and in fact only as log(*N*) for very plastic synapses (that show good generalization). This would be disastrous for large systems, which is why our primary concern is to improve this scaling behavior with *N*. Of course, if the learning occurred offline, or if the hardware system in question had access to large amounts of external memory that could be used to store intermediate results of the learning procedure (and thus didn't have to compute updates purely locally), the limitation of low-precision variables would be less pressing. Indeed, even binary synapses can achieve a memory capacity that scales linearly with *N*, albeit with a slightly smaller coefficient, if the binarization is performed only after learning is complete (which again requires temporary storage of higher precision weights during learning).

Below, we will introduce a number of simple synaptic memory models with limited precision weights that exhibit excellent scaling behavior during online (one-shot) learning, and are suitable for a digital implementation using binary variables. We will consider event-driven dynamics, such that weights and internal states of synapses are updated only when new memories stored. Any mention of timescales on which variables change or memories decay is understood to refer to a discrete number of time steps demarcated by the intervening plasticity events. Continuous, physical time will not play any role, except in that we will assume that the binary components used to implement the synaptic dynamics are sufficiently stable on the longest memory timescales that we wish to achieve. If this is not the case, e.g. if a binary element spontaneously switches states at a rate that is higher than the inverse of the desired forgetting timescale, several such elements will have to be combined with error-correcting interactions in order to implement one sufficiently stable effective binary variable.

B. Optimal online learning with complex synapses

Recently we showed that by augmenting a low-precision (possibly binary) synaptic weight by internal (hidden) low-precision variables, the resulting complex synapse model can achieve excellent scaling properties if its SNR decays as a $t^{-1/2}$ power law with the age *t* of the synaptic modification [4]. In this case, the memory capacity grows as $N/\log N$, and the initial SNR scales as $\sqrt{N/\log(N)}$. These scaling properties each differ only by logarithms from the best possible power-law growths that can be achieved for these two quantities separately (at the expense of substantially reducing the other), and represent an optimal compromise between the two.

Our goal here is to build simple models of steady state synapses that achieve the same scaling properties, and exhibit an approximate $t^{-1/2}$ power-law decay, using only a small number of binary variables. The biological model of [4] can be fully discretized, such that it has a finite number of states corresponding to a joint assignment of one of the allowed values for every one of the discrete physical variables, which could trivially be represented by several binary variables each. However, such a representation would not be very efficient in terms of the number of binary variables needed. Also, it would require stochastic dynamics with small transition probabilities, which is difficult to engineer.

In [5] the internal dynamics of the synapse was described by a Markov chain with *M* states. These authors optimized certain measures of memory performance over all possible transition matrices for a binary value of the synaptic weight assigned to each state. In particular, they derived an area bound on the integral under the (linear plot of the) SNR curve, which they showed cannot exceed $\sqrt{N}(M-1)$ for any such Markov chain model. Since the SNR has to be above threshold for a memory to be recalled, this implies that also the memory capacity is bounded by $\sqrt{N}(M-1)$ times a constant. The models that saturate these bounds are multi-state models (with binary weight readout) consisting of states connected in a linear chain topology. However, such models have small initial SNR, and when they achieve the longest possible memory lifetimes they are pathologically rigid, barely encoding new memories at all (due to small transition probabilities away from the end states). This is at least partially due to the fact that the area maximized also counts regions where the SNR is already below the retrieval threshold (of say one). It also indicates that one may want to choose a different optimality criterion (cost function) instead, such as e.g. the area under the doubly logarithmic plot of the SNR versus the number of memories, which is maximized by the $t^{-1/2}$ power-law decay [4].

Nevertheless, we will use the deterministic version of the multi-state model as a starting point for our heuristic constructions of synaptic models, since we know that it at least comes close to optimal efficiency in the sense of [5]. We limit ourselves to deterministic models, because they are easy to implement, and focus on simple rules that operate on a state space represented by binary variables. It is likely that generalizing to stochastic models would allow for somewhat more efficient implementations, but the results of [5] suggest that the gains that can be derived from this step might not justify the additional difficulties in building such synapses.

C. Multi-state model

The multi-state model consists simply of a finite precision synaptic weight variable (with a limited number of possible values) without any additional hidden states [6]. We can consider the allowed values to be equally (say integer) spaced, and arranged symmetrically around zero (which is why we will refer to the middle of the dynamical range simply as zero). If we assume for simplicity that the sequence of inputs is binary with equal step sizes and equal probabilities for potentiation and depression (such that the input is trivially balanced), each synaptic modification will increase or decrease the value of the weight by one unit, except when the plasticity event would take the weight outside of the (hard) bounds at the upper/lower end of the dynamical range, in which case it will induce no change.

If the model is implemented using *b* binary switches (bits) the number of states will be $M = 2^b$, and for balanced inputs the resulting decay of the SNR will be exponential with a timescale (memory lifetime) of order $M^2 = 2^{2b}$. Intuitively, this is because the synaptic weight executes an unbiased random walk under such balanced inputs, and the time until its state hits one of the bounds grows as the square of the number of steps required. The equilibrium distribution is uniform across the set of states. A potentiation event would simply shift it upwards by one step (a fraction $1/M$ of its dynamical range), leading to a distribution that again looks flat except at the

boundary states [6]. For the binary readout version of this model mentioned in the previous section, all states above zero are considered potentiated and assigned a weight of say one, and similarly all states below zero are considered depressed and assigned weight minus one. In this case, a fraction $1/M$ of synapses will have their internal variable cross zero, and therefore change their synaptic weight when a potentiation/depression event is applied to the equilibrium distribution. For both readouts of the synaptic weight, the initial SNR of the multi-state model is small, of order $\sqrt{N}/M$.

For any of the models discussed below, if hardware requirements force the synaptic weight implemented on chip to be binary (as e.g. in crossbars), one can construct such a binary readout, and retain the original low-precision weight as an internal variable that changes in response to synaptic plasticity events as before, but influences the actual synaptic weight used during inference only through the binary readout. These binary weight versions of our models will exhibit slightly lower memory performance, but the same scaling behavior with N (see Supp. Note 5 and Fig. S4 of [4]).

II. Results

Here we describe simple constructions of synaptic models, and compare them in terms of the number of binary variables required to implement them, and their memory performance.

A. Multiple chains: Combining multi-state models

We can use multi-state models as building blocks for models with better generalization performance, and since we think of their exponentially large number of states as being implemented through binary encoding by a chain of binary switches, we refer to them simply as chains. In order to enhance its initial SNR and endow it with an approximate power-law decay, we can augment a multi-state model of a sufficiently large memory capacity (i.e., long forgetting timescale) with shorter chains. These consist of fewer binary variables (and fewer states), and therefore exhibit a smaller memory capacity in isolation. However, they also have a larger initial SNR, and we can judiciously combine the advantages of both large and small models by defining a joint readout.

Since we would like the model's SNR curve to approximate a power law, we choose the forgetting timescales of the different chains to be uniformly spaced on a logarithmic scale (such that the ratio of successive timescales will be a constant). This means that the number of bits required for each chain will differ by a constant between successive chain. The design tradeoff in choosing this constant is that for small values we will obtain a very good approximation to a power law because the timescales are closely spaced, but at the price of using many bits, while for a larger value the approximation becomes worse, but can be more efficiently implemented. We will consider sets of chains that differ in length by two bits, which we found to be a good compromise, though other choices are of course possible. This implies that successive timescales will increase by a factor of 16 for balanced inputs (but only by a factor of 4 for strongly unbalanced ones). If the longest timescale chain consists of $b=2m$ bits (for integer m), this also means that there will be m chains in total.

Because the dynamical ranges of the constituent chains are quite different, combining them to obtain the joint readout that will define our synaptic weight requires normalizing each one by its corresponding number of possible states, so that they can be compared on an equal footing. Even simpler, since we know that a binary readout works well for an individual multi-state model, it is sufficient in practice to add the signs of the contributions of the different chains to compute the synaptic weight as $w = \sum_i \mathrm{sign}(w_i)$, where w_i denotes the state of the ith chain. Since the initial values and timescales of the exponential SNR curves of the different chains are proportional to $1/M$ and M^2, respectively, this leads to a good approximation of a $t^{-1/2}$ decay, as shown in Fig. 1. The number of distinguishable values of w equals the number of different timescales plus one (i.e., $m+1$). If a binary weight is required, we can define it by again taking the sign of this w (which is unambiguous for odd m) without substantially reducing the memory performance.

Note that a synaptic plasticity event will alter the state of each of the chains in the usual manner, and in this sense we can think of them almost as parallel synapses connecting the same pair of neurons. However, crucially the actual synaptic weight used for inference/recall is summarized in a single (possibly binary) number at any point during the ongoing learning process. Combining different timescale chains in this way is similar to combining populations of binary synapses with different switching probabilities (see Supp. Note 9 of [4]), but it naturally leads to a $t^{-1/2}$, rather than a t^{-1} decay of the SNR.

While this multi-chain model exhibits the desired power-law decay, and the associated good scaling properties for both the initial SNR and the memory capacity, its implementation is unfortunately not very efficient. If we compare it to a $2m$ bit multi-state model with the same longest timescale of order 2^{4m}, we require $2m-2$ bits for the second slowest chain, $2m-4$ for the next one, and so on down to two bits for the fastest chain. This adds up to a total of $m(m+1)$ bits, which grows quadratically with the logarithm of the longest timescale (that limits the memory capacity). For large m this is worse than the number of bits required for (the fully discretized version of) the biological model of [4], with variables of the same timescales represented using binary encoding. Nevertheless, this model is a useful intermediate step, which we can build on to find more efficient ones that reduce the number of bits required such that it grows only linearly with the logarithm of the maximal memory capacity, which is the best-case scenario [5].

B. Single chain with direction markers

Consider again a chain of $b=2m$ bits performing binary encoding of a variable with $M=2^b$ possible (integer-spaced) values, accumulating inputs as in the multi-state model. We can artificially divide it up into m groups of two bits each, arranged from left to right to represent low to high powers of two (see Fig. 2a). The first group corresponds to a set of four states, and if we keep potentiating the synapse, thus adding to its weight, these four states are traversed in a cyclic order. In particular, whenever the state changes beyond the largest representable value it is reset to the smallest possible value, with the provision that at the same time we increment by one the state of the next group (which therefore changes on a longer timescale). Similarly, for a sequence of depression events the states are traversed in the opposite direction, with the reset from the smallest to the largest allowed value accompanied by a decrement of the next group of bits.

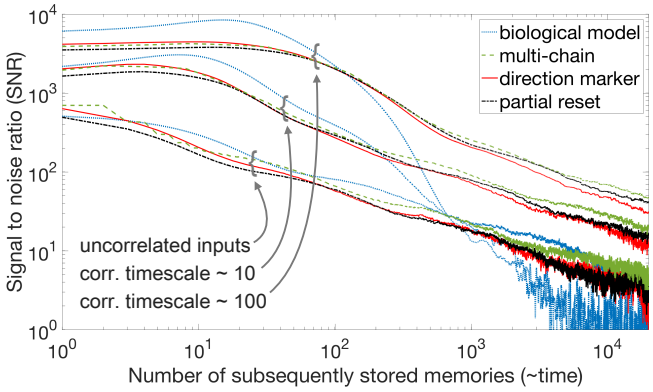

Fig. 1. Signal to noise ratio for ideal observer memory retrieval in a population of $N=10^7$ synapses as a function of the number of memories (stored since the one whose SNR is plotted) for four models with seven approximately matched timescales: The biological model of [4] with 32 levels per variable (dotted), the multi-chain model (dashed), the direction marker model (solid), and the partial reset model (dash-dotted). For each model three SNR curves are shown: One for uncorrelated synaptic modifications (bottom), in which case all models approximate a $t^{-1/2}$ power-law decay, and two for correlated inputs with correlation timescales of $-1/\log(0.9)$ (about 10; middle) and $-1/\log(0.99)$ (about 100; top). While the biological model doesn't respond well to correlated inputs, the other three models resume their power-law decay beyond the correlation timescale. Note that for uncorrelated inputs the first four variables would be sufficient to cover the range of timescales shown, but more are needed for correlated inputs.

Because of this reset of the shorter timescale variables that occurs in binary encoding whenever longer timescale variables are changed, the current value of the shorter timescale group of bits is not informative about the recent history of synaptic modifications, which is why in the multi-state model the SNR decays exponentially (with only a single timescale). However, we can capture the trend of the accumulated recent inputs (averaged over a certain time interval) by augmenting the short timescale group of bits by an additional marker bit. Whenever the state of this variable crosses zero, it keeps track of the direction of motion, i.e., whether the input that caused the zero-crossing was a potentiation or depression event, in which case the direction marker is set to one or minus one, respectively. Resets also change the sign of the variable, but are ignored by the marker bit, which is why the information it encodes is independent from the variable being above/below zero.

We can repeat this construction of adding a binary direction marker for every group of bits (i.e., for every timescale). For the last group we have a choice: We can treat it as in the multi-state model, where it has hard bounds that no plasticity event can overcome, in which case the relevant readout on this longest timescale is simply the last bit (the sign of the slowest variable). Alternatively, we can turn the last group of bits into another cyclic variable by allowing the dynamics to wrap around, in which case we also have to add a marker bit for this variable to keep track of the direction of motion on the longest timescale. The reason why one might want to do this at the expense an additional bit per synapse is that in this case the equilibrium distribution is flat not just for balanced, but also for unbalanced inputs, which can enhance the robustness of the model to auto-correlated inputs, as discussed below.

Constructing a joint readout proceeds again by simply adding up the marker bits for the different timescale groups. This leads to a good approximation to the desired $t^{-1/2}$ decay of

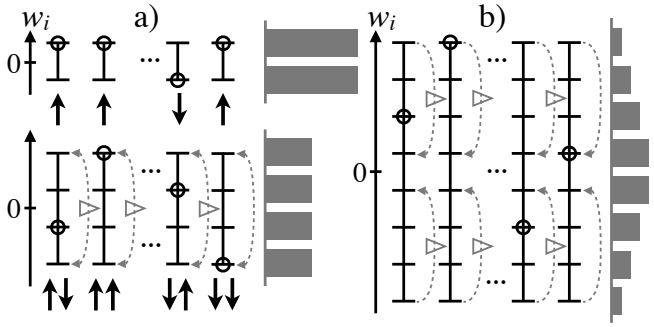

Fig. 2. Schematics of the direction marker (a) and partial reset models (b). The former consists of $2m$ bits performing binary encoding of the cumulative input (bottom), divided up into m groups of two bits each, whose states vary on increasingly longer timescales from the left to the right. Whenever a variable would exceed its bounds, it is reset to the opposite boundary value. If and only if this happens the input will also change the following variable by one unit. Each variable has an additional marker bit (top), that records the direction of its last crossing of zero (not counting resets). The final synaptic weight is the sum of these direction markers. The marginal equilibrium distributions of all variables and markers are uniform. In the partial reset model the three bits per group are combined into one variable with eight states, and the reset occurs towards the middle of the dynamical range. The synaptic weight is given by the sum of the signs of these variables. Their equilibrium distributions are triangular (for balanced inputs). In both models the reset of the last variable can be omitted, which would alter its distribution.

the SNR (see Fig. 1). This larger SNR compared to the multi-state model has been achieved at the cost of adding one bit per timescale, in our case for every group of two bits of the original chain. For the same longest timescale, the implementation of this model (with m different timescales) thus requires $3m$ instead of $2m$ bits, which still grows only linearly with the logarithm of the maximum memory capacity, instead of quadratically as for the multi-chain model. We have thus obtained a rather efficient construction: E.g. for a synapse with $m=7$ timescales (as in Fig. 1) this model requires 21 bits of storage, whereas the multi-chain model with the same timescales needs 56 bits, and the biological model with 32 distinguishable values per variable uses 35 bits (and requires stochastic interactions with small transition probabilities).

C. *Partial reset model*

Another model that exhibits similar memory performance and efficiency can be obtained by a further modification of the direction marker model. In this model, the marker bits and the bits performing binary encoding are no longer independent, but instead we combine them into groups of (with our choice of timescale ratio) three binary switches, representing different timescale variables with eight states each (see Fig. 2b).

As above, these variables implement a counter for the total cumulative input, but instead of using standard binary encoding, which resets each variable to the opposite end of its dynamical range when its value would exceed one of its bounds, here we instead stipulate that its value is reset to the state closest to zero that has the same sign. Since the reset takes the state of the variable close to the middle of its dynamical range, we refer to this as the partial reset model. As before, the reset is accompanied by an increment/decrement of the next variable by one unit, which again corresponds to a change of four units in the previous variable. Because of this partial reset, the state of each variable is correlated with the recent direction

of motion (averaged over the appropriate timescale). E.g. if a variable has the largest possible value, we know for certain that it approached this value from below, since there is no reset to boundary values. We can therefore use a simple readout that adds the signs of all variables to determine the synaptic weight, as for the multi-chain model. This corresponds to summing the values of the last bit from each group.

The (marginal) equilibrium distributions of the different timescale variables are no longer flat, but in fact pyramid-shaped. This means that the states closest to zero, which are the most plastic in the sense that they can lead to a change in the synaptic weight in one step, have the highest occupancy, in contrast to the rigid chains of [5] that maximize their memory lifetime when only their least plastic states are occupied. As before, we have a choice whether or not to introduce a (in this case partial) reset for the last variable, thus turning a bounded variable into a cyclic one (here with two distinguishable cycles for repeated potentiation and depression). The number of bits required to implement this model (for approximately matched timescales) is the same as in the previous model, and the memory performance is also very similar (see Fig. 1).

D. Robustness to auto-correlated inputs

For the biology-inspired synaptic model of [4], net imbalances in the effective rates of potentiation and depression are problematic, because it operates with bounded variables that can be pushed towards the edges of their dynamical range by a non-zero drift in the synaptic modifications, which prevents the synapse from correctly accumulating further inputs. While this problem can be solved locally, e.g. by a homeostatic mechanism that subtracts a running estimate of the mean input (averaged over an interval longer than the longest timescale), there remains a secondary issue when the synaptic inputs generated by the learning rule are correlated. An auto-correlated sequence of desired plasticity steps, even though the mean subtraction renders it balanced in the long-term average, can look imbalanced in either direction on shorter timescales. E.g. for a simple exponential auto-correlation function, inputs will tend to align across intervals of the order of the correlation timescale. Such short-term imbalances can still saturate one of the bounded dynamical variables (especially the fastest one), leading to errors in the further accumulation of inputs, and we cannot cure this problem by simply subtracting a running average on this shorter timescale, since this would limit the memory lifetime by erasing longer-term memory traces.

Fortunately, the direction marker and partial reset models use cyclic variables that by construction cannot get stuck in boundary states (except perhaps for the longest timescale variables, if one choses to not make them cyclic). Analogously, the multi-chain model incorporates long chains that correctly track inputs even when its shorter ones have hit their bounds. Therefore, these three models can deal much more gracefully with auto-correlated inputs, as shown in Fig. 1. Note that the overall higher magnitude of the SNR is simply due to successive memories being correlated, and is not indicative of a better memory performance in terms of information storage. The important feature of these SNR curves is that after an initial transient lasting about one correlation time interval, they again approximate a $t^{-1/2}$ decay, and don't exhibit a breakdown due to saturation that occurs in the biological model.

III. Discussion

We have presented simple deterministic models of complex synapses that can be efficiently implemented using digital, low-precision variables and may be suitable for neuromorphic hardware with on-chip online learning. These models exhibit close to optimal scaling behavior for both the initial SNR (leading to good generalization), and the memory capacity. We have described these models as generalizations of a multi-state model with a long timescale. Another, equivalent view is that rather than building independent subsystems to implement each of the different timescales of our synaptic model (as for the multi-chain model, which uses too many bits), we have introduced interactions between the different low-precision variables (groups of bits, which separately would each have only a small memory capacity), to build up longer and longer timescales. This effectively reuses some of the bits that would be required to implement long timescale chains in isolation.

Whereas in [4] the interactions between different timescale variables were bidirectional, here we considered simple feedforward inputs from shorter to longer timescale variables, combined with appropriate resets of the former. The reason bidirectional interactions were necessary in the biological implementation was that we insisted on the first variable alone interacting with the neural activity (i.e., it received the input and determined the synaptic weight), with the longer timescales being implemented by hidden variables. For designing neural hardware, however, it appears plausible that the final weight may depend on several variables.

An additional feature of the models presented here is their robustness to auto-correlated inputs, which should be very helpful in machine learning tasks using e.g. gradient descent, in which successive update steps are often highly correlated. This feature will be even more important in real-world tasks that require continual learning. Many aspects of the natural world are known to be characterized by power-law or heavy-tailed distributions. To learn from such naturalistic inputs, we will need artificial neural systems that can deal with synaptic modifications which exhibit correlations on all possible timescales. Future studies will test the proposed synapses in these kinds of real-world tasks, and we hope that this will help pave the way for the development of energy-efficient, autonomous learning systems.

Acknowledgment

We thank Rajit Manohar for the kind invitation to present this work at Asilomar.

References

[1] S. C. Liu, T. Delbruck, G. Indiveri, A. Whatley and R. Douglas (eds.), "Event-Based Neuromorphic Systems," John Wiley & Sons (2015).

[2] C. Mayr, S. Sheik, C. Bartolozzi and E. Chicca (eds.), "Synaptic Plasticity in Neuromorphic Systems," *Front. Neurosci.* Media (2016).

[3] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," *Neural Comput.* **6**, 957–982 (1994).

[4] M. K. Benna and S. Fusi, "Computational principles of synaptic memory consolidation," *Nat. Neurosci.* **19**, 1697–1706 (2016).

[5] S. Lahiri and S. Ganguli, "A memory frontier for complex synapses," *Adv. Neural Inf. Process. Syst.* **26**, 1034–1042 (2013).

[6] S. Fusi and L. F. Abbott, "Limits on the memory storage capacity of bounded synapses," *Nat. Neurosci.* **10**, 485–493 (2007).