META-LEARNING BIOLOGICALLY PLAUSIBLE UPDATE RULES FOR UNSUPERVISED AND SEMI-SUPERVISED REPRESENTATION LEARNING

Aishwarya Balwani
abalwani6@gatech.edu

Vima Gupta
vgupta345@gatech.edu

Kevin Shu
kshu8@gatech.edu

ABSTRACT

Better understanding the brain's synaptic mechanisms for learning has long been an area of active research. However, much of it has largely been restricted to carefully hand-designing and testing biologically plausible rules of "learning how to learn" that have thus far been significantly outperformed by backpropagation when used to train artificial neural networks. In this work we propose to learn parametric synaptic update rules that take into account the biological restrictions of only using neuron-local information for updates, whilst simultaneously taking advantage of both, the inductive bias as well as recurrence of artificial recurrent neural networks (RNNs) to learn expressive artificial neural networks. In particular, we investigate the possibility of meta-learning Hebbian mechanisms of synaptic plasticity that lead to a good, low-dimensional latent space in artificial RNNs in an unsupervised manner – a phenomenon found in most biological intelligent agents that exhibit moderate to high levels of cognition. We test our methods on commonly used synthetic datasets and show that the learnt update rules are capable of i) leveraging unlabelled data when learning from a limited number of samples, and ii) generalizing across multiple learning tasks.

1 Introduction

The past decade has witnessed machine learning make big strides in solving a variety of problems across domains using artificial neural networks (ANNs) trained on large labelled datasets [1, 2, 3, 4]. However, learning in the absence of copious amounts of labelled samples still remains an open challenge, with the performance of ANNs on unsupervised and semi-supervised tasks being much lower than that on supervised tasks. This is in stark contrast with learning that occurs in nature, where most intelligent agents are able to learn different tasks fairly well with only a handful of labelled samples. This consequently raises the questions: What are the ways in which ANNs diverge from their biological counterparts and cause them to behave differently? Which aspects of biological neural networks can we incorporate into our current methods and designs to help address some of these issues? We identify two key ways in which conventional ANNs are markedly different from biological ones and can potentially be worked upon to lead to better systems.

The first is a matter of structure, where we advocate using RNNs as opposed to fully connected, feed forward multi-layer perceptrons (MLPs) when trying to learn an unsupervised task. Feedback connections are used in biological neural networks in many ways [5, 6] and provide the system with the ability to update its internal states, exhibit temporal dynamic behaviour, as well as associative memory. MLPs lack any feedback connections that could allow for top-down signaling, thus giving RNNs a distinct advantage when it comes to learning patterns [7] and dealing with sequential behaviour and dynamics [8, 9]. RNNs also share weights across layers and have fewer learnable parameters than MLPs with the same number of layers while being more expressive due to their recurrence [10].

Second, we suggest that using biologically realistic training mechanisms instead of backpropagation might lead to improvements in robustness and generalization of ANNs. During learning, the brain uses synaptic plasticity as a means to improve behaviour, updating the weights of the connections between two neurons as a function of the correlations in their activity or firing patterns [11, 12, 13]. Specifically, when the activity of a pre-synaptic neuron successfully drives that of a post-synaptic neuron, the connection between the two is potentiated. These updates are strictly local and do not make use of a global error signal or require transfer of information from post-synaptic to pre-synaptic neurons, thus making conventional backpropagation [14, 15] an unlikely candidate for bio-realistic credit assignment in the brain.

In our work we first show that RNNs do indeed outperform MLPs of similar "structure" when trained to perform unsupervised tasks. In particular, we find that while both classes of ANNs achieve low error rates in terms of mean squared error (MSE) on an autoencoding task, RNNs learn a latent space that is far more respectful of the true latent structure of the data in comparison to MLPs. We subsequently attempt to meta-learn, (i.e., learn to learn) parameterized local update rules as applied to RNNs, with the goal being able to successfully learn tasks that make use of little to no labels. Through our experiments we provide evidence that the learnt bio-realistic update rules show promise in being able to train artificial systems that generalize across both, different tasks and datasets.

1.1 Related Work

Biologically Plausible Learning Rules. Research towards understanding how networks of interconnected neurons solve tasks in a manner similar to the brain has been carried out by both, the neuroscience and machine learning communities. Examples include works which lay an emphasis on the neuron-locality of the procedure, using only information that is temporal and/or spatially local for any updates [16, 17], as well as those which allow for distinct sets of feed-forward and feedback connections [18, 19]. Another set of well-known learning rules are those such as spike timimng dependant plasticity (STDP) [20, 21] and Hebbian update procedures [22, 23] that loosely speaking are an embodiment of the phrase *cells that fire together, wire together*. Furthermore, there also exist a broad range of evolutionary algorithms that mimic biological mechanisms such as mutation, recombination and natural selection that have found success as a learning heuristic across several types of tasks [24].

Meta-Learning with Biologically Plausible Mechanisms. Much of the well-established meta-learning research in the machine learning community [25, 26, 27] has leveraged backpropagation and is therefore considered biologically implausible. However, there has also been a long and rich history of learning to learn with biologically plausible mechanisms; Some of the earliest work in the area of learning with local learning rules include [28, 29] where the general idea is to have an explicit parameterized learning rule that is a function of neuron local variables that is optimized over a set of training tasks using either 0^{th} order (e.g., simulated annealing, genetic algorithms) or 1^{st} order (i.e., gradient-based) optimization methods. More recent works along these lines that have now moved onto using fully connected or recurrent neural networks to parameterize the learning rule include [30, 31, 32, 33, 34].

2 Scientific Questions

The specific questions that we aim to answer through our experiments are as follows:

1. Do RNNs have a structural advantage over MLPs that make them more suitable for unsupervised learning?
2. Can we meta-learn good Hebbian update rules using gradient descent?
3. Is it possible to train an RNN with a local Hebbian-based plasticity rule with an acceptable level of success?
4. How do such training rules compare with regular backpropagation in terms of losses and accuracy?

3 Hypotheses

In line with the questions we hope to answer, the hypotheses and conjectures that we make are as follows:

- RNNs are indeed architecturally better suited to learn good latent representations from unlabelled data due to their dynamics and ability to memorize.
- It should be possible to meta-learn good Hebbian-like update rules using gradient descent, given the fact that it might indeed simulate evolutionary changes in living systems.
- While we do conjecture that it should be possible to train an RNN with a hebbian update, we believe that training might not be as easy as that using backprop. As a matter of fact, consistent stability in training RNNs is hard even using conventional techniques and we anticipate troubles in being able to do so with local update rules. On a similar note, we also anticipate longer training times due to the fact that we update one synapse at a time while backpropagation effectively updates the entire network simultaneously.
- We ultimately believe that the performance of our proposed methods might not be as high as that of backpropagation, given the relatively limited and greedy nature of our updates.

4 Methods

4.1 Data

For our experiments, we sample data from upto three different 2D datasets, namely circles, moons, and Gaussian blobs, each of which consists of two different classes (Fig. 2b - Top). We project the 2D data into a 10D space by multiplying them with a fixed random matrix, thus making the the ambient dimension of our data higher than its latent dimension. Our goal throughout is to be able to learn good latent representations of the data that respect their true 2D latent structure and make them distinguishable with minimal additional training and labelled data.

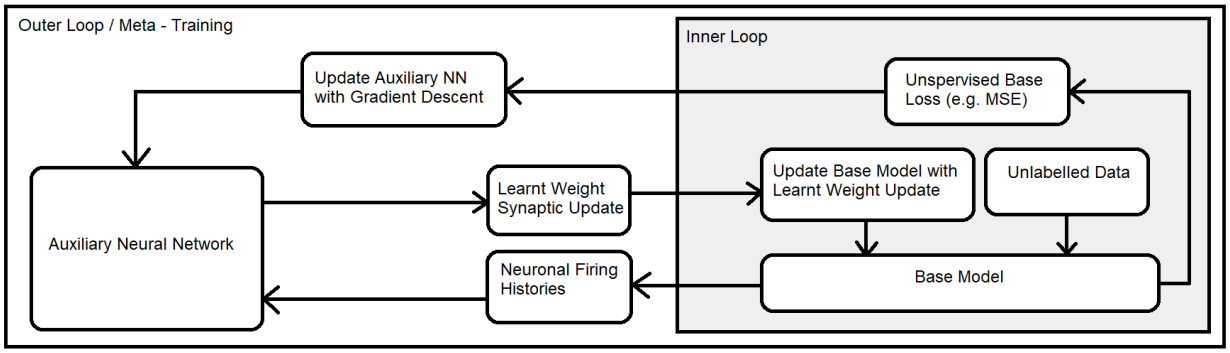

Figure 1: *Schematic for meta-learning with unlabelled data.* Our meta-learning scheme made up of an inner loop and an outer loop; The inner loop consists of a base model (in our case, an RNN) which takes as input unlabelled data and outputs a loss based on an unsupervised training objective such as autoencoding. Weights of the RNN are updated using an auxiliary neural network in the outer loop. The auxiliary nerual network takes as its inputs the firing rates of pre- and post-synaptic neurons to produce the weight update for a synapse and is trained using gradient descent.

4.2 General Setup

Our meta-learning model includes two levels of learning: i) The *inner loop* which learns a task using the learnt update rule for a given set of meta-parameters, and ii) The *outer loop* or *meta-training loop* that optimizes the meta-parameters to allow the inner loop to perform well on the *meta-objective* by scoring the quality of the inner model based on its loss.

Our inner model is an RNN, which consists of an input layer, a recurrent layer, and an output layer. In the inner training loop, the inner model is updated with a local update rule that only considers the firing patterns of the pre-synaptic and post-synaptic neurons, thus epitomizing Hebbian learning. Our inner loop uses no labelled data and simply updates the inner model's weights using the local update rule in a completely unsupervised manner. The outer loop consists of an auxiliary neural network that taskes as its inputs the firing rates of the pre- and post-synaptic neurons, and predicts the amount by which the weight of the synapse shared between the two neurons needs to be updated, thus giving us a parameterized way of learning the local update rule. The auxiliary network also recieves the loss of the inner loop as its own loss and is consequently trained using gradient descent.

4.3 Inner Model

Our inner model transforms data drawn from some n-dimensional vector space into an m-dimensional hidden state. The weights of our neural network consist of an $n \times m$ matrix, where M_{in} transforms input vectors into hidden vectors, an $m \times m$ matrix, M_{rec} which makes up our recurrent layer, and a $m \times n$ dimensional matrix, M_{out} for our output layer. For our activation function, we define the k**-cap** operation a_k, where for $x \in \mathbb{R}^n$, $a_k(x)$ is the n dimensional vector where $a_k(x)_i = 1$ if x_i is one of the k largest entries in x, and 0 otherwise.

We then pass our data through the RNN, starting with $x_0 \in \mathbb{R}^n$ and first computing $x_1 = M_{in}x_0$ followed by $a_k(x_1)$, and then for $t = 2, \ldots, T+1$, we let $x_t = a_k(M_{rec}x_{t-1})$. Finally, we let $x_{out} = a_k(M_{out}x_T)$ be our final output.

Consequently, $M = (M_{in}, M_{rec}, M_{out})$ is a parameterization of our model where $M(x)$ denotes the output of the layer M (i.e. x_{out}) on the input x.

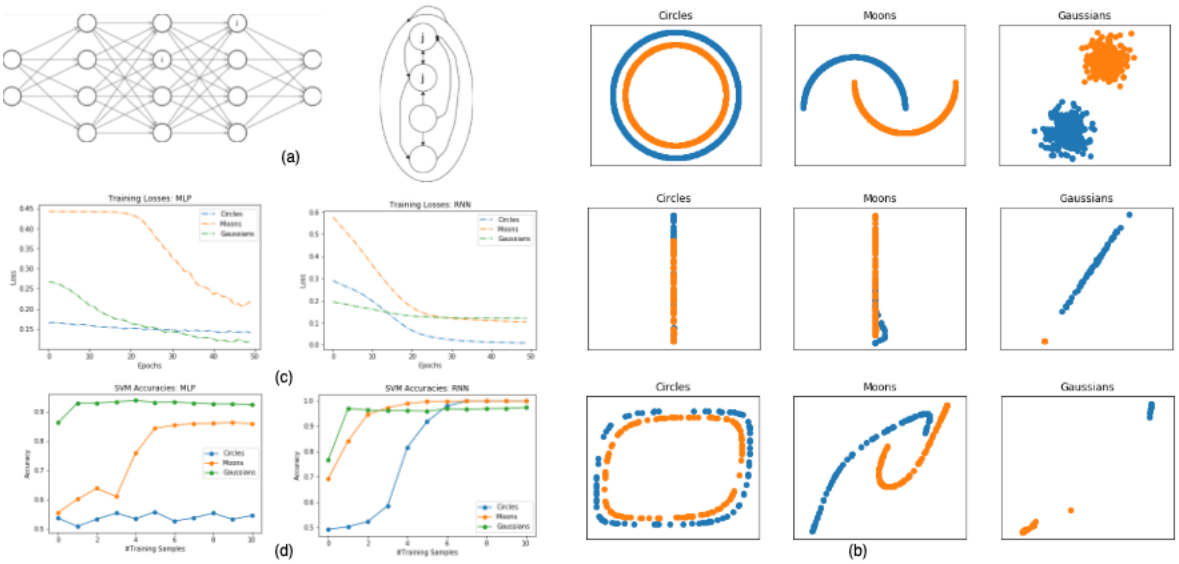

Figure 2: *Recurrent Neural Networks v/s Feed Forward Multi-Layer Perceptrons.* In (a) we show the two architectures that we compare, with the MLP to the left and RNN to the right. The RNN in our experiments is "unrolled" thrice, thus effectively simulating three layers of a feed-forward MLP. In (b) on the top we show the three original datasets (before projection), their latent representations learnt by a 3-layer MLP (middle), and those learnt by a thrice unrolled RNN, all using an MSE loss on an autoencoding task. In (c) we show the training losses of the MLPs (left) and RNNs (right) trained using backproagation. Panel (d) depicts the accuracy reached by an SVM trained on the latent representations extracted from the MLP (left) and RNN (right) when trained on varying numbers of labelled samples.

4.4 Outer Model

When a data point x_0 is presented to the network, we compute each of the x_i's for $i > 0$, after which we need to update the weights of the RNN. Our updates are represented by functions f_{in}, f_{rec}, f_{out}, which we think of as taking in the outputs of the outputs of m neurons, and then outputting an update to the weights. Our update consists of repeatedly performing the following operation:

$$M'_{ij} = M_{ij} + f(x_2, \ldots, x_{T+1})$$

where here, M is one of M_{in}, M_{rec} or M_{out}, and f is the corresponding update function. One issue with this formulation is that it appears that the updates to all of the weights will be the same; to fix this, we may order the entries of the x_i depending on i and j. The choice of which i, j it is encoded in the sorting of the x_i.

To represent the function f, we use a separate auxiliary neural network to take in the inputs $x_2, \ldots, x_{T+1}$ and output a single update. Representing the weights of the neural network computing f as θ, we can think of f as being parameterized by θ, and we emphasize this dependence using the notation f_θ.

In order to train the outer model, we try to get f to decrease the autoencoding loss of the RNN after every step or sample. That is, the loss of the model after 1 step is

$$\|(M + f_\theta(x_2, \ldots, x_{T+1})(x) - x\|^2$$

where $(M + f_\theta(x_2, \ldots, x_{T+1})(x)$ denotes the output of our model on input x after updating the weights of M by f.

We then use gradient descent (using the PyTorch framework), to compute the gradient of this loss with respect to the parameter θ (treating the weights of the inner model as constant).

5 Results

After applying the procedure described in the previous section, we can compare the results of this model to typical models, such as MLP or recurrent neural networks.

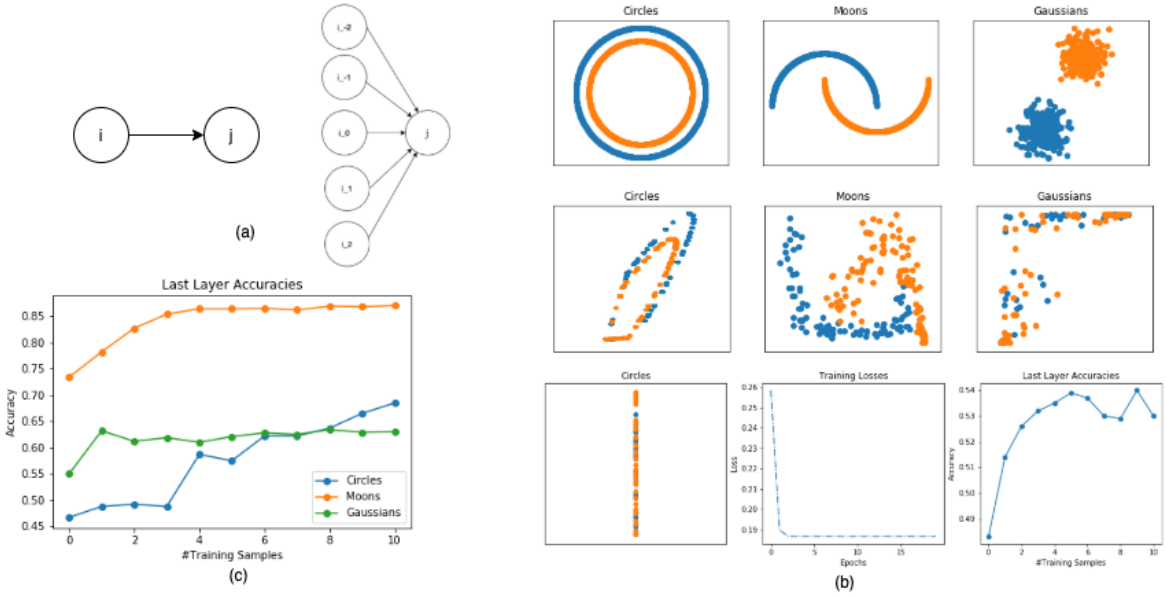

Figure 3: *Simple Hebbian update v/s Hebbian update with larger pre-synaptic fields.* In (a) we show the two Hebbian update mechanisms that we compare, with the simple Hebbian update to the left and Hebbian update with larger pre-synaptic fields on the right. In (b) on the top we show the three original datasets (before projection), their latent representations (middle) learnt by a learnt by a thrice unrolled RNN, using the simple Hebbian update rule. In the bottom, we show the latent representation for the circles data and corresponding loss and accuracy functions. In (c) we show the training losses of the RNNs trained using the Hebbian update rule, where we have the last layer accuracies reached by an SVM trained on the latent representations on the circles, moons and Gaussian datasets.

5.1 RNNs trump MLPs for unsupervised representation learning.

Fig. 2 conclusively shows that RNNs are better than MLPs with the same number of "hidden" layer, both quantitatively (Fig. 2c,d) and qualitatively (Fig. 2a). In particular, we notice that the training losses are lower for the RNNs overall and that the accuracies reached by training a simple SVM on their latent representations are also higher than those of their MLP counterparts. We quantified the amount by which the geometry is respected by the two types of neural networks by computing the Earth Movers' Distance between the learnt latent representations and the original 2D representations of the data, and found that the values for the RNNs (Moons = 0.1415,Gaussians = 0.565, Circles = 0.0205) are consistently lower than those of the MLPs (Moons = 0.285,Gaussians = 0.4635, Circles = 0.121).

5.2 Comparing a simple Hebbian update with one that uses an extended field of pre-synaptic histories.

Fig 3 shows the results when we extend the purview of the pre-synaptic weights taken into consideration for generating the weight updates from just the single neuron i_0 to all its neighbours $i_{-1}, i_{-2}, ...i_1, i_2$. While we expected the added freedom to let us learn better representations, we found that the learnt representations were mostly degenerate (e.g. 3c - Bottom shows the poor performance of the method on the circles dataset) as compared to the simple Hebbian update rule and its results as depicted in Fig. 3c.

5.3 Semi-supervised learning with limited number of labelled samples.

Fig. 4 showcases how latent space representation learnt using a unsupervised training, which is subsequently used as the feature space for supervised training with limited number of labelled samples, give better accuracy. When the last layer is re-trained using the learnt Hebbian update but with a cross-entropy loss, the upwards trend in accuracy proves that the learnt plasticity rules can generalize across tasks.

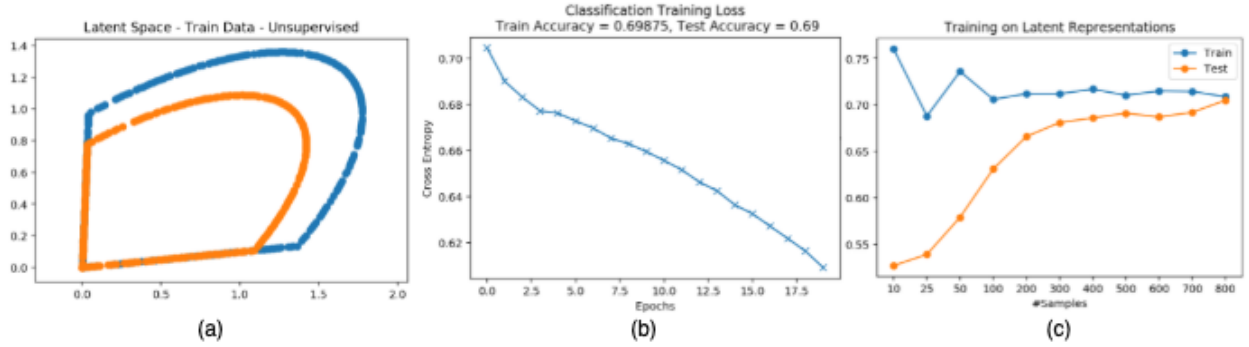

Figure 4: *Semi-Supervised Learning.* In (a) we show the latent space representation learnt using a unsupervised training that is subsequently used as the feature space for supervised training with limited number of labelled samples. In (b) we show the trend for the classification training loss that goes down as we increase the number of epochs for our supervised training. In (c) we show the accuracies achieved by both training and test data where the last layer is re-trained using the learnt Hebbian update but with a cross-entropy loss thereby proving that the learnt plasticity rules can generalize across tasks.

6 Discussion

We briefly discuss some of the ways in which we could both test and improve the results of our proposed methods.

The need to preform hyper-parameter tuning and use larger datasets to truly see the efficacies of the proposed methods. We have found that meta learning Hebbian training rules can in fact produce reasonable models, which are competitive with the baseline neural network models we trained using gradient descent. This departure from conventional update mechanism for recurrent neural networks leaves room for improvement in terms of final accuracy. The quality of representations would likely improve significantly with extensive parameter tuning on larger datasets. Furthermore, using methods such as Bayesian optimization for the purposes of outer loop training might also prove to be fruitful.

Attention as a mechanism to learn more expressive Hebian rules. At present there is no structure enforced on the weights being learnt in the auxiliary neural network when we work with the extended-Hebbian update that takes in as inputs the firing histories of all pre-synaptic neurons in the in the neighbourhood of the post-synaptic neuron. Such a lack of constraint on the weighted importance of neighbouring neurons has likely lead to sub-optimal and over-fit updates generated by the auxiliary neural network. Content-based attention mechanism (with the source being the pre-synaptic firing history and the target being post-synaptic firing history), which is known to be low rank by nature [35] might be a good way to ensure some degree of structure and in

Potential use cases where bio-realistic update rules might outshine conventionally trained ANNs. The proposed Hebbian update mechanism does not leverage gradients in any form once meta-learnt, and might therefore very likely outperform its counterparts trained using backpropagation in the presence of gradient-based adversarial attacks. Similarly, we also think that models trained using local Hebbian updates might perform well when the datasets are inherently noisy.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

[4] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[5] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363, 2013.

[6] Frank Tong. Primary visual cortex and visual awareness. *Nature reviews neuroscience*, 4(3):219–229, 2003.

[7] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[8] Aitor Almeida and Gorka Azkune. Predicting human behaviour with recurrent neural networks. *Applied Sciences*, 8(2):305, 2018.

[9] Umut Güçlü and Marcel AJ van Gerven. Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in computational neuroscience*, 11:7, 2017.

[10] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. *arXiv preprint arXiv:1711.00811*, 2017.

[11] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[12] H Markram and B Sakmann. Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer v pyramidal neurons. In *Soc. Neurosci. Abstr*, volume 21, page 2007, 1995.

[13] Wulfram Gerstner, Richard Kempter, J Leo Van Hemmen, and Hermann Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–78, 1996.

[14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[15] David Parker. Learning-logic: Casting the cortex of the human brain in silicon. 1985.

[16] Pierre Baldi and Peter Sadowski. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks*, 83:51–74, 2016.

[17] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

[18] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.

[19] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[20] Henry Markram and Misha Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382(6594):807–810, 1996.

[21] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215, 1997.

[22] Tim VP Bliss and Terje Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, 1973.

[23] Thomas Trappenberg. *Fundamentals of computational neuroscience*. OUP Oxford, 2009.

[24] Ruhul Sarker, Masoud Mohammadian, and Xin Yao. *Evolutionary optimization*, volume 48. Springer Science & Business Media, 2002.

[25] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[26] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016.

[27] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[28] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Citeseer, 1990.

[29] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, volume 2, 1992.

[30] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, 21(6):860–868, 2018.

[31] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Meta-learning update rules for unsupervised representation learning. *arXiv preprint arXiv:1804.00222*, 2018.

[32] Keren Gu, Sam Greydanus, Luke Metz, Niru Maheswaranathan, and Jascha Sohl-Dickstein. Meta-learning biologically plausible semi-supervised update rules. *bioRxiv*, 2019.

[33] Jack Lindsey and Ashok Litwin-Kumar. Learning to learn with feedback and local plasticity. *arXiv preprint arXiv:2006.09549*, 2020.

[34] Rares C Cristian, Max Dabagia, Christos Papadimitriou, and Santosh Vempala. Learning with plasticity rules: Generalization and robustness, 2021.

[35] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.