# Synaptic caching to reduce energy consumption from learning

Balázs Ágoston Nyírő, James Read-Tannock and Reece Hill.

. School of Psychology, University of Nottingham, Nottingham, NG7 2RD, UK

## Abstract

Energy efficiency is a core source of evolutionary selection pressure, due to its importance for survival. Therefore, computational mechanisms in the brain such as synaptic plasticity are likely optimised to minimise energy consumption where possible. Li and van Rossum (Li and van Rossum, 2020) propose a model of the various forms of plasticity in which more transient forms of plasticity act to cache changes in synaptic weights, pending later consolidation. We reproduce computational models that demonstrate the virtues of this approach for reducing energy consumption in the learning process.

## 1 Introduction

Neuroplasticity is a broad term referring to structural and functional changes in the nervous system over the lifespan of an organism. In some cases, plasticity involves changes to brain connectivity arising from the growth of new neurons (neurogenesis), but more commonly changes in functional connectivity are thought to underlie learning and memory. Generally, this takes place by modifying the strength of the synaptic connections between neurons (although there is also evidence for non-synaptic forms of neuronal plasticity (Mozzachiodi and Byrne, 2010). This is known as long-term potentiation (LTP) and long-term depression (LTD) of synapses.

While much research into synaptic plasticity has focused on LTP and LTD of synapses at timescales of up to one hour (Clopath, 2012), much longer lasting plasticity is required for memories that are maintained for days or months. Plasticity over these different timescales is sometimes distinguished as early LTP (E-LTP) and late LTP (L-LTP) respectively. Experiments have shown that these forms of plasticity rely on distinct mechanisms, with L-LTP reliant on gene transcription and protein synthesis (Frey et al., 1988, Krug et al., 1984, Matthies et al., 1990, Nguyen et al., 1994, Otani et al., 1989), likely in order to produce new receptors and expand the synaptic bouton.(Abraham and Williams, 2003) Therefore, inducing L-LTP is likely to be more metabolically costly than E-LTP alone, which produces less enduring changes to the synapse, such as AMPA receptor phosphorylation and retrograde signalling to increase the probability of vesicle release from the presynaptic neuron. Similar considerations apply to LTD, which also requires protein synthesis for its late phase (Kauderer and Kandel, 2000, Linden, 1996).

At least two differing forms of consolidated memory can be induced in *Drosophila* fruit flies under different regimes of associative learning (Tully et al., 1994). Firstly, a massed training protocol of classical conditioning, with training cycles performed in rapid succession, produces a form of memory that does not appear to be reliant on protein synthesis and that decays rapidly to baseline over the course of a day or two. Secondly, a spaced training protocol with 15 minutes be-

tween training cycles causes a form of memory that is disrupted by protein synthesis inhibitors, but that otherwise does not decay for a week or more after training.

Neuroplasticity in the latter case, where protein synthesis is required, is known to be more metabolically costly than the plasticity induced by the massed training protocol. Mery and Kawecki (2005) showed that fruit flies subjected to the spaced training protocol described above starved to death 20% faster than flies subjected to the massed training protocol. Energy efficiency is a core source of evolutionary selection pressure for the brain (Niven and Laughlin, 2008), so it is likely that the different forms of neuroplasticity have evolved to operate in tandem to reduce the energy cost of learning.

A core factor increasing energy usage is the indirect route by which neural network parameters must converge towards optimal values. As the number of patterns to be classified by a given number of synapses becomes high, a single synaptic weight must encode information about multiple patterns simultaneously. Consequently, weight updates are frequently reversed and overwritten during training, and can be approximated as a random walk. If all changes to weights are made in the persistent form, this process will be costly. In the brain, this would correspond to the unnecessary production and transportation of plasticity-related proteins. Taken together with the evidence that forming long-term memory is metabolically costly, this suggests that temporary forms of plasticity help to minimise the costs of learning, by encoding a short term memory cache that can later be consolidated.

Li and van Rossum (2020) test the hypothesis that temporary forms of plasticity (E-LTP) can reduce the metabolic cost of learning by such a synaptic caching algorithm. To this end, they implement artificial neural networks with simulated energy costs associated with modifying weights. They split weights into transient and persistent components. For transient weights, the amount of energy expended is solely and directly proportional to their summed magnitude in each time step (cost of maintenance). However, these weights may decay exponentially to zero over time. To prevent losing what has been learnt, it is therefore important to consolidate transient weights into a form that is persistent. This alternative form is free to maintain, but incurs a cost whenever it is altered. Fine-tuning the threshold at which weights are consolidated is shown to greatly reduce the energy cost of learning[1], depending upon the rate of transient weight decay, the maintenance cost of transient weights, and the cost of consolidation. It is plausible that different paradigms could be true, perhaps where transient weights have both maintenance and alteration costs, and in general these alternatives reduce the effectiveness of synaptic caching to some degree, while leaving the general principle intact.

Additionally, Li and van Rossum show a trade-off between the speed of learning and the energy cost when the cost of consolidation and the rate of transient weight decay are both relatively large. This report is structured as follows. In Section 2 we describe the model of synaptic caching developed by Li and van Rossum, which we have reproduced. In Section 3, the results and key findings of the modelling work are recreated, and Section 4 concludes with some discussion of the relevance of these findings and possible future directions.

## 2 Single layer perceptron model

Li and van Rossum begin by using the perceptron learning algorithm (Rosenblatt, 1958) as a simple example of a neural network, mediated by synaptic weights for investigation of energy usage. Roughly

---

[1]Especially in the case where a neural network is close to its maximum memory capacity.

corresponding to a single neuron receiving inputs from its dendritic tree, the perceptron generates predictions by the scalar product of an input vector with its synaptic weights, with the resulting scalar value then input to a Heaviside step function to determine activation.

$$\hat{y} = \Theta(\mathbf{x}^T \mathbf{w}) \qquad (1)$$

To study the energy costs of learning, it is initially assumed that there is a constant, proportional cost associated with both positive and negative changes in synaptic weights, so that the total energy used in the learning process can be expressed as follows:

$$M_{perc} = \sum_{i=1}^{N} \sum_{t=1}^{T} |w_i(t) - w_i(t-1)|^{\alpha} \qquad (2)$$

This is simply the sum over the total time steps T, and the total number of synapses N of the absolute value of all weight changes, raised to the exponent $\alpha$. As a consequence, the energy cost of learning $M_{perc}$ will be proportional to the total distance travelled in weight space until convergence is reached or training is stopped.

As the number of patterns per synapse becomes high, the marginal energy cost of learning new patterns becomes higher, diverging as the ratio approaches 2 patterns to every synapse. This results from a narrower 'cone' of effective weight combinations in weight space, meaning that the training process takes longer to reach convergence. Not only does this occur for the actual energy used in training the perceptron, but also for the theoretical minimum energy required to reach the final weights (i.e., the L1 norm of the final weights vector). Also as the pattern:synapse ratio approaches 2:1, the ratio of actual energy to minimum energy also diverges, i.e., the learning algorithm becomes increasingly inefficient.
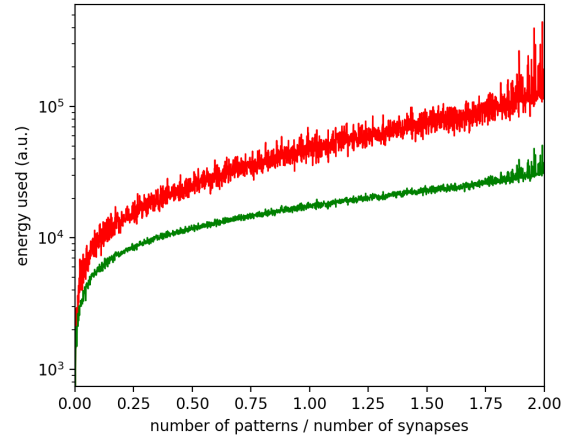


Figure 1: Energy used at different ratios of classified patterns to perceptron 'synapses', representing the density of learning required. The difference between the actual energy used (red line) and the theoretical minimum energy required to reach the final weights (green line) becomes greater and diverges as the ratio of patterns:synapses approaches 2:1.

When introducing the synaptic caching functionality, we are required to set a magnitude threshold at which the transient weights signal for their consolidation into persistent weights. By considering synapses as working together or independently, we create three ways in which consolidation can be implemented. If triggered by one synapse that exceeds the threshold, we term this a "local threshold". This may then either lead to local consolidation (only at the triggered synapse), or global consolidation where all synapses are simultaneously consolidated. The former case may be called local-local, and the latter local-global. In the third and final case, global-global, consolidation of all weights is triggered when the sum of synaptic weights exceeds the threshold.

These different algorithms produce comparable efficiency gains, but the local-global algorithm has the benefit of resembling the synaptic tagging and capture model of synaptic plasticity. Frey and Morris (1997) pro-

posed this mechanism, wherein L-LTP or L-LTD can be induced in a synapse that has not itself been sufficiently stimulated to trigger protein synthesis in the cell body. In this model, when early phase long-term plasticity occurs at a synapse, it becomes 'tagged'. If another synapse in the same cell has recently triggered the synthesis of proteins, the synapse may then 'capture' the proteins synthesised in the cell body to facilitate late phase plasticity.

The efficiency gains of synaptic caching can be illustrated by Figure 2, which shows the energy used by learning at different values of the consolidation threshold, where zero implies no synaptic caching (i.e., all weights are immediately consolidated). The key finding here is that there is an optimum consolidation threshold that minimises the total energy used in learning, where unnecessary consolidations are skipped but before reaching a level where maintenance of transient memory begins to cause excessive energy use. The location of this optimal threshold (minimum of black curve) depends greatly on the relative cost of consolidation and maintenance (see Figure 4), and furthermore on the rate at which transient weights decay. Higher rates of decay lead to training taking longer and hence more time for maintenance energy costs to accrue.

An interesting difference in the results of our modelling as compared to Li and van Rossum is that maintenance energy (and hence total energy) approaches a maximum value when consolidation energy approaches zero. This tapering off of the gradient is logical because when the consolidation threshold becomes so high that consolidation no longer occurs, alterations to the threshold will cease to affect energy usage. Figure 3 shows another non-linearity that occurs when the rate of transient plasticity decay is high. In the intermediate range of consolidation thresholds the gradient of the maintenance energy curve becomes much steeper, as
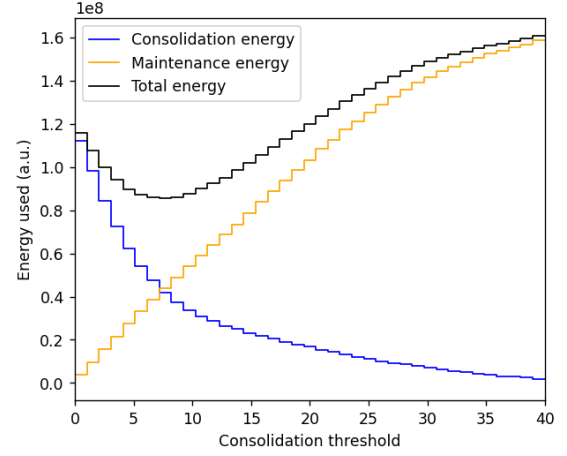


Figure 2: Plot of energy used for maintenance and consolidation of synaptic weights under the zero decay regime.
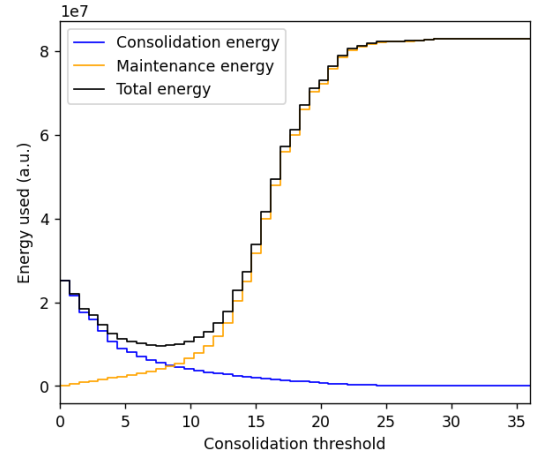


Figure 3: Plot of energy used for maintenance and consolidation of synaptic weights when a significant rate of decay is added to transient weights. The maintenance energy curve assumes a sigmoidal shape.

4

large weights do not reach the threshold due to rapid decay. Together with the aforementioned tapering off, this causes a sigmoid shaped maintenance energy curve.

Finally, Figure 4 shows the optimal threshold for consolidation in relation to the cost of transient plasticity. As maintenance costs for the transient plasticity increase, the optimal threshold lowers, eventually favouring almost constant consolidation as a way to avoid the energy cost of maintenance. In reality, the situation is probably less extreme in neurons, although E-LTP is known to increase synaptic energy usage as vesicle release and firing rates are increased.
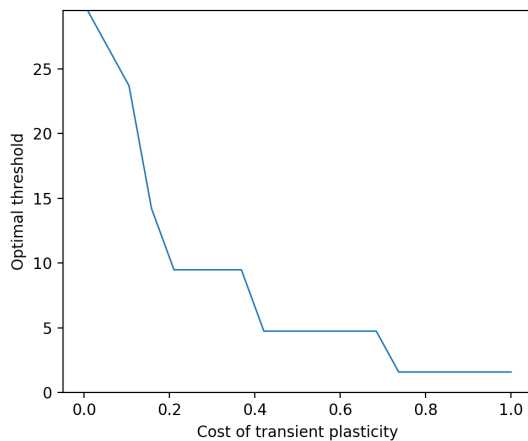


Figure 4: Plot of the optimal consolidation threshold in relation to the maintenance cost of transient synaptic weights.

## 3 Multi-layer perceptron model

The single layer perceptron is suitable for modelling the energy demand of an individual neuron; however, by using a multilayer perceptron we can take this further, exploring at a cell-system level. The network consists of an input, a hidden and an output layer. The number of hidden units in the middle layer is 100 by default, but this can be changed. For the input layer, sigmoid functions are used, and for the output layer a softmax function is applied. The MNIST database, which contains grayscale images of handwritten numbers, is used to train the network. There are 70,000 images, all 28 x 28 pixels. The database is first normalised so that each data point will fall between zero and one. Then the images are divided into train and test groups with a ratio of 60,000:10,000. Forward-propagation is performed on all training images one-by-one, and then back-propagation is performed after the errors are calculated. The weights are updated by Stochastic Gradient Descent. The weight changes are first written to a transient weight vector and then, in the absence of consolidation, are immediately written to the weight vector that symbolises persistent memory. It is important to note that although the program computes the sum of the initial, transient and persistent weights, these initial values and subsequent weight changes are stored separately throughout.

If there is no consolidation and no transient memory, weight updates are immediately stored in persistent memory. Furthermore, in this case, there is only the cost of maintaining persistent memory. In humans, long-range (Kattner et al., 2017) and short-range mechanisms (Jepma et al., 2016) have been observed to affect the speed of learning. The human body can accelerate the learning process by releasing norepinephrine and dopamine. In machine learning, this corresponds to changing the learning rate. The results of simulations (Figure 5) show that increasing the learning rate, while allowing faster learning within certain limits, strongly increases the energy demand of learning while not significantly changing its practical maximum accuracy. If the learning rate is too high, no substantial learning takes place, but the energy demand of the process increases rapidly. In biological terms, this may correspond to an overly heightened, potentially stressful state. From this, we conclude that time-efficient learning can be more costly for

5

the neuron than one where learning is slow. The ordering and run-up of our results show a different picture from that obtained by Li and van Rossum, 2020. This may be due to a difference in initial conditions, and the fact that our program needed more epochs to learn with a lower learning rate, yielding a higher total learning cost.
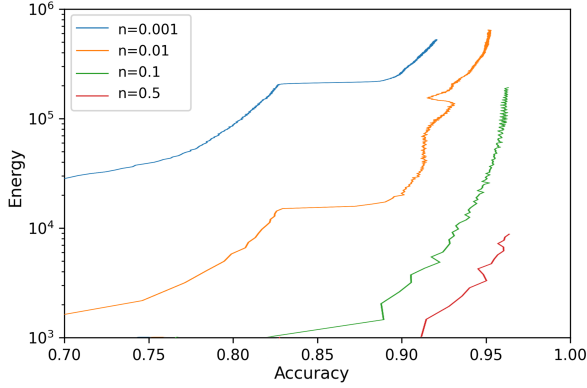


Figure 5: Total energy requirements of a multilayer perceptron as a function of achieved accuracy for different learning rates.

As discussed, a more energy-efficient approach to neuroplasticity could use synaptic caching. To simulate this, the program uses a transient weight vector. In this case, the weight changes are still written to the transient memory first, but are consolidated per unit time. Then, the cost of the transient and the memory is calculated (for the preceding time unit) and decay is applied with a predefined constant, a decay rate $\beta$ (equation 3).

$$w_i = w_i * e^{-\beta} \qquad (3)$$

The amount of energy that can be saved by synaptic caching depends largely on the decay rate and the maintenance cost of the transient memory. However, it can be seen that at lower learning accuracies the energy saved is significant, but towards the maximum achievable accuracy this rate decreases exponentially and approaches the value without caching. Meanwhile, the theoretical minimum energy requirement can be as much as 10-20% lower (Figure 6).
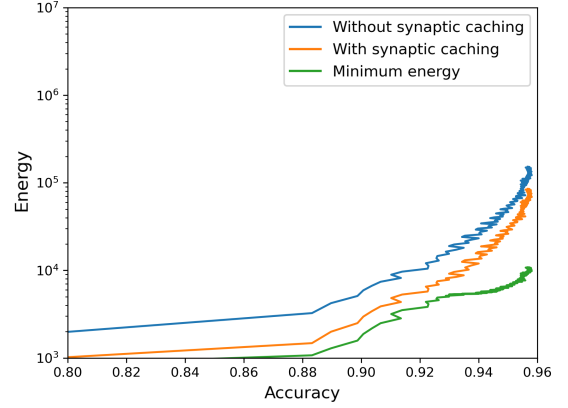


Figure 6: Total energy demand of a multilayer perceptron as a function of the achieved accuracy with and without synaptic caching.

A further way to reduce metabolic energy use is to reduce the number of hidden layer neurons. A low number of neurons greatly increases the cost of learning, just as more accurate learning can be costly for the organism. Few neurons learn a pattern much more slowly than a larger network. However, it can also be said that the total energy consumption increases almost linearly with the increase in the number of neurons. Synaptic caching also provides additional energy savings in this respect. It is important to mention, however, that the dynamics and ratios outlined above are strongly influenced by the properties of the database to be learned and the transfer functions used for the layers. Furthermore, when performing our own simulations, we ran the learning for fewer epochs so that, for example, the high energy consumption of a small number of hidden units did not appear in the graphs (Figure 7). See Section 4.2 for further discussion of this.
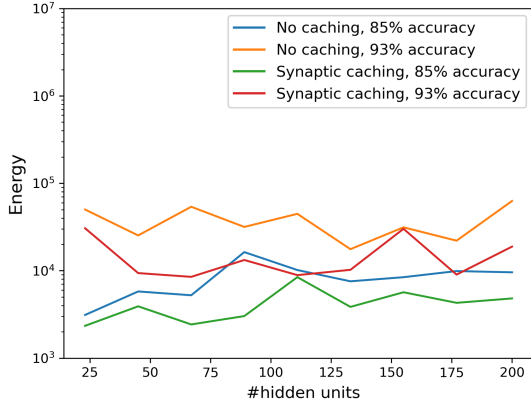
Figure 7: The total energy demand of a multilayer perceptron as a function of the number of hidden neurons in its hidden layer under different consolidation modes.

## 4 Discussion

### 4.1 Rewriting the perceptron - handling weight-related events

Li and van Rossum published the code they used to simulate the perceptron.(Li and van Rossum, 2020) However, this codebase is no longer functional and demands some rewriting to be compatible with today's software. In order to reproduce their findings, we therefore used our own implementation of the perceptron, developed independently and using Python version 3.10.2. Our source code is available on GitHub (single layer perceptron: https://github.com/reecehill/PBM-Project1.git, multilayer perceptron: https://github.com/nyirobalazs/multilayer_perceptron.git). In doing so, we recognised some processes of the perceptron where nuances could arise.

An example of which is the handling of weight-related events at each timestep. It stands that for any given epoch there are three processes that modify the values of weights: 1) the weights are subjected to decay; 2) weights are updated post-prediction; 3) transient weights beyond a threshold are consolidated and set to zero.

Clearly, the order in which these events occur is consequential; weights at time step, t, will hold different values if the order of processing were changed. For example, if a decay function, f, were applied to the weights upon starting an epoch, t, then predictions will be made based on the decayed weights.

$$\hat{y} = \Theta * (x^T * f[w(t)]) \qquad (4)$$

Alternatively, epochs ending with decay would use the weights as they arrived from the previous epoch, assuming consolidations are yet to occur.

$$\hat{y} = \Theta * (x^T * f[w(t-1)]) \qquad (5)$$

Given the iterative nature of the perceptron, for most instances the effect is minimal and we evolve the same weights, just offset by a timestep. Nevertheless, this offset combined with an epoch limit could lead to the loss of one timestep's results, yielding different values of total energy for the different processing orders. This is an unlikely scenario, and is easily mitigated by setting a maximum epoch well above what is required. Perhaps a nuance of greater detriment is that of consolidation, and when it occurs relative to the timestep in which its threshold is exceeded. If weights satisfy criteria for consolidation at timestep, t, two narratives could follow. First, we describe the process where consolidations occur in-place. Then, we propose an alternative, where weights are flagged for consolidation and processed in the next epoch.

It is plausible that weights are consolidated in the same timestep, as the authors' code shows to be the case. In this instance, a weight satisfying criterion for consolidation in timestep, t, would be rewritten to zero in the same timestep. Consequently, if we were to plot a vector of a single weight's value over time, we would never see that it exceeded the threshold; after all, when the weight

does so, it is set to zero (assuming a local-global caching algorithm is used). We cannot be certain that a weight at zero arrived at such through consolidation, or through the underlying random walk, either. We also theorise it could lead to an under-prediction of total energy that a neuron uses during the learning process. For when consolidation is computed in this way, the maintenance energy that may have been used to increment the transient weight from the previous epoch to the threshold is lost. Thus, we offer two solutions.

First, weights passing a threshold in time t, could be consolidated at the beginning of timestep, t+1. Assuming that the order of weight-related events are processed appropriately, this solves the aforementioned problem of plotting weights over time; weights above the threshold will now exist in the vector. However, dependence on changing the epoch and the order in which weight-related changes are processed, as proposed here, could lead to further problems. For instance, consolidating at the beginning of t+1, would nullify any effect of the proceding decay function on values that are consolidated (as they are now zero). Likewise, if decay were to be initiated first, how would one handle a decayed weight that now falls below the threshold for consolidation?

To this end, an alternative solution is one that begins with the author's. Weights are consolidated in the same timestep, t, as they passed the threshold. In addition however, we capture the value of the weights before they are consolidated. We find the difference between these and the weight values from the previous epoch to calculate the energy that was used to reach the threshold ($M_{pre}$). We then add this to the current notion of maintenance cost for transient weights to

form a "corrected" variant.

$$M_{pre} = c \sum_{i=1}^{N} \sum_{t=1}^{T} |s_i(t)| \qquad (6)$$

$$where\ s_i(t) \geq threshold$$

$$M_{trans} = c \sum_{i=1}^{N} \sum_{t=1}^{T} |s_i(t)| + M_{pre} \qquad (7)$$

Using the corrected maintenance cost is one attempt to overcome the shortfalls of a model that is discretized in time. Most biological processes are observed to be in equilibria with their inverse processes. The production of intracellular proteins may be countered by clean-up enzymes that ensure their destruction; a system's measured activity is continuous, often in units of concentration. If we reduce neuronal plasticity in a similar way, we see that the learning rate of our perceptron could, in theory, be applied gradually between each epoch. Perhaps, the changes required to elicit increased transient plasticity, such as altering the concentration of enzymes involved in protein production, occur over time in a neuron. It follows that a neuron will not know if a transient weight will meet a threshold - until it does. By which point, the neuron has expended energy to maintain the synapse's connectivity as it increased from its value at t, to t+1. Other ways to approach this issue could include adding an additional dimension or variables to the simulation, one that stores the weights' value after the three weight-related events that occur in each epoch. Though this would be at the expense of computer memory, and would require further mathematical inquiry to derive energy.

### 4.2 Limitations of our perceptrons

Throughout Section 2, we discussed the perceptron model as described by Li and van Rossum. However, due to limited access to
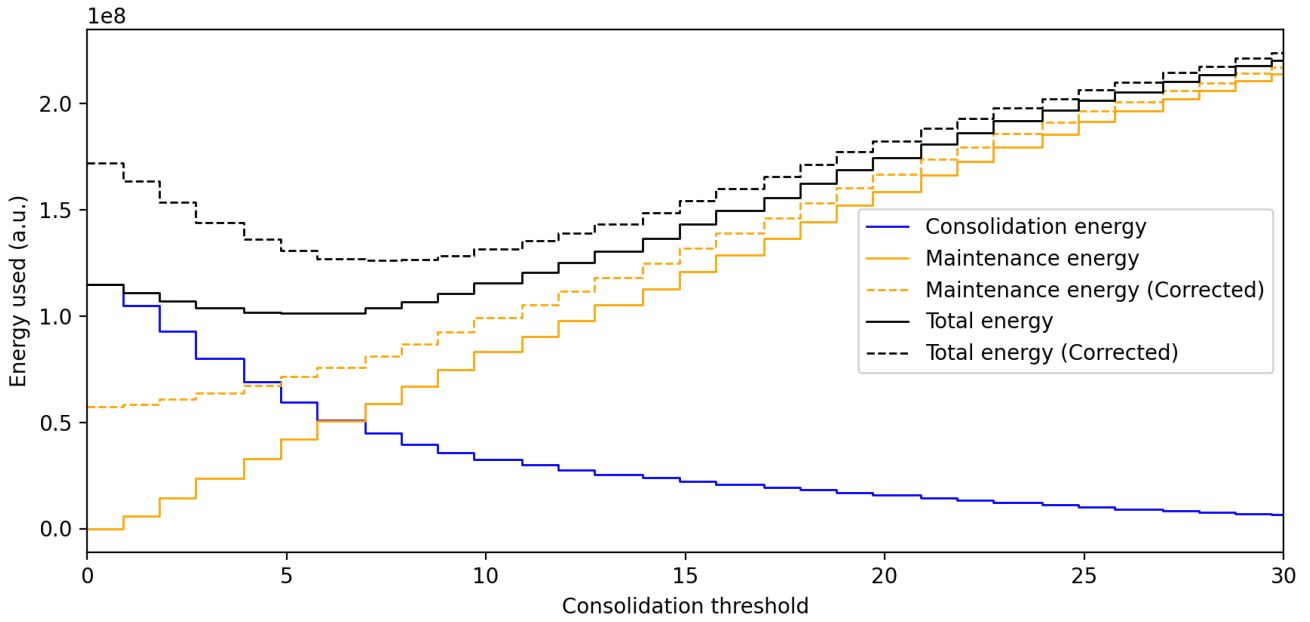
Figure 8: Illustration of the potential divergence in energy consumption depending on the order of consolidation and energy usage.

comparable computing power we were unable to replicate their results entirely. Despite our plots of energy revealing behaviour that is similar to their findings, we note that our energy expenditure is often to a lesser magnitude. The most probable cause is the size of patterns they task the perceptron to learn — a 1,000-element vector of negative and positive 1s. As we used computers intended for personal use, we were forced to limit patterns to 250 elements. Additionally, we decreased the maximum number of epochs per simulation from 10,000 to 2,500. Patterns are learnt in fewer epochs, and so we yield lower energy values. Nevertheless, similar qualitative results can be obtained from the smaller model, illustrating the benefits of synaptic caching across a range of model specifications. Similar remarks can also be made to explain the results of our multilayer perceptron.

## 5 Conclusion

In this work, we attempted to replicate the findings of Li and van Rossum. We found

similar results, though it is important to acknowledge that this work remains largely theoretical; it includes many assumptions of the neuron and hypothesises many quantified values of the neuron.

Further work, for instance, could be done to study the learning rate of an average neuron, for input into the perceptron model to further its biological plausibility. Similarly, exploring the effect of more than two memory types is needed. Scientific inquiry could expand from the current persistent (consolidated) and transient memory types, towards including a full range of memory types including anaesthesia-sensitive memory (ASM), anaesthesia-resistant memory (ARM) and long-term memory (LTM) and others (Tully et al., 1994). In doing so, the perceptron model will more accurately replicate the activity of a neuron. Such exploration is already feasible using our implementation of the perceptron, although these results were omitted in this report due to scope. In any case, computational modelling of energy efficiency in synaptic plasticity is a

promising area for further research into learning in biological brains.

## References

Abraham, W. C., & Williams, J. M. (2003). Properties and mechanisms of ltp maintenance [PMID: 14678579]. *The Neuroscientist*, *9*(6), 463–474. https://doi.org/10.1177/1073858403259119

Clopath, C. (2012). Synaptic consolidation: An approach to long-term learning. *Cogn Neurodyn*, *6*(3), 251–257. https://doi.org/10.1007/s11571-011-9177-6

Frey, U., Frey, S., Schollmeier, F., & Krug, M. (1996). Influence of actinomycin D, a RNA synthesis inhibitor, on long-term potentiation in rat hippocampal neurons in vivo and in vitro. *J Physiol*, *490*(Pt 3), 703–711. Retrieved March 10, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1158708/

Frey, U., Krug, M., Reymann, K. G., & Matthies, H. (1988). Anisomycin, an inhibitor of protein synthesis, blocks late phases of LTP phenomena in the hippocampal CA1 region in vitro. *Brain Res*, *452*(1-2), 57–65. https://doi.org/10.1016/0006-8993(88)90008-x

Jepma, M., Murphy, P. R., Nassar, M. R., Rangel-Gomez, M., Meeter, M., & Nieuwenhuis, S. (2016). Catecholaminergic Regulation of Learning Rate in a Dynamic Environment [Publisher: Public Library of Science]. *PLOS Computational Biology*, *12*(10), e1005171. https://doi.org/10.1371/journal.pcbi.1005171

Kattner, F., Cochrane, A., Cox, C. R., Gorman, T. E., & Green, C. S. (2017). Perceptual Learning Generalization from Sequential Perceptual Training as a Change in Learning Rate. *Current Biology*, *27*(6), 840–846. https://doi.org/10.1016/j.cub.2017.01.046

Kauderer, B. S., & Kandel, E. R. (2000). Capture of a protein synthesis-dependent component of long-term depression. *Proc Natl Acad Sci U S A*, *97*(24), 13342–13347. https://doi.org/10.1073/pnas.97.24.13342

Krug, M., Lossner, B., & Ott, T. (1984). Anisomycin blocks the late phase of long-term potentiation in the dentate gyrus of

freely moving rats. *Brain Res Bull*, *13*(1), 39–42. https://doi.org/10.1016/0361-9230(84)90005-4

Li, H. L., & van Rossum, M. C. (2020). Energy efficient synaptic plasticity (P. Latham, R. L. Calabrese, & W. Senn, Eds.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *9*, e50804. https://doi.org/10.7554/eLife.50804

Linden, D. J. (1996). A protein synthesis-dependent late phase of cerebellar long-term depression. *Neuron*, *17*(3), 483–490. https://doi.org/10.1016/s0896-6273(00)80180-4

Matthies, H., Frey, U., Reymann, K., Krug, M., Jork, R., & Schroeder, H. (1990). Different mechanisms and multiple stages of LTP. *Adv Exp Med Biol*, *268*, 359–368. https://doi.org/10.1007/978-1-4684-5769-8_39

Mozzachiodi, R., & Byrne, J. H. (2010). More than synaptic plasticity: Role of nonsynaptic plasticity in learning and memory [Place: Netherlands Publisher: Elsevier Science]. *Trends in Neurosciences*, *33*(1), 17–26. https://doi.org/10.1016/j.tins.2009.10.001

Nguyen, P. V., Abel, T., & Kandel, E. R. (1994). Requirement of a critical period of transcription for induction of a late phase of LTP. *Science*, *265*(5175), 1104–1107. https://doi.org/10.1126/science.8066450

Niven, J. E., & Laughlin, S. B. (2008). Energy limitation as a selective pressure on the evolution of sensory systems. *J Exp Biol*, *211*(Pt 11), 1792–1804. https://doi.org/10.1242/jeb.017574

Otani, S., Marshall, C. J., Tate, W. P., Goddard, G. V., & Abraham, W. C. (1989). Maintenance of long-term potentiation in rat dentate gyrus requires protein synthesis but not messenger RNA synthesis immediately post-tetanization. *Neuroscience*, *28*(3), 519–526. https://doi.org/10.1016/0306-4522(89)90001-8

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain [Place: US Publisher: American Psychological Association]. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Sossin, W. S. (2021). *The Oxford Handbook of Neuronal Protein Synthesis* [Google-Books-ID: KYskEAAAQBAJ]. Oxford University Press.

Tully, T., Preat, T., Boynton, S. C., & Del Vecchio, M. (1994). Genetic dissection of consolidated memory in Drosophila. *Cell*, *79*(1), 35–47. https://doi.org/10.1016/0092-8674(94)90398-0