Does forgetting save the brain's metabolic energy for future learning?

Pooja Nagendra Babu (20216094)
MSc, Computational Neuroscience, Cognition and AI
University of Nottingham

21st September 2020

Abstract

The human brain only weighs 2% of the total body mass but is responsible for 20% of the body's resting metabolism. Therefore, it is critical that the brain is an energy-efficient organ. Theoretical evidence has suggested that the energy cost to form new memories rises steeply with the number of memories already stored in the system. Thus, the brain might have mechanisms to lower the memory load in order to decrease the energy consumption for future learning. That is, the brain forgets some memories to make future learning more energy-efficient. In this project, I study the impact of forgetting on the amount of energy expended on future learning and network performance using a single-neuron model called the perceptron. By implementing forgetting as exponential decay of weights of the perceptron, I show that the maximum capacity of the perceptron decreases with an increase in decay rate. I devise a method to measure the accuracy of the perceptron as a function of decay value and show that accuracy and learning time decreases with increase in the decay rate, and energy increases with increase in decay rate. Lastly, I implement training schedules that would lead to catastrophic forgetting of some old memories to investigate if the perceptron saves energy for future learning when the network permits some of the initial learning to be forgotten. I show that such a strategy does conserve energy to learn new memories, whereas passive forgetting with decay on weights does not benefit in saving energy for future learning.

Contents

List of Figures

List of Tables

Introduction

Learning in the brain is a process by which the brain encodes information of the surroundings in the form of synaptic connections. These connections, over time, strengthen to form memories, help us to adapt to the surroundings, and makes us who we are. The strengthening of connections, also known as synaptic plasticity, is the biological process by which specific patterns of neural activity change the strength and efficacy of synaptic connections. It is the fundamental mechanism for learning and memory, and facilitates brain development and recovery from brain lesions. Many scientists have made important contributions to the understanding of synaptic plasticity, that dates back to the Spanish neuroanatomist Santiago Ramon y Cajal in the 1890s who suggested that the capacity of the brain could be augmented by increasing the number of connections [Cajal and i Laboratori de Ciències Mèdiques de Catalunya, 1893]. The properties of synaptic transmission rose to prominence during the twentieth century when Donald Hebb postulated that synapses change as a consequence of simultaneous firing, which forms the neural basis of learning and memory [Hebb, 1949].

One of the phenomena underlying synaptic plasticity is long-term potentiation (LTP), which is the persistent strengthening of synapses based on the neural activity. There are two forms of LTP: early-phase LTP and late-phase LTP. If only the early-phase LTP occurs, which lasts for minutes to an hour, short-term memories are formed [Zucker and Regehr, 2002]. On the other hand, late-phase LTP causes the synaptic strength to last for days or weeks. This phase is characterised by gene transcription and protein synthesis in the postsynaptic neuron. Experimental evidence supports this kind of plasticity in the dentate gyrus of the rabbit hippocampus [Bliss et al., 2003]. The phenomenon of late-phase LTP is known to be the neural basis of forming long-term memories and synaptic consolidation [Clopath, 2012].

As memories start to accumulate in the brain over time, it may be beneficial for the brain to have a mechanism to eliminate memories that are obsolete. For example, retaining outdated memories might impede judgement and make it difficult to adjust to a changing environment. Preserving strong and disabling memories associated with brain disorders like post-traumatic stress disorder (PTSD) can also be debilitating. This phenomenon of getting rid of unwanted memories, called forgetting, is considered as the flip side of learning. Forgetting can occur due to the failed retrieval of an intact engram — molecular traces in a set of neurons that store the learned information [Schacter et al., 1978] — through the biological degradation of molecular and cellular memory traces or when a fraction of engram cells become disconnected from the engram cell circuit [Davis and Zhong, 2017]. This form of forgetting is gradual and referred to as natural forgetting [Richards and Frankland, 2017]. Forgetting can also be artificially induced with interventions by reversing learning-induced changes in the synaptic strength by manipulating a protein kinase C (PKC) isoform, PKM-ζ, that plays a key role in maintaining LTP and memory [Tsokas et al., 2016].

Although forgetting appears to signify as a failure of the brain to recall memories, it is considered essential in processing incoming information. In a study involving Drosophila, forgetting is shown as an adaptive feature of the memory system to adjust to the changing environment by removing unrewarded memories [Brea et al., 2014]. A computational study by [Richards and Frankland, 2017] shows that forgetting offers advantages for memory-guided decision making in environments that change and are noisy. They propose that forgetting enhances behavioural flexibility by eliminating outdated infor-

mation and helps to generalise by preventing overfitting memories to instances from the past that may not be helpful in predicting the future.

Learning and memory, specifically long-term memory, formed through synaptic plasticity is an energy-intensive process [Mery and Kawecki, 2005]. Studies on Drosophila flies show that they increase their glucose intake during late stages of long-term memory formation, especially in the neurons of the mushroom body which is the fly's main memory centre [Plaçais et al., 2017]. Another study on Drosophila showed that under food shortage, the fly's brain disables the formation of energy-intensive long-term memories as a strategy for survival [Plaçais and Preat, 2013]. They postulate that the shutdown of late-phase LTP upon starvation may correspond to a mechanism of conservation between energy homeostasis and the ability to form long-term memories.

A more recent computational study by [Li and van Rossum, 2019] shows that the more memories a network has, the higher the metabolic cost to conduct future learning. Motivated by this, I investigate if forgetting some memories in the brain could conserve energy to learn new patterns in the future. I analyse this with the help of a single neuron model, also called a perceptron. I present input patterns to the perceptron and measure the energy consumption while I implement forgetting. I focus on two types of forgetting — passive forgetting and catastrophic forgetting — and examine the energy expended to learn new input patterns. I show that a trained perceptron with catastrophic forgetting saves more energy to learn new patterns when compared to a trained perceptron with passive forgetting.

Types of forgetting

Traditionally, forgetting in the brain is regarded as a slow and natural decay of memories over time due to the general instability of biological mechanisms. This form of forgetting is referred to as *passive forgetting* [Davis and Zhong, 2017]. Passive forgetting may occur due to the loss of context cues over time or interference due to other similar memories. This kind of forgetting could be translated to decay in weights over time in the context of neural networks. This kind of decay can improve generalisation by suppressing irrelevant components of the weight vector and can suppress some effects of static noise on the targets [Krogh and Hertz, 1992]. The results from [Li et al., 2012] establish that a Hebbian Learning rule with passive forgetting in a chaotic neural network (CNN) acts as a fuzzy-like pattern classifier that performs better than the ordinary CNN.

Learning can also happen continually where the networks learn by accommodating knowledge over time by learning a sequence of tasks. This process called continual learning is the ability to learn consecutive tasks without forgetting how to perform the previously learned tasks [Kirkpatrick et al., 2017]. This poses a problem for the networks to learn a sequence of tasks where learning new information may degrade the performance of previously learned tasks due to information loss. This property of the networks to forget the old information when learning the new information is called *catastrophic forgetting* [McCloskey and Cohen, 1989]. It is considered as a form of *active forgetting* which in neuroscience is considered as a method where the brain has mechanisms to remove memories that become unused [Davis and Zhong, 2017]. This can lead to a trade-off between the extent to which the system can become plastic in order to learn new information and remain stable in order to not catastrophically forget the old acquired knowledge, often referred to as the stability-plasticity dilemma [Abraham and Robins, 2005]. In contrast, humans and other animals can learn in a continual fashion in a lifelong manner. Experimen-

tal evidence suggests that in order to prevent catastrophic forgetting, there are specialised systems in the hippocampus that allows for learning new information which, over time, will be transferred to the neocortical system for long-term storage [Parisi et al., 2019].

Results

Learning in a perceptron

The metabolic energy expended due to synaptic plasticity during learning is studied in a perceptron. A perceptron is a single neuron model which linearly classifies the input patterns into binary classes. A simple perceptron, with its binary input and output, is used for modelling the operation of the cerebellar cortex [Brunel et al., 2004]. In our case, the input patterns are random patterns each associated to a randomly selected binary output. The perceptron takes the input patterns and calculates the output which is matched with the desired output. The synaptic weights are updated when there is a mismatch between the desired and actual output f according to the perceptron learning rule 1.

$$f = \begin{cases} 1 & if \sum_{i=0}^{N} w_i * x_i > 0 \\ 0 & otherwise \end{cases} \tag{1}$$

Here w_i is the weight of synapse i, x_i is the i^{th} input pattern, and N is the number of synapses. x_0 and w_0 are the bias values which are set to 1 and 0 respectively. The weights are updated for each pattern as per equation 2 until all the patterns are correctly classified [Van Der Malsburg, 1986].

$$w_i(t+1) = w_i(t) + \eta(d_j - f_j(t))x_{j,i} \tag{2}$$

where η is the learning rate which is set to 1, d_j is the desired output. The learning time is measured as the time taken for the perceptron to learn all the input patterns over multiple iterations, which is measured in epochs.

The amount of metabolic energy required to modify a synapse during the learning process is a parsimonious model [Li and van Rossum, 2019]. In this model, the metabolic energy for every modification of synaptic weight is proportional to the amount of change of the weights. The total metabolic cost M to train a perceptron is the sum of the changes in the synaptic weights,

$$M = \sum_{i=1}^{N} \sum_{t=1}^{T} |w_i(t) - w_i(t-1)|^{\alpha} \tag{3}$$

where w_i is the weight of synapse i, and T is the total number of time-steps required for the perceptron to learn the inputs. The exponent α is set to 1.

Forgetting affects memory capacity

Firstly, I implement passive forgetting in the perceptron which is implemented as a slow exponential decay of the synaptic weights which occurs at every time-step. With the decay in place, I try to check the capacity of the perceptron which can be defined as the maximum number of patterns that the perceptron

could successfully train. It is clear from the equation 3 that the energy required by the perceptron is proportional to the number of input patterns to be classified as more patterns require longer training time. Without decay, the critical capacity of a perceptron with N synapses and P patterns is when $P = 2N$ [Mitchison and Durbin, 1989]. However, when there is decay on the synaptic weights, the maximum capacity of the perceptron will be less than the critical value. The decay on the synaptic weights during the training process would forget (or unlearn) some of the previously learned patterns resulting in the maximum capacity being $P < 2N$. In order to test this, I train the perceptron for different values of synapses N and calculate the capacity as a function of the decay rate (Figure 1).

For the given range of decay rates, the maximum capacity of the perceptron displays a linear relationship with the decay value. As the decay value increases, the maximum capacity of the perceptron decreases aiding to the rapid decay of synaptic weights during the training period. A similar relationship can be seen in the energy expended by the perceptron for training the maximum number of patterns (Figure 2 (left)). The total energy decreases with the increase in the decay value. The larger the decay value, fewer the weight updates and lesser the energy required to train all the patterns presented to the perceptron due to decrease in the memory capacity. The relationship holds good for different values of N.

The plot showing the energy consumed per pattern is interesting (Figure 2 (right)). The curves represent the plot of a convex-like function. For $N = 1500$ and $N = 1000$, energy per pattern is high for decay 10^{-6} as the capacity of the perceptron is high and thus requires more energy to train all the patterns. The value of energy per pattern decreases as the decay value approaches 10^{-5} because the capacity decreases. As the decay reaches the value 10^{-4}, energy per pattern starts to increase. In this range, the high value of decay rate overpowers and the perceptron updates the weights more often even though the capacity is low, resulting in more energy consumption.

However, for $N = 500$ and $N = 250$, the lowest points of the curves appears to be different, near or beyond the decay value 10^{-4}. This can be explained by looking at the total energy curves for different N (Figure 2 (left)). The curves for higher N decrease faster with the increase in decay value in contrast to lower N. For lower N, the value of decay is not high enough to have a significant contribution on weight updates and hence energy. Thus, the lowest points of energy per pattern for lower N are beyond the decay value 10^{-4}.

For the fitted lines in Figure 1, I calculate the slope of the lines and find a linear relationship between the slope and the number of synapses, N (Figure 3). Note that for $N = 250$, a straight line is fit for the decay range of 4.10^{-5} to 10^{-4} only, while for other N, they are fitted from 10^{-6} to 10^{-4}. The coefficients of the fitted lines for different N are shown in the table 1.

Even though we see this linear relationship between patterns and decay rates for the medium range of decay values, this does not hold true when I consider lower and higher decay rates. The curve depicting maximum capacity tends to flatten for these values, resembling a sigmoid-like curve. This is because at zero decay on a log scale would yield the maximum capacity $P = 2N$ and for higher decay P is always greater than 0. I tested this for $N = 250$ and is as shown in the Figure 4.

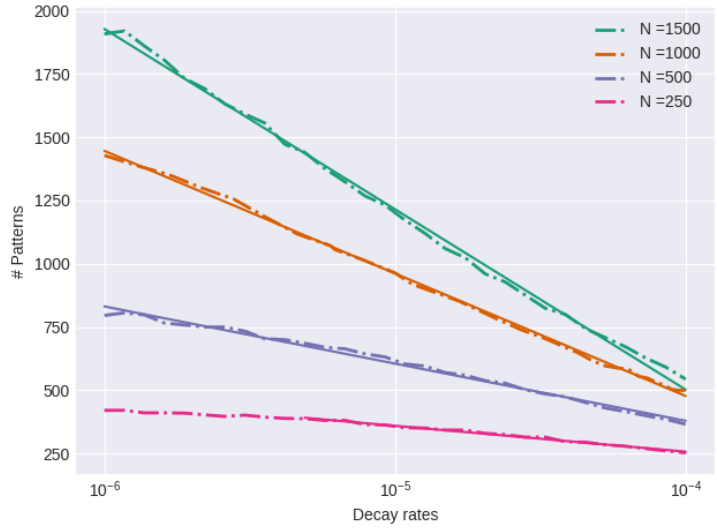

Figure 1: The maximum capacity of the perceptron as a function of decay rate. This is calculated for different values of N. The dotted lines shows the actual curve and the solid line shows a straight line fit to the curve. Each data point the graph is an average of 50 runs.

Forgetting affects accuracy

The performance of a perceptron is measured based on its ability to learn the input patterns and appropriately classify them. This measure, also called the accuracy of a perceptron, is defined as the number of input patterns that are correctly classified out of all the input patterns presented. A perceptron with a passive decay on its weights may not always learn all the patterns during the training process. Depending on the value of the decay rate, the training process may reach a point where there is no improvement in the accuracy of the perceptron. So, it becomes imperative to quit training at the right moment instead of trying to train the perceptron over multiple epochs with no improvement in its accuracy and waste unnecessary metabolic energy.

In order to achieve this, I devise a method to monitor the accuracy of the perceptron over the training process and quit it when the accuracy no longer improves. In this method, I calculate the accuracy of the perceptron in each epoch and quit the training when the mean accuracy of the last $(n+1)$ to $2n$ epochs exceeds or equals the mean accuracy of the last n epochs. I calculate the optimal value of the window size n, by training the perceptron with different values of n and analysing the value of accuracy at which the perceptron quits the training (Figure 5). The objective to calculate the optimal window size n is, if n is too small, the perceptron can quit training prematurely due to statistical fluctuations. On the other hand, if n is too large, the perceptron may over-train and over-estimate the metabolic energy cost. Hence, I decide on the optimal value as $n = 25$ because the mean accuracy starts to plateau around that value. I performed this analysis for decay 10^{-6} and 10^{-5} and found that window of $n = 25$ holds good for both values (Figure 5).

I then test the behaviour of the perceptron by fixing the number of synapses $N = 1000$ and the number of patterns to train as 1600. I measure the accuracy, learning time, and the energy consumed by the perceptron for a decay range of 10^{-8} to 10^{-2} (Figure 6).

From figure 6, we can see that with the increase in decay rate of the weights, the accuracy of the perceptron decreases and plateaus for higher decay values. A similar pattern can be observed for epochs or

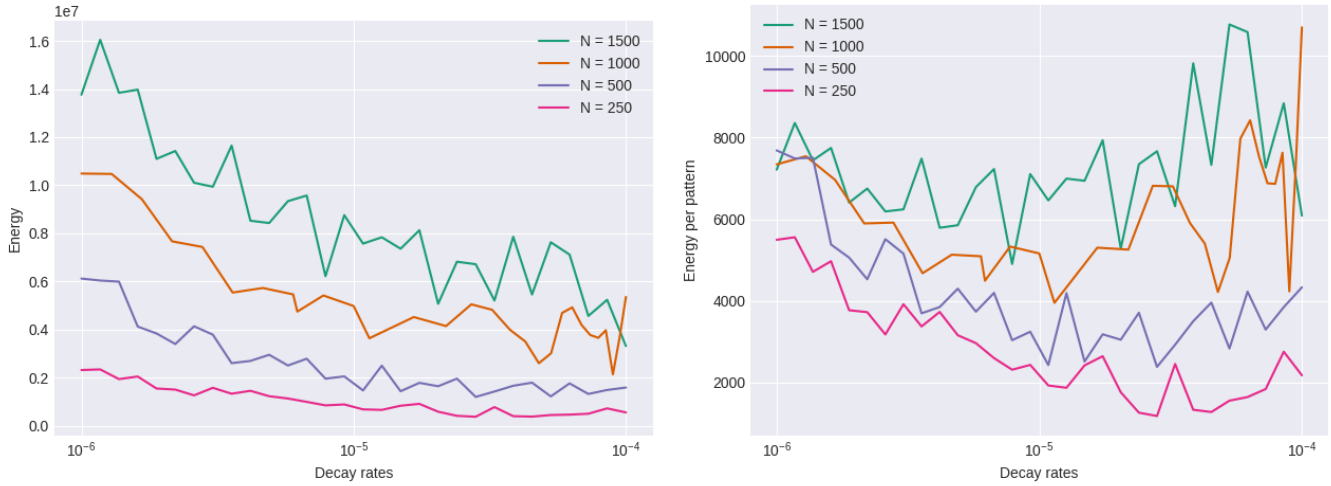

Figure 2: (left) The energy required to train the maximum capacity of the perceptron as a function of decay rate. (right) Energy required to train per pattern when the perceptron reaches its maximum capacity. Each data point in both the graphs is an average of 50 runs.

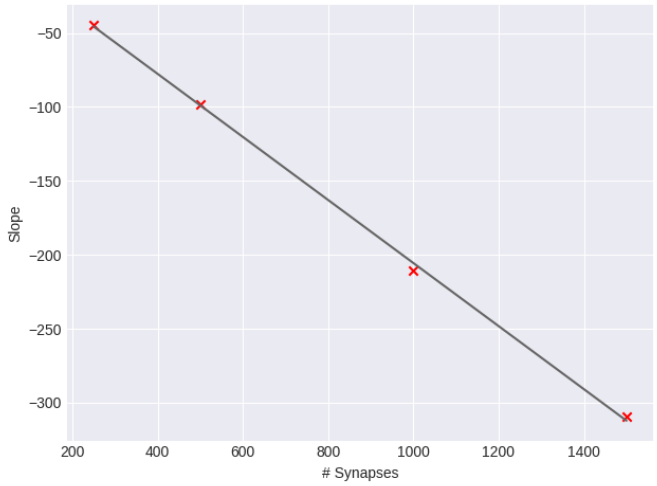

Figure 3: A plot of slope of the fitted lines in Figure 1 vs the number of synapses N.

N	Slope	Intercept
1500	-309.41380411	-2346.92473118
1000	-210.26772697	-1458.99595753
500	-210.26772697	-524.04731183
250	-44.34603812	-150.74210526

Table 1: Coefficients - slope and intercept values of fitted lines for different N

the learning time of the perceptron, where the learning time is high for low decay rates as the perceptron has a larger capacity, whereas as the decay value increases, the perceptron capacity reduces and so does the learning time until it plateaus for high decay values.

However, the trend of the curve showing the energy required for the perceptron to train the input patterns for a range of decay values is not straightforward (Figure 6 bottom). The energy expended by the perceptron to learn the patterns increases as the decay rate increases till the decay value reaches 10^{-5} after which the energy decreases till the decay value reaches 10^{-4} after which the value of energy increases again. In order to gain more insight into this behaviour of energy consumed, I check how the weights of the perceptron are updated in each epoch during the process of training as the value of total energy expended is directly determined by the change of synaptic weights (Equation 3). Thus, I plot the number of updates to the synaptic weights versus each epoch in the training (Figure 7).

The number of updates to the weights for the decay value of 10^{-6} is fewer than the updates for decay 10^{-5} (Figure 7 left). Hence the total energy consumed for 10^{-6} is less than for 10^{-5}. The number

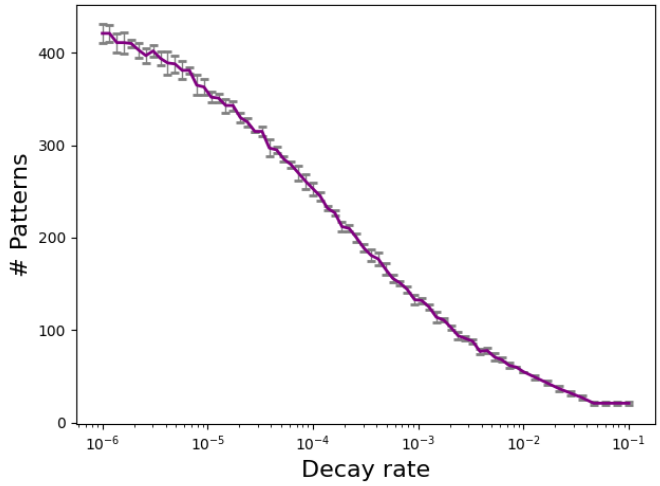

Figure 4: Maximum number of patterns trained by a perceptron for N=250 and for a wide range of decay values. Each data point is an average of 50 runs and the error bar is the standard deviation of the 50 runs.

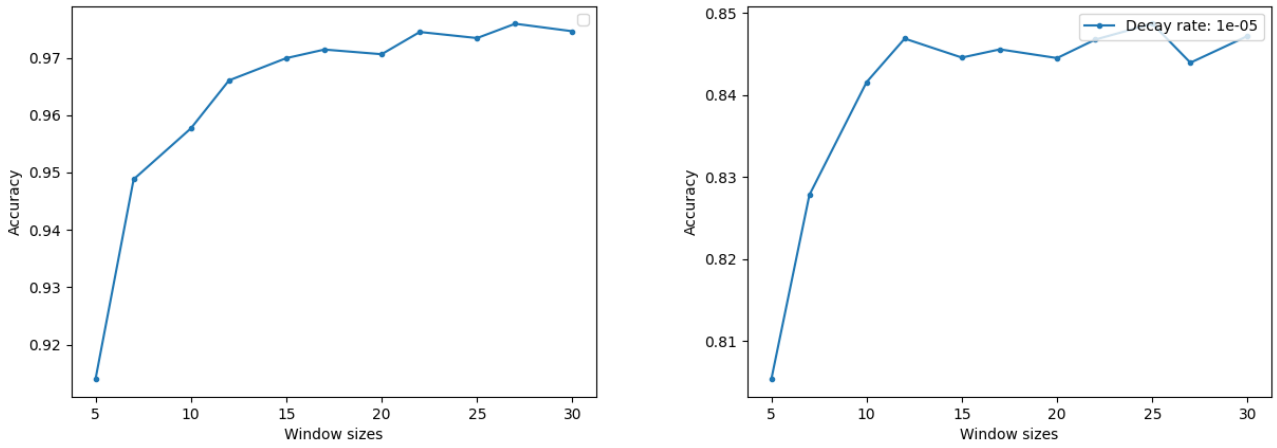

Figure 5: Accuracy with which the perceptron quits for different n where n represents the window size. The value of decay was fixed to 10^{-6} in the left and 10^{-5} in the right figure .

of updates to the weights increase as the decay rate approaches 10^{-5} which is reflected as increase in the energy expended.

Even though the number of updates to the weights is larger for the decay value of 10^{-4}, the perceptron has trained for a fewer epochs when compared to the decay value of 10^{-5} (Figure 7 left). So by the end of training, the perceptron has accumulated only a few synaptic weight updates in the former case and hence a dip in the energy curve for that decay value (Figure 6 bottom).

However, for the range of decay values in 10^{-4} and 10^{-3}, the energy expended by the perceptron increases again with the increase in decay. Clearly, the number of updates to the synaptic weights is more for decay value 10^{-3} when compared to the decay value of 10^{-4} even if there is not much of a difference in the number of epochs required for training (Figure 7 right). Also, accuracy of the perceptron is higher for the latter case (Figure 6 top-left) accounting for fewer weight updates than the former. This explains why there in an increase in the energy curve in the decay range 10^{-4} and 10^{-3}.

To summarise, the energy required to train the perceptron increases as the decay increases because

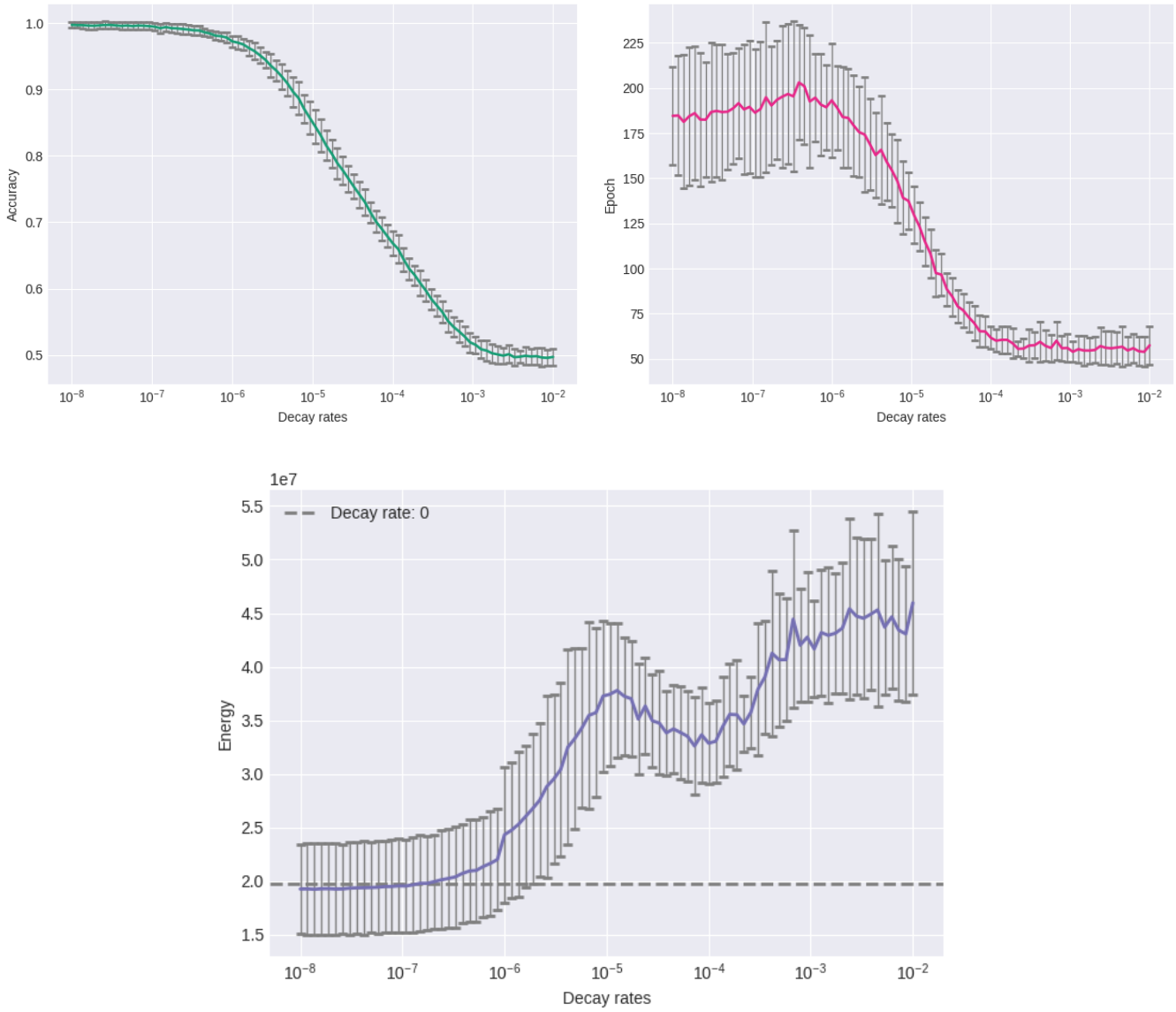

Figure 6: Accuracy, learning time, and energy consumed are measured for the perceptron with a decay in weights, resulting in passive forgetting. (Top left) Accuracy of the perceptron measured as the number of input patterns that are correctly classified among all the input patterns. (Top right) Epochs or the learning time of perceptron is the measure of the time (in epochs) required for the perceptron to learn input patterns until there is no further improvement in its accuracy. (Bottom) Energy expended during the training process. The dashed line indicates the energy value required to train the perceptron when there is no decay. The results indicate the values obtained by training the perceptron with $N = 1000$ and 1600 patterns over a decay range of 10^{-8} to 10^{-2}. Each data point is the average of 50 runs. The uncertainty shown for each data point is the standard deviation of 50 runs.

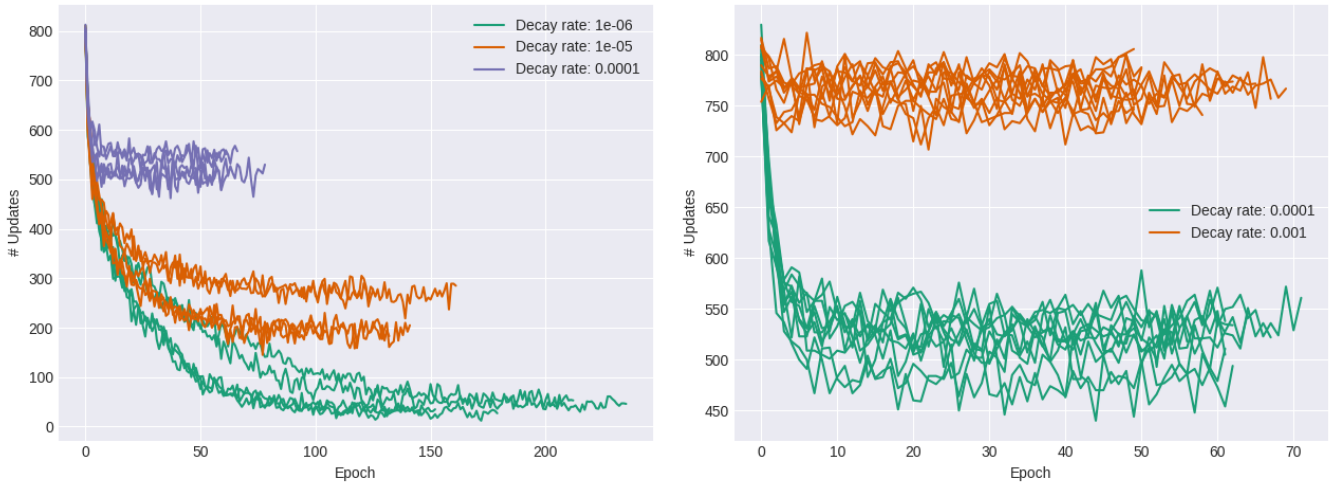

Figure 7: The number of updates to the synaptic weights of the perceptron during the training process. (Left) The number of updates to weights for the decay values 10^{-6}, 10^{-5}, and 10^{-4}. (Right) The number of updates to the synaptic weights over multiple runs for the decay values 10^{-4} and 10^{-3}. Each line on the plots represent a single run.

with the decay in weights the perceptron spends more time steps to learn the input patterns which leads to more updates in its weights thus leading to the increase in the total energy required for training.

Forgetting and energy conservation for future learning

In the previous section, we studied the accuracy of the perceptron while learning the input patterns with passive forgetting. We examined the energy consumption of the perceptron to learn new patterns while it was forgetting at the same time. However, biologically, memories are formed and stored initially and with time obsolete memories are forgotten. This has been shown to be beneficial in animal's survival [Brea et al., 2014]. At the same time, the brain consumes energy when it learns and forms new memories. A computational study by [Li and van Rossum, 2019] showed that the network spends more energy on future learning if it has a larger memory load. Drawing inspiration from this, I wanted to investigate if by allowing the network to decrease the memory load through forgetting, the network saved energy for future learning.

In order to test this, I develop two kinds of algorithms: one with passive forgetting, and another with training schedules that lead to catastrophic forgetting. I train the perceptron with a set of patterns which are considered as 'prior memories', followed by learning of new patterns. I then compare the energy required to train the perceptron using each of the algorithms with a benchmark algorithm in which the perceptron is trained without forgetting.

To give more details about the training procedure, a perceptron with $N = 1000$ synapses is considered and trained with an *initial training set* that contains 1000 patterns without any decay on the synaptic weights. This forms a perceptron with prior memories. For the *benchmark* algorithm, the perceptron is then trained with a *new training set* that consists of the first 1000 patterns along with an additional 100 new patterns. The energy expended in training these patterns would be the *benchmark* value that is used to compare with other algorithms. I develop three algorithms of training the perceptron that allows for catastrophic forgetting of some of the prior memories, one algorithm with passive forgetting, and another with the combination of the two.

Catastrophic forgetting is implemented as intentional forgetting of some of the old patterns from the initial training set learnt by the perceptron, with interference by introducing some new patterns to train while excluding those old patterns from the new training set. The algorithms with passive forgetting are implemented similar to the previous section where the synaptic weights decay away at a constant exponential rate after each epoch of training, with an added task here of learning new patterns. All the algorithms are implemented on a perceptron with $N = 1000$ synaptic connections. The number of patterns used to train the perceptron with some prior memories, the number of patterns allowed to be forgotten, the number of prior memories retained, and the number of patterns in the new training set for each algorithm is listed in table 2.

The first algorithm is implemented by initially training the perceptron with 1000 patterns, then by training with a new set of 700 randomly selected old patterns and 100 new patterns and measuring the energy required in training the new set. Note that here forgetting is implemented as intentional forgetting of some of the old patterns and interfering those by learning new patterns. The goal of this algorithm is to check the energy consumption with a new training set with total number of patterns in the new training set (800) less than the initial training set (1000). This algorithm is called *catastrophic forgetting 1.*

For the second algorithm, I initially train the perceptron with 1000 patterns and then allow the previous 1000 patterns to be interfered by only training with 100 new patterns the second time and measure the energy consumed. The objective here is to measure the energy consumed with a significantly lower number of patterns in the new training set than the number used to train originally. This algorithm is named *catastrophic forgetting 2.*

The third algorithm is implemented by initially training a perceptron with 1000 patterns, then by training a new set of 900 randomly selected old patterns and 100 new patterns, and measuring the energy expended in training the new set. The aim of this algorithm is to measure the energy consumption having a new training set $(900 + 100 = 1000)$ equal to the number of patterns in the original set (1000). This algorithm is called *catastrophic forgetting 3.*

We can observe from figure 8 that all the three catastrophic forgetting algorithms conserve some energy when compared with the benchmark value. The total number of patterns in the new training set for catastrophic forgetting 1 and catastrophic forgetting 2 algorithms is less than the number of patterns used for catastrophic forgetting 3 (Table 2 last column). Catastrophic forgetting 2 conserves the most energy because the new training set contains only the new patterns with all the originally trained patterns free to be forgotten. The perceptron takes fewer epochs to quickly learn the new patterns which are far less than the patterns in the original set, thus spending less energy.

The total number of patterns in the final training set of catastrophic forgetting 1 algorithm is less than the number of patterns used in catastrophic forgetting 3. In the former case, a perceptron with prior memories learns the new set of patterns more quickly because more old patterns are no longer required to be memorised as opposed to the latter. This allows the perceptron to learn the new patterns with fewer updates to the weights, consequently expending less energy. Energy conservation with catastrophic forgetting 3 is noteworthy here because the number of patterns used to train originally (1000) is the same as the new training set, which is made up of 900 old patterns and 100 new patterns. The perceptron clearly uses less energy for updating weights to learn 100 new patterns while retaining 900 old patterns

Algorithm	Initial training set	# patterns removed from the initial training set	# patterns retained from the initial training set	# new patterns	Total patterns: new training set
Benchmark	1000	0	1000	100	1100
Catastrophic forgetting 1	1000	300	700	100	800
Catastrophic forgetting 2	1000	1000	0	100	100
Catastrophic forgetting 3	1000	100	900	100	1000
Passive forgetting 1	1000	0	1000	100	1100
Passive forgetting 2	1000	100	900	100	1000

Table 2: Number of patterns for training the perceptron with different algorithms. The number of synapses in the perceptron is set to $N = 1000$

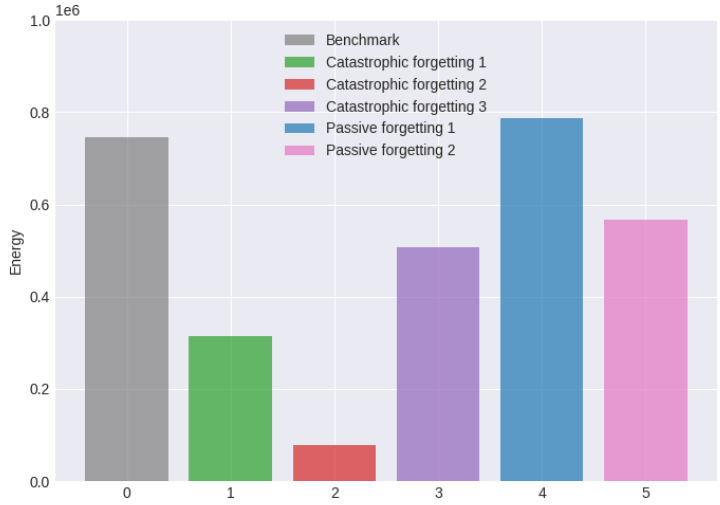

Figure 8: Energy consumed to learn a new training set for all the five different algorithms of learning in a perceptron with forgetting (catastrophic or passive). Here, the perceptron has $N = 1000$ synapses and trained for 1000 initial patterns. Afterwards, the perceptron is trained with a new training set that contains a certain amount of initial patterns decided by individual algorithms and 100 new patterns. The decay value for passive forgetting is fixed to 10^{-6}. The energy data points is an average value of 50 runs .

in comparison to learning the same number of patterns from scratch.

The last two algorithms are implemented with passive forgetting. In the first case, I train the perceptron initially with 1000 patterns with decay on the synaptic weights, then train 100 new patterns along with the old patterns with the same decay rate on weights. This algorithm is called *passive forgetting 1.* In the last algorithm, I first train the perceptron with 1000 patterns with decay on the synaptic weights, then train with a new set of 900 randomly selected old patterns by allowing 100 old patterns to be forgotten catastrophically, along with 100 new patterns with the same decay rate on weights and measure the energy expended in training the new set. This algorithm is called *passive forgetting 2.* The objective of the two algorithms is to examine if the gradual decay of memories conserves any energy to benefit future learning.

First, the perceptron is trained with a fixed decay value of 10^{-6} (Figure 8) for passive forgetting algorithms. The total number of patterns used for passive forgetting 2 is less than the patterns used for passive forgetting 1.

In order to clearly understand the energy consumption in these two scenarios, we must revisit how the perceptron learns. Learning in a perceptron is understood as a search for a weight vector in the space

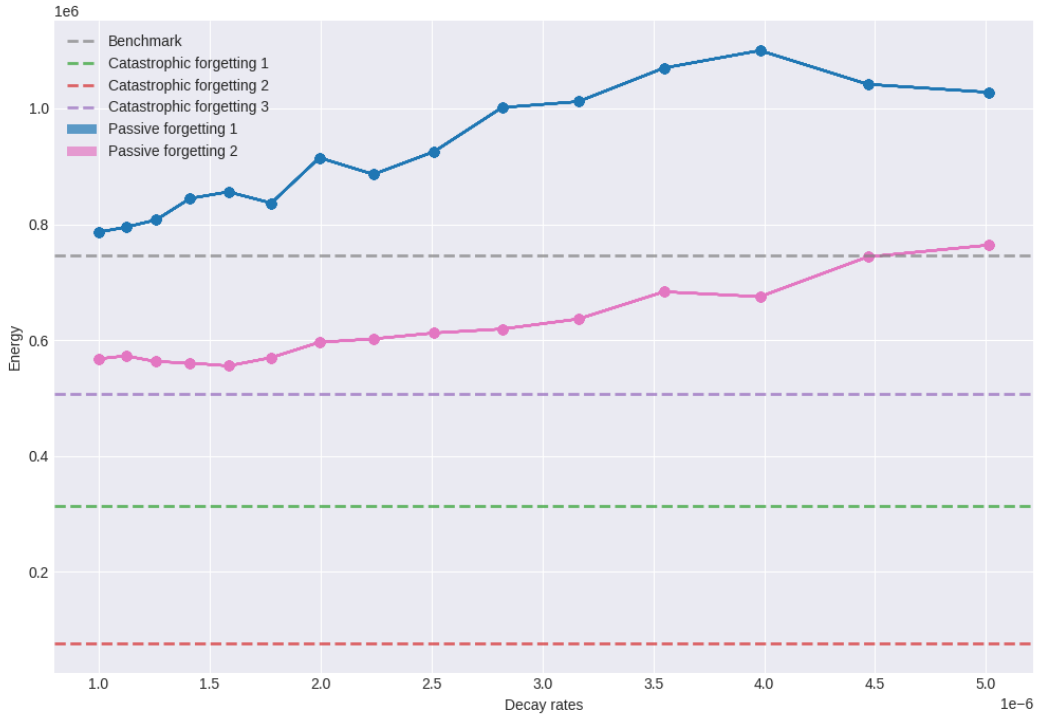

Figure 9: The energy consumption measured over a range of decay values from 10^{-6} to $5 * 10^{-6}$ for passive forgetting. The dashed lines represent the energy values for benchmark and three learning algorithms with catastrophic forgetting. The energy data points in this graph is an average value of 50 runs.

of synaptic weights that classifies all the patterns. The search of synaptic weights follows a random walk until a solution for the weight vector is found, and the metabolic cost of learning is proportional to the length of the random walk [Li and van Rossum, 2019]. Thus, intuitively, it appears that the gradual decay of weights at every epoch of the training process in passive forgetting would help in turning the search space of the weight vector of the perceptron to the right direction much faster such that all the patterns are classified correctly, as a result, conserving energy in the process.

However, in reality, the constant decay of weights don't actually lead to forgetting specific memories. The weight decay could also correspond to the new memories which makes it harder for the perceptron to learn new memories. Therefore, the weights get frequently updated in the training process, increasing the length of the random walk, and inevitably increasing the energy consumption. This behaviour can be clearly observed in the energy consumption of passive forgetting 1 algorithm where the value exceeds the benchmark (Figure 8). On the other hand, passive forgetting 2 with a combination of decay in weights and intentional forgetting consumes less energy than the benchmark. This combination gives the perceptron a little leeway to learn the new patterns more effectively than if it had learnt all of them at once (benchmark).

It is now interesting to compare the results of catastrophic forgetting 3 and passive forgetting 2 algorithms as they both have the same total number of patterns $(900 + 100)$ in the new training set, with the only difference to the decay of weights in the latter. As discussed above passive forgetting 2 performs better than the benchmark because the perceptron was allowed to forget 100 patterns catastrophically. However, it does not perform better than catastrophic forgetting 3 because the additional decay of synaptic weights in passive forgetting 2 further contributes to consuming more energy.

I extend this analysis over a range of decay values (10^{-6} to $5 * 10^{-6}$) for passive forgetting algorithms such that the perceptron trained all the patterns with 100% accuracy (figure 9)). The energy values of catastrophic forgetting algorithms are not affected by this as they are not dependent on the decay values. The passive forgetting 2 algorithm conserves energy compared to benchmark because the perceptron was allowed to forget some of the prior memories. The amount of energy conserved decreases with the increase in the decay value till $4.5 * 10^{-6}$ after which there is no gain in energy savings. This is because the weights decay faster and the perceptron has to spend more energy into updating the weights to learn new patterns. The passive forgetting 1 algorithm did not conserve energy for any value of decay showing that forgetting with only decay on weights is not beneficial for future learning. Hence, the catastrophic forgetting algorithms perform well overall and conserve more energy as compared to passive forgetting algorithms.

Discussion

Synaptic plasticity is known to be the neural basis of learning in animals. The synapses strengthen during learning and lead to the formation of short-term and long-term memories. Forgetting plays an integral part in the maintenance of these memories and helps the animals to adapt to varying environments. The formation of long-term memories through synaptic plasticity is an energy-intensive process. It has been shown theoretically that networks with fewer memories use less energy on future learning, therefore I wondered if forgetting can help networks save energy on learning. With the help of a perceptron, I induced algorithms with passive forgetting and training schedules that led to catastrophic forgetting. Firstly, I implemented passive forgetting as a constant decay on the synaptic weights of the perceptron and studied the maximum capacity of a perceptron. I showed that for a different number of synaptic connections, the maximum capacity of the perceptron decreased with the increase in the decay value. I also showed that the energy required to train the perceptron to its maximum capacity with passive forgetting decreased with the increase in the decay value due to a reduction in training time as the capacity decreased. Even though biologically, not all the synaptic connections of a neuron contribute to memories, the result signifies the limitations on the capacity of forgetful networks to hold memories.

Secondly, I devised a method to measure the accuracy of the perceptron as a function of the decay rate. I measured the accuracy, learning time, and the energy consumed by the perceptron and showed that the accuracy and learning time decreased with increase in decay till it flattened for high decay values, and the total energy consumed increased with increase in decay values.

Finally, in order to investigate if old patterns that are allowed to be forgotten in the perceptron saved energy to learn new patterns, I used the passive forgetting and interference mechanisms leading to catastrophic forgetting in the brain and implemented algorithms to train the perceptron. I demonstrated that catastrophic forgetting algorithms conserved a significant amount of energy for future learning. Biologically, the intentional forgetting and interference mechanisms used in catastrophic forgetting algorithms is observed in an fMRI study of word-picture associations that showed suppression of non-practised memories by the way of forgetting by inhibition [Wimber et al., 2015].

I showed that passive forgetting algorithms do not perform very well with respect to conserving energy because of the additional decay on weights. I also showed that even though passive forgetting

2 appears to conserve some amount of energy for future learning, it does so due to the prior memories being forgotten catastrophically and not due to decay on its weights. I attested this by comparing it with the energy consumed due to passive forgetting 1 which forgets only with decay on its weights. Overall, I concluded that algorithms with catastrophic forgetting performed better and conserved energy for future learning as opposed to passive forgetting algorithms.

More realistically, learning in animals is not a onetime phenomenon. We normally learn something new multiple times by practice or recall spaced over multiple intervals. This could be mimicked in our model by changing the passive forgetting algorithm 2 with more frequent exposure of new patterns to the perceptron. For instance, at each epoch, the randomly selected 900 old patterns in passive forgetting 2 could be trained in three subsets with 300 old patterns in each subset along with 100 new patterns. This way, the perceptron would learn new patterns three times rather than only once at each epoch. Perhaps, with this new schedule that trains new patters more often than the prior memories, passive forgetting could save energy. However, due to time constraints, I could not test this and would have to leave this investigation for future study.

In this study, energy is calculated by a parsimonious model with only the metabolic cost of synaptic plasticity as it offers a significant contribution. In reality, when synaptic connections are changed, neuronal properties are altered, which can lead to a difference in energy consumption in the networks. While we could incorporate these to calculate energy, the model becomes too complex to measure.

The brain typically contains billions of neurons which are connected with trillions of connections. Although our analysis interprets the results for a single-neuron model, it would be beneficial to extend our study to a larger network of neurons with single or multiple layers as this would be more representative of the connections in the brain.

Acknowledgements

I would like to thank my supervisor Ho Ling Li for assisting me through this project. Her guidance helped me to understand the topic better; the discussions, her invaluable inputs and feedback throughout aided me to complete this research project.

References

[Abraham and Robins, 2005] Abraham, W. C. and Robins, A. (2005). Memory retention - the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73 – 78.

[Bliss et al., 2003] Bliss, T. V. P., Collingridge, G. L., Morris, R. G. M., and LÃžmo, T. (2003). The discovery of long-term potentiation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1432):617–620.

[Brea et al., 2014] Brea, J., Urbanczik, R., and Senn, W. (2014). A normative theory of forgetting: Lessons from the fruit fly. *PLOS Computational Biology*, 10(6):1–9.

[Brunel et al., 2004] Brunel, N., Hakim, V., Isope, P., Nadal, J.-P., and Barbour, B. (2004). Optimal information storage and the distribution of synaptic weights: Perceptron vs. purkinje cell.

[Cajal and i Laboratori de Ciències Mèdiques de Catalunya, 1893] Cajal, S. and i Laboratori de Ciències Mèdiques de Catalunya, A. (1893). *Nuevo concepto de la histología de los centros nerviosos: conferencias pronunciadas en la Academia y Laboratorio de Ciencias Médicas de Cataluña en los días 14, 18 y 19 de marzo de 1892*. Imprenta de Henrich y Cia.

[Clopath, 2012] Clopath, C. (2012). Synaptic consolidation: An approach to long-term learning. *Cognitive neurodynamics*, 6:251–7.

[Davis and Zhong, 2017] Davis, R. L. and Zhong, Y. (2017). The biology of forgetting - a perspective. *Neuron*, 95(3):490 – 503.

[Hebb, 1949] Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. Wiley, New York.

[Kirkpatrick et al., 2017] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

[Krogh and Hertz, 1992] Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization. In Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems 4*, pages 950–957. Morgan-Kaufmann.

[Li and van Rossum, 2019] Li, H. L. and van Rossum, M. C. W. (2019). Energy efficient synaptic plasticity. *bioRxiv*.

[Li et al., 2012] Li, Y., Zhu, P., Xie, X., He, G., and Aihara, K. (2012). Learning-induced pattern classification in a chaotic neural network. *Physics Letters A*, 376(4):412 – 417.

[McCloskey and Cohen, 1989] McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press.

[Mery and Kawecki, 2005] Mery, F. and Kawecki, T. J. (2005). A cost of long-term memory in drosophila. *Science*, 308(5725):1148–1148.

[Mitchison and Durbin, 1989] Mitchison, G. and Durbin, R. (1989). Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics*, 60:345–365.

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

[Plaçais et al., 2017] Plaçais, P.-Y., de Tredern, Ã., Scheunemann, L., Trannoy, S., Goguel, V., Han, K.-A., Isabel, G., and Preat, T. (2017). Upregulated energy metabolism in the drosophila mushroom body is the trigger for long-term memory. *Nature communications*, 8(1):15510–.

[Plaçais and Preat, 2013] Plaçais, P.-Y. and Preat, T. (2013). To favor survival under food shortage, the brain disables costly memory. *Science*, 339(6118):440–442.

[Richards and Frankland, 2017] Richards, B. and Frankland, P. (2017). The persistence and transience of memory. *Neuron*, 94:1071–1084.

[Schacter et al., 1978] Schacter, D. L., Eich, J. E., and Tulving, E. (1978). Richard semon's theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 17(6):721 – 743.

[Tsokas et al., 2016] Tsokas, P., Hsieh, C., Yao, Y. N., Lesburguéres, E., Wallace, E., Tcherepanov, A., Jothianandan, D., Hartley, B., Pan, L., Rivard, B., Farese, R., Sajan, M., Bergold, P., Hernandez, A., Cottrell, J., Shouval, H., Fenton, A., and Sacktor, T. (2016). Compensation for pkmζ in long-term potentiation and spatial long-term memory in mutant mice. *eLife*, 5.

[Van Der Malsburg, 1986] Van Der Malsburg, C. (1986). Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. In Palm, G. and Aertsen, A., editors, *Brain Theory*, pages 245–248, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Wimber et al., 2015] Wimber, M., Alink, A., Charest, I., Kriegeskorte, N., and Anderson, M. (2015). Retrieval induces adaptive forgetting of competing memories via cortical pattern suppression. *Nature Neuroscience*.

[Zucker and Regehr, 2002] Zucker, R. S. and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annual Review of Physiology*, 64(1):355–405. PMID: 11826273.