A biologically motivated rule for supervised learning with spiking neurons

Arjun Rao* Anand Subramoney* Robert Legenstein Wolfgang Maass

Institute for Theoretical Computer Science
Graz University of Technology
8010 Graz, Austria

* = first authors

March 23, 2020

Abstract

Layer 5 pyramidal (L5P) neurons are the most common neuron type in the cerebral cortex, and they span across all the cortical layers. There is increasing experimental evidence that the various dendritic compartments in these neurons play an important role in the way how these neurons integrate input coming from different cortical layers. Experimental studies showed that when a somatic spike coincides, within a given time window, with distal dendritic input that triggers NMDA spikes, it triggers a broad calcium spike at the apical compartment which causes the soma to spike in a high frequency burst. Thus, the L5P neuron acts as a "coincidence detector", detecting coincidences between somatic and distal dendritic spikes. There exists further experimental evidence that these calcium spikes play an important role in the synaptic plasticity at the distal dendritic compartments. A simple model of this coincidence detection dynamics along with theoretically grounded plasticity rules that are both functional and match the experimental data has so far been missing. Here, we propose a plasticity rule called the Spike-based Logistic Regression (SLR) rule for a multi-compartment neuron model, that uses this coincidence detection mechanism. The plasticity rule is well supported by biological data and has a clear theoretical interpretation. Our theoretical analysis suggests a functional role for L5P neurons as logistic regressors that learn to predict the somatic firing using input to the distal dendritic compartments. We tested the performance of the SLR plasticity rule with this neuron model on reconstruction and temporal sequence prediction tasks, and show that these tasks can be successfully solved using this plasticity rule.

1 Introduction

Pyramidal neurons are the most common neuron type in the cortex [Nieuwenhuys, 1994]. The morphology of the layer 5 pyramidal (L5P) cell spans across all cortical layers [Binzegger et al., 2004] and each cell receives top-down feedback input [Kuhn et al., 2008] at the distal dendrites, and sensory inputs at the basal dendrites. The pyramidal neurons consist of two major spike initiation zones – the soma of the neuron that exhibits predominantly sodium (Na^+) spikes, and a second spike initiation zone near the apical tuft that exhibit calcium (Ca^{2+}) spikes or plateau potentials [Larkum, 2013,

Amitai et al., 1993, Yuste et al., 1994, Schiller et al., 1997, Larkum and Zhu, 2002]. The distal dendritic branches also exhibit electrogenesis [Larkum et al., 2009, Major et al., 2013] in the form of NMDA spikes that can decrease the threshold for Ca^{2+} spikes in the calcium spike initiation zone in the apical dendritic compartment [Larkum et al., 2009]

The Ca^{2+} spikes are triggered when input is coincident between the soma and the distal dendrites. Specifically, when input to the distal dendritic tuft follows somatic firing, this triggers a broad Ca^{2+} spike which causes the neuron to burst at high frequencies. While the presence of coincidence detection mechanism is generally accepted, the functional role of this mechanism is not yet clearly understood.

In this paper, we develop a neuron and plasticity model, supported by data from biology, that proposes a clear functional role for the L5P neuron as a supervised learner. Specifically, we introduce a multi-compartment neuron model and a supervised learning rule for the distal dendritic weights, called the Spike-based Logistic Regression (SLR) rule, that makes use of the mechanism for detecting coincidences between the apical and basal compartments of the L5P neuron. The SLR rule as applied to the synapses incident on the dendrites depends only on whether there was an NMDA spike, whether this caused a coincidence, and on the branch-local NMDA compartment potential. So the learning rule at the dendrite does not need access to any somatic state (unlike the Urbanczik-Senn rule), which makes the learning rule very biologically realistic.

Using the SLR rule, the dendrites learn to predict the activity of the soma, i.e. the bottom-up input, when given top-down input that originates from higher regions or from layers that output encoded versions of the bottom-up input. To validate the rule, we setup laminar cortical models where the bottom-up sensory input is encoded by L2/3 neurons and the top-down input is decoded by the L5P neurons, and train these models to perform reconstructing of the sensory input. In addition, we theoretically show that the SLR rule performs stochastic gradient descent on the logistic cross-entropy loss.

This paper is structured as follows: First, we describe the neuron model and learning rule. Then we discuss the results of applying the SLR rule for three experiments: (a) classifying MNIST digits (b) reconstructing MNIST digits using a winner-take-all (WTA) hidden layer, and (c) predictive reconstruction of a rotating bar using the recurrent-WTA encoding of the sequence. This is followed by the proof of the learning rule performing stochastic logistic regression, and a description of the methods used in the experiments.

2 The Neuron model and learning rule

2.1 The Neuron model

We introduce a multi-compartment model of the layer 5 pyramidal neuron (see fig. 1) that consists of the somatic compartment, the apical dendritic compartment, which is the Ca^{2+} spike initiation zone, and one or more distal dendritic compartments where NMDA spikes can be triggered by synaptic input.

The distal dendritic compartment of the neuron is modeled as a stochastic spike response model where the instantaneous spiking rate for NMDA spikes is given by

$$\rho(t) = \rho_{max}\, \sigma\left(\beta(u(t) - u_0)\right) \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

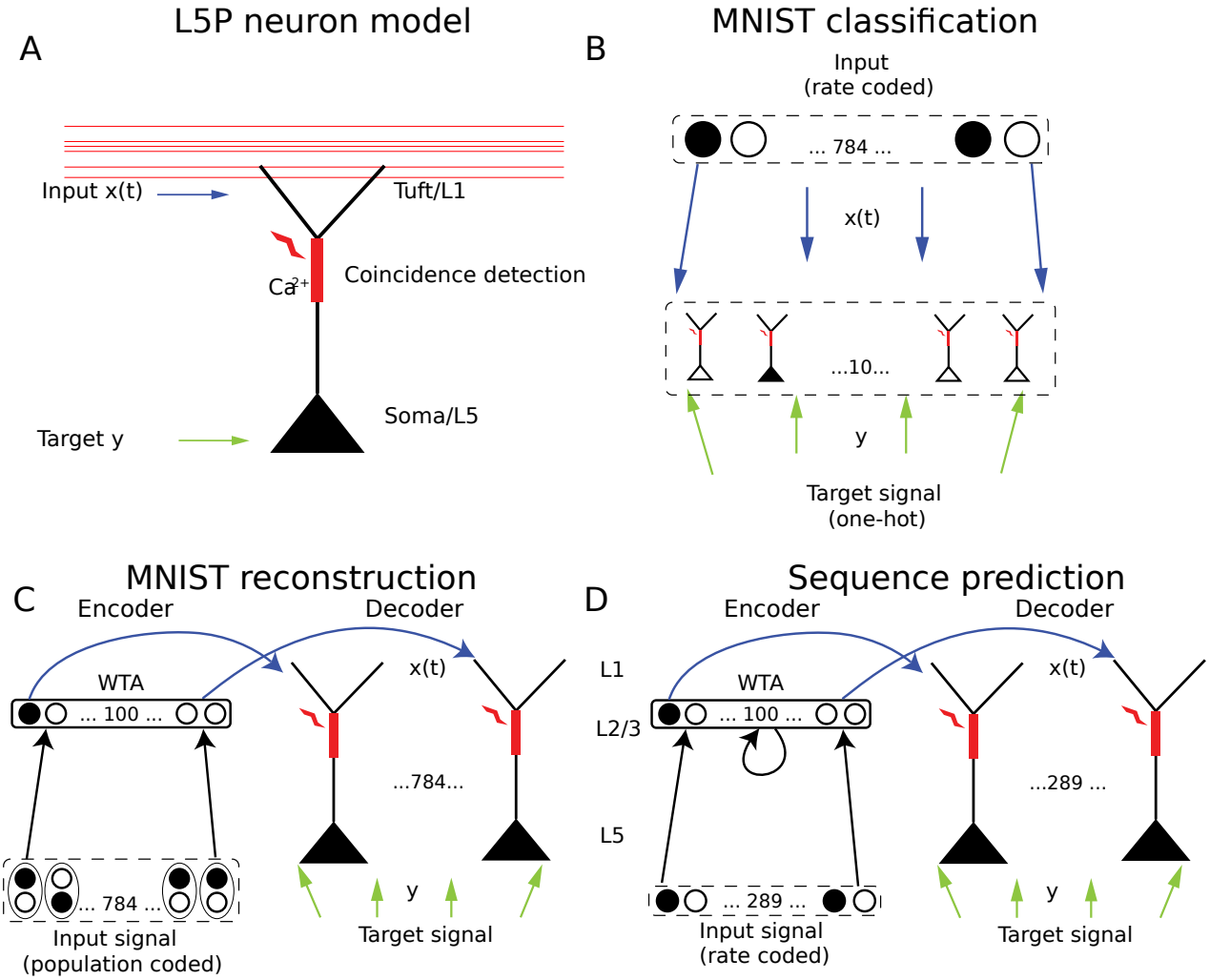

Figure 1: **A** The schematic of the L5P neuron model used with three types of compartments – the somatic compartment in L5, the apical dendritic compartment which serves as the Ca^{2+} initiation zone, and the distal dendritic compartments in L1. **B-D** Figures demonstrating the network setup for the three experiments. **B** Shows the network for the one-vs-rest MNIST digit classification. We use a rate coded input digit and the one-hot encoded target label to train the synapses incoming to the apical tuft of the L5P neurons using the SLR rule. **C** Shows the network used to auto-encode MNIST digits. We can see the population coded input, encoding using a WTA, and the L5P neuron decoding stage along with the target signal being fed into the soma of the L5P-neurons. **D** Shows the network used to predict sequences of rotating bars. We use rate coded input, laterally recurrent WTA circuit for encoding the hidden state, and an L5P neuron based decoding stage trained using the target signal being fed into the soma of the L5P-neurons.

Here $u(t)$ represents the membrane potential at the dendritic compartment, which is the weighted sum of the input post-synaptic potentials (PSPs) $\mathbf{x}(t)$ weighted by the incoming synaptic efficacies $\mathbf{w}$. That is,

$$u(t) = \mathbf{w}^T \mathbf{x}(t) \tag{2}$$

The NMDA spikes are denoted by $s_{\text{nmda}}(t)$ i.e. $s_{\text{nmda}}(t) = 1$ if there was a spike in the interval $[t - dt, t]$ for an infinitesimally small time dt.

The somatic compartment is also modeled as a spike response model with rate $\rho_s(t)$ which depends on the binary target $y \in \{0, 1\}$. $\rho_s(t)$ takes a high value $\rho_{target,high}$ to represent $y = 1$ and a low value $\rho_{target,low}$ to represent $y = 0$. We assume that the normal firing of the somatic compartment is influenced almost entirely by the target y.

When an NMDA spike and a somatic spike occur within τ_c ms of each other, we term this as a *coincidence event*. Each coincidence event triggers a Ca^{2+} spike at the apical dendritic compartment that lasts for τ_{Ca} ms. We denote this Ca^{2+} spike with the variable $z(t)$ where:

$$z(t) = \begin{cases} 1 & \text{if last coincidence event happend in interval } [t - \tau_{\text{Ca}}, t] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.2 Learning Rule

In the laminar model of the cortex, L5P neurons receive top-down feedback input from the higher regions of the cortex at L1 at the distal dendritic tufts and receive bottom-up sensory input at L5 at the basal dendrites. We model the top-down input with $\mathbf{x}(t)$ and the bottom-up feedback input with the targets y. We consider the L5P neurons to act as predictors at each layer, predicting the bottom-up input based on the top-down input it receives.

Specifically, we formulate our model as a regressor that learns to predict the sensory input based on the feedback input it receives. The spiking inputs to the dendritic compartments of the neuron are transformed to PSPs $\mathbf{x}(t)$. This represents the independent variable in the regression problem. The corresponding binary target a $y \in \{0, 1\}$ is given to the soma by causing it to fire at a high rate $\rho_{target,high}$ if $y = 1$ or at a low rate $\rho_{target,low}$ if $y = 0$. Eventually the dendritic firing rate predicts the probability that the target is 1 given the dendritic input, thus performing regression on the input $\mathbf{x}(t)$ with target y.

The incoming synapses at the distal dendrite are updated according to the following learning rule – the Spike-based Logistic Regression (SLR) Rule:

$$\Delta\mathbf{w}(t) \;=\; \eta \left(\frac{z(t)}{q(t)} - 1 \right) \; s_{\text{nmda}}(t) \; \mathbf{x}(t) \tag{4}$$

Where η is the learning rate and $q(t)$ is given by the following equation:

$$q(t) \triangleq \frac{\rho(t)}{\rho_{max}} = \sigma\left(\beta\left(u(t) - u_0\right)\right) \tag{5}$$

This learning rule can be written in discrete time as shown in table 1 which illustrates the coincidence-gated nature of the learning rule. No plasticity is triggered in the absence of NMDA spikes. When the NMDA compartment spikes without a corresponding coincidence event, a depressing plasticity update (LTD) is triggered. When a calcium spike is triggered due to a coincidence between the NMDA and the somatic spikes, the incoming synapses to the distal tuft are subject to a

NMDA spike $s_{\text{nmda}}(t)$	Ca^{2+} spike $z(t)$	weight update ($\Delta \mathbf{w}$)
0	0	0
0	1	0
1	0	$-\ \eta\ \mathbf{x}(t)$
1	1	$\eta\ \frac{\mathbf{x}(t)}{q(t)}\ -\ \eta\ \mathbf{x}(t)$

Table 1: Synaptic weight updates for simulation in discrete time in timesteps of Δt and where $q(t)$, $z(t)$, $x(t)$ are defined in the text.

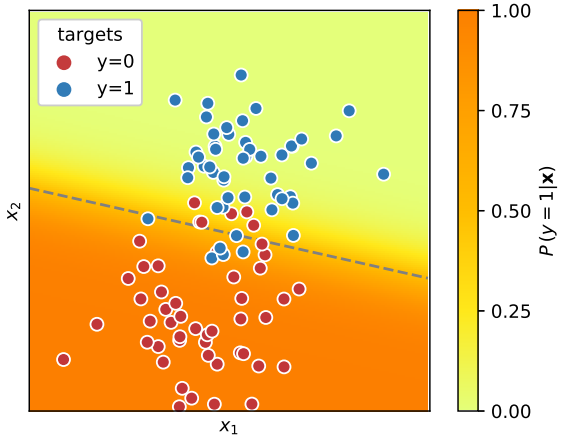

Figure 2: Schematic illustration of logistic regression. Two-dimensional input patterns $\mathbf{x}(t)$ from two classes are shown (circles) with color indicating target class. Dashed gray line shows the linear separation boundary (hyperplane) resulting from logistic regression. Background color indicates probability $p(y(t) = 1|\mathbf{x}(t))$ according to the logistic regression model.

potentiating update (since $q(t) \in (0, 1)$) (LTP). This update is consistent with biology for both LTD [Kampa et al., 2006] (Fig. 1) and LTP [Kampa et al., 2007] (Fig. 2).

In our learning setup, we consider the dendritic inputs $\mathbf{x}(t)$ to be the independent variable, and targets y to be binary (i.e. $y \in \{0, 1\}$). The somatic spike rate takes values $\rho_{target,high}$ and $\rho_{target,low}$ if y equals 1 or 0 respectively. In this setup, we show in sec. 4 that the above rule performs stochastic gradient descent on logistic loss function

$$\begin{aligned}\mathcal{LL} &= E_{\mathbf{x}(t),y}\left[y \log q(t) + (1-y) \log\left(1 - q(t)\right)\right] \\ &= E_{\mathbf{x}(t),y}\left[y \log \sigma\left(\beta\left(u(t) - u_0\right)\right) + (1-y) \log\left(1 - \sigma\left(\beta\left(u(t) - u_0\right)\right)\right)\right] \end{aligned} \tag{6}$$

Minimizing this loss function adjusts the weights $\mathbf{w}$ so that $q(t) = \frac{\rho(t)}{\rho_{max}}$ approximates $p(y = 1|\mathbf{x}(t))$

2.3 Logistic regression

The difference $u(t) - u_0$ between the membrane potential $u(t)$ and the threshold u_0 that appears in equation (1) defines in combination with equation (2) a scalar $a(t)$ for each input pattern $\mathbf{x}(t)$

$$a(t) = \mathbf{w}^T \mathbf{x}(t) - u_0, \tag{7}$$

with $\mathbf{w}$ being the vector of synaptic weights. The equation $a(t) = 0$ defines a hyperplane which separates input patterns into two classes (Fig. 2). On one side of the hyperplane are all patterns $\mathbf{x}(t)$ with $a(t) < 0$. The second class contains all patters $\mathbf{x}(t)$ with $a(t) \geq 0$. In the supervised learning problem, each pattern $\mathbf{x}(t)$ comes with a target $y(t)$ that indicates the true class of this pattern. The goal is then to find parameters $\mathbf{w}$ (synaptic weights in our case) such that the assignment to classes matches as good as possible the true class assignment, i.e., we want that $a(t) < 0$ if $y(t) = 0$ and $a(t) \geq 0$ if $y(t) = 1$.

The classical algorithm for this problem is the perceptron learning rule [Rosenblatt, 1958]. However, perceptron learning only succeeds if the data is linearly separable, that is, if there exists a hyperplane that can perfectly separate the classes. This condition is usually not fulfilled if there is noise in the data or if the data distribution is complex. Logistic regression avoids this problem by taking a probabilistic perspective [Bishop, 2006]. In can handle noise by assuming a model for the probability $p(y(t) = 1|\mathbf{x}(t))$ that target $y(t) = 1$ is observed for a given pattern $\mathbf{x}(t)$. This model is given by

$$p(y(t) = 1|\mathbf{x}(t), \mathbf{w}) = \sigma(a(t)) = \sigma(\mathbf{w}^T \mathbf{x}(t) - u_0). \tag{8}$$

Logistic regression then finds the optimal value of the parameter vector $\mathbf{w}$ for this model using the well-known machine learning principle of maximum likelihood learning (which maximizes the probability that the given targets are observed for the corresponding patterns in this model). By doing so, it can find a well-separating hyperplane also for data which is not linearly separable. More specifically, logistic regression amounts to finding the weights $\mathbf{w}^*$ that minimize the so-called cross entropy error function

$$E(\mathbf{w}) = -\sum_t \left[y(t) \log p(y(t) = 1|\mathbf{x}(t), \mathbf{w}) + (1 - y(t)) \log \left(1 - p(y(t) = 1|\mathbf{x}(t), \mathbf{w})\right)\right] \tag{9}$$

From the model in equation 8, this becomes

$$E(\mathbf{w}) = -\sum_t \left[y(t) \log \sigma \left(\mathbf{w}^T \mathbf{x}(t) - u_0 \right) + (1 - y(t)) \log \left(1 - \sigma \left(\mathbf{w}^T \mathbf{x}(t) - u_0 \right) \right) \right] \tag{10}$$

This amounts to a convex optimization problem, which implies that there exists only a single optimum which can easily be found for example by gradient-based learning.

3 Results

We tested the coincidence-detection based learning rule in the following three experiments to analyze its ability to model learning in the cortex. First we performed a simple experiment to test the if the rule could be used to classify MNIST digits through one-vs-rest classification. This experiment was used to validate the learning rule as a regressor, and the setup is not biologically realistic. Next, we model the laminar structure of the cortex where the bottom-up (sensory) input

is encoded by neurons in L2/3 and the top-down input is decoded by L5P neurons to perform input reconstruction. We use WTA circuits trained using the previously published Spike Expectation Maximization [Nessler et al., 2013] rule to model the L2/3 layer for encoding, and the L5P neurons trained using SLR to model the L5P neurons for decoding. In the third experiment, we use the same laminar setup as in the second experiment, but use it to do reconstruction of simple sequences with the addition of recurrent connections in L2/3 with a corresponding learning rule [Kappel et al., 2014]. We describe these experiments below in detail.

3.1 Classification of MNIST digits

We first use the SLR learning rule to perform MNIST classification as a simple test of how well the learning rule performs as a logistic regressor. The network structure is described in figure 1B. The input is binarized MNIST digits that are rate coded – one input neuron is used per pixel of the MNIST image and each neuron spikes with a low or high firing rate depending on whether the corresponding pixel is 0 or 1 respectively. The classifier consists of 10 L5P-neurons with each neuron responsible for predicting the likelihood of the input being a particular digit. The target signal used to train the network is implemented by clamping the firing rate of the soma of the L5P neurons in a one-hot manner i.e. only the soma of the neuron that corresponds to the target digit is clamped to a high firing rate, and that of all others to a low firing rate. During the testing phase, the predicted target label is the digit corresponding to the L5P neuron with the highest dendritic firing rate ρ. We achieve an error rate of 12% which is the state of the art for a linear classifier [LeCun et al., 1998]. The success of the one-vs-rest classifier indicates that the L5P Neuron is able to successfully provide an estimate of the confidence of its prediction as opposed to merely performing binary classification.

3.2 Reconstruction of MNIST digits

We model the laminar cortical structure where the bottom-up sensory input is encoded by L2/3 neurons. The L2/3 layer is implemented using multiple winner take-all (WTA) circuits where each WTA's input connections are learned independently using a local learning rule that does spike-based (stochastic) expectation maximization (SEM) [Nessler et al., 2013]. Using multiple WTAs is advantageous as the responsive neurons of each WTA network encode slightly different information thus enabling the efficient reconstruction of different styles of digits. Each neuron in the WTA is tuned to respond to a specific hidden cause that generated that input, which, in the case of MNIST images specializes roughly for a specific style of a specific digit. The input to the WTA is population coded i.e. two neurons fire for every pixel. One of them fires with a high rate ($\rho_{inp,high}$) if the pixel value is one, and with a low rate ($\rho_{inp,low}$) if it is zero. The other one is the inverse such that the total firing rate of the two neurons is constant.

The output of the WTA, which is the encoded representation from the L2/3 neurons is then decoded by the L5P neurons to provide a reconstruction of the input. The distal dendritic compartment receives the encoded representation of the digits as input and the incoming connections are trained using SLR. The soma is clamped to fire at the target rate based on the value of the corresponding image pixel.

After training, the distal dendritic compartments of the L5P neurons are able to predict the sensory input given only the encoded representation. The network setup is shown in figure 1C.

In table 2, we show the negative log likelihood generated by the model for the reconstructed images for various number of WTA circuits. The data in Table 2 demonstrates that an increase in the number of WTA units results in an increase in performance. Some samples of the reconstruction

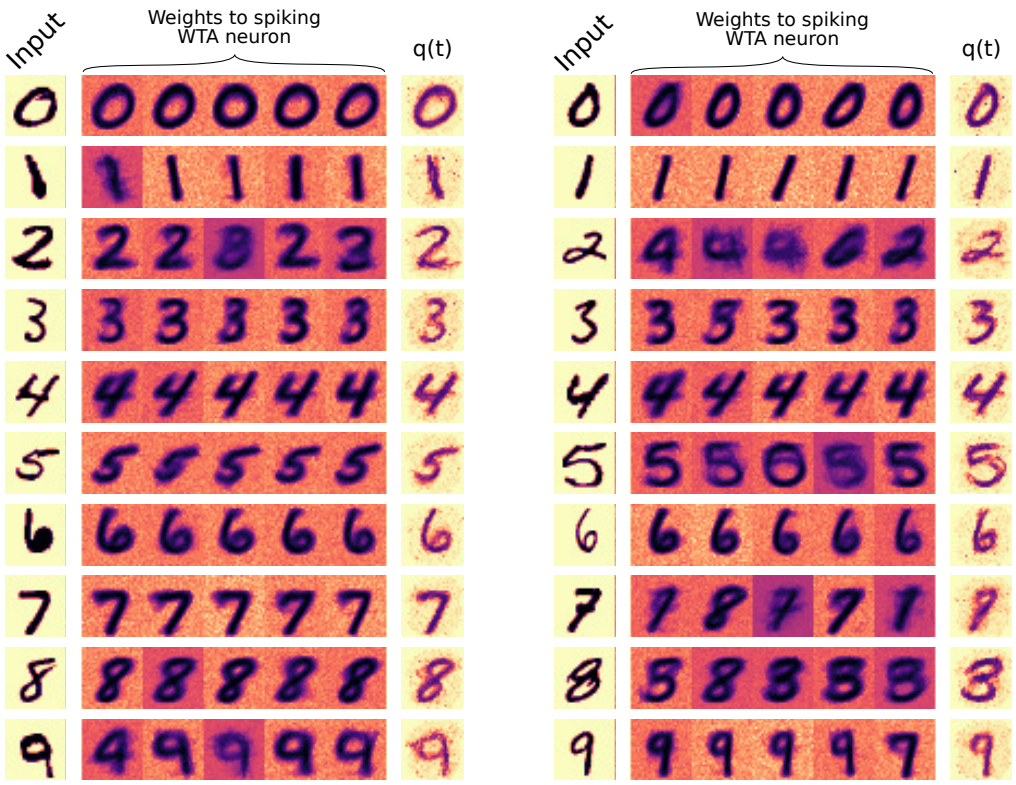

Figure 3: The figure shows the reconstruction of images using 5 WTA networks of 100 neurons each as the encoding stage. Each column above contains the following sub-columns left-to-right: The input image (which is also the target image), The incoming synaptic weights to the spiking WTA neuron in the 5 WTA networks, and average $q(t) = \sigma\left(\beta\left(u(t) - u_0\right)\right)$, which is the predicted probability of a pixel taking the value 1.

	Number of WTA circuits		
No. of images	1	3	5
1200	165	144	137

Table 2: Mean negative log likelihood in nats (lower is better) of the target MNIST images. As a comparison, the log likelihood when the L5P neuron is replaced with pure logistic regression, trained on 1200 images and 5 WTAs, is 126.3

task are shown in figure 3, including the input, the weights learned by the WTAs, the spiking rate of the distal dendritic compartment given a WTA-encoded version of the input and the value of the quantity q which corresponds to the predicted probability of a pixel taking the value 1.

3.3 Sequence Prediction

We now explore if the L5P neurons trained using the SLR learning rule can be used to decode signals with a temporal structure. Specifically, we use the L5P neurons as decoders in the task of predicting sequences of images of rotating bars. The network used here is shown in figure 1D. The setup is similar to the one used for the MNIST digit reconstruction task (section 3.2 and figure 1C), but with the addition of lateral recurrent connections in the WTA circuit representing L2/3 neurons. These recurrent weights are trained using the learning rule that is an extension of the SEM rule for recurrent connections [Kappel et al., 2014]. This enables the WTA activity to learn a hidden markov model representation of the sequence of input images – at each point in time, the neuron that spikes represents a particular hidden state of the markov chain that represents the current sequence.

The input provided to the WTA is a rate-coded representation of the images of the rotating bars. Each image is a 17×17 image of a bar at a particular orientation. There are two sequences of bars – one where the bars rotate clockwise over time, and another where they rotate anticlockwise. Each image in the sequence is provided as input for a duration of $80ms$. The target to the L5P neurons, identical to the case of MNIST digit reconstruction, is provided by clamping the firing rate of the L5P neurons to the rate coded representation of the input at that time.

During testing, the network is shown part of the sequence of images of the rotating bars, and is required the complete the sequence by predicting the input for the remainder of the sequence. The reconstructions of predicted sequence states are shown in figure 4. Halfway through the sequence, the input image is repeated to force the network to take the input into consideration when predicting HMM state. After that, the input is stopped, while the network is expected to output the remaining part of the sequence with the bar rotating clockwise or anticlockwise. From the figure, it can be seen that the network is able to complete the sequence – the recurrent WTA learns to represent the remainder of the sequence in its state while the L5P neuron decodes these states into image pixels.

4 Methods

4.1 Proof that rule performs logistic regression

In this section, we demonstrate that the rule described in equation 4 performs stochastic gradient descent on the logistic loss between inputs $\mathbf{x}(t)$ and targets y.

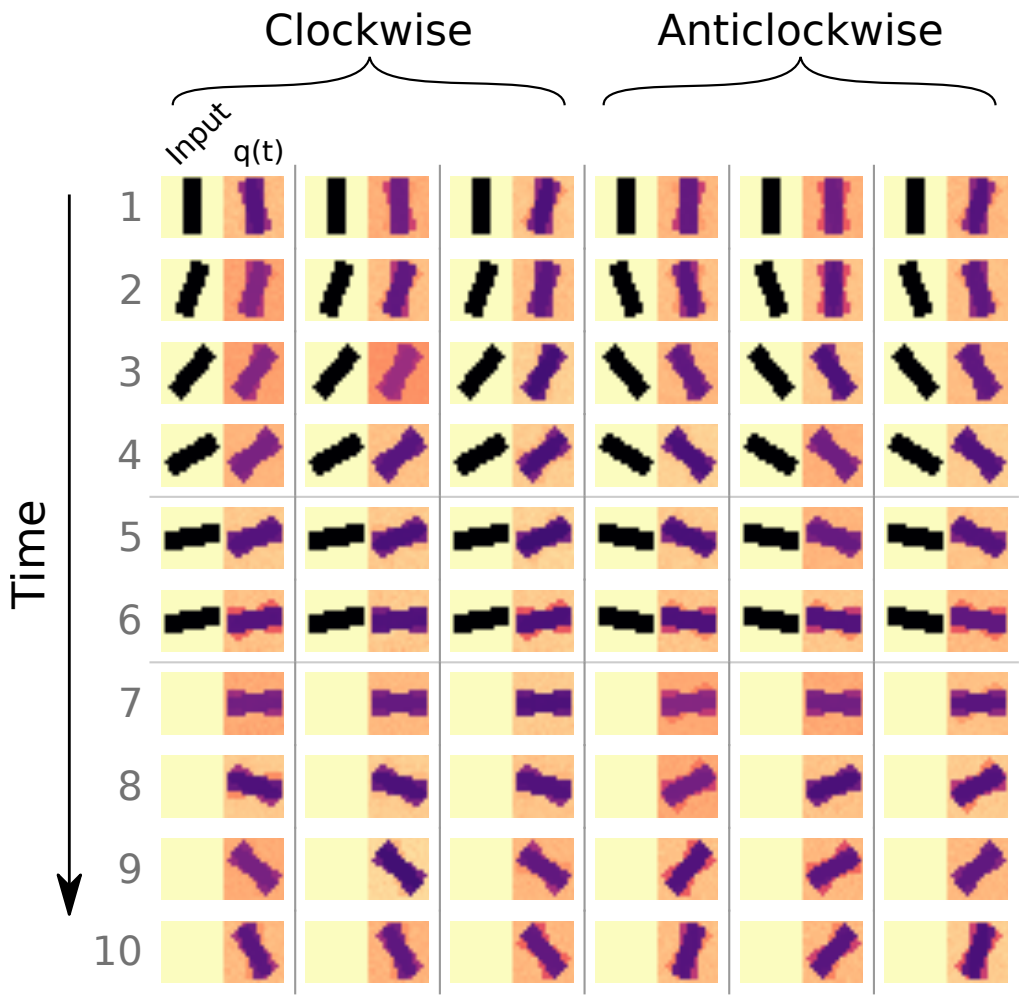

Figure 4: Reconstruction of the images of the rotating bars in the presence of input, and without the presence of input (i.e. when the WTA network predicts the subsequent states). Each image contains the follwoing images left to right. First (left), is the input image. Second (right) is the image containing the mean of $q(t)$ over the $80ms$ interval for each pixel. Each column represents a different presentation of a sequence of rotating bars, with the first three being clockwise and the next three anticlockwise. Time progresses from top to bottom with each image representing an 80ms interval (indexed 1-10). In time intervals 5, 6 we provide the same input thereby breaking the natural pattern of sequences, which forces the network to take the input into consideration when predicting the HMM state. From intervals 7 onwards we withhold input to the WTA network, thereby allowing it to predict the next states purely based on the current state (i.e. without input).

First, we establish the following. $\mathbf{x}, y$ are random variables representing a data point and the corresponding target respectively. We assume that target y is a bernoulli random variable meaning that $y \in \{0, 1\}$. When performing our simulations, each data-point is represented as follows. The independent variable data point $\mathbf{x}$ is fed as a spike train and converted to the vector $\mathbf{x}(t)$, which are the corresponding presynaptic activations whose weighted sum forms the input to the distal dendritic compartment. The binary targets $y \in \{0, 1\}$ are represented by the somatic firing rate $\rho_s(t)$. In accordance with the binary nature of the variable y, the somatic compartment spikes with a high spike rate $\rho_{target,high}$ when $y = 1$, and conversely, a low spike rate $\rho_{target,low}$ when $y = 0$. The precise meaning of the term high and low becomes clear when explaining the assumptions required for the proof below.

With the above setup, the average weight change over time caused by the learning rule in equation 4 is given by

$$E_t\left[\Delta \mathbf{w}\right] = E_t\left[\eta\left(\frac{z(t)}{q(t)} - 1\right)\ s_{\text{nmda}}(t)\ \mathbf{x}(t)\right]$$

Now, we take into account the method of providing the data. Each sample $\mathbf{x}_i, y_i$ of the random variables $\mathbf{x}, y$ is converted to a time series $\mathbf{x}_i(t), \rho_{si}(t)$ of the same length. What this means is that we may think of $\mathbf{x}(t)$, $z(t)$, and $s_{\text{nmda}}(t)$ as random variables that are sampled in a ergodic manner and therefore we can convert the time average into an expectation over these random variables

$$\begin{aligned}
E_t\left[\Delta \mathbf{w}\right] &= E_t\left[\eta\left(\frac{z(t)}{q(t)} - 1\right)\ s_{\text{nmda}}(t)\ \mathbf{x}(t)\right] \\
&= E_{\mathbf{x}(t),\, y,\, s_{\text{nmda}}(t),\, z(t)}\left[\eta\left(\frac{z(t)}{q(t)} - 1\right)\ s_{\text{nmda}}(t)\ \mathbf{x}(t)\right] \\
&= E_{\mathbf{x}(t),\, y}\left[E_{s_{\text{nmda}}(t)|\ \mathbf{x}(t),\, y}\left[E_{z(t)|\ \mathbf{x}(t),\, y,\, s_{\text{nmda}}(t)}\left[\eta\left(\frac{z(t)}{q(t)} - 1\right)\right] s_{\text{nmda}}(t)\right] \mathbf{x}(t)\right]
\end{aligned}$$

We first observe that updates are only non-zero when there is an NMDA spike. Hence we can simplify the above to

$$E_t\left[\Delta \mathbf{w}\right] = E_{\mathbf{x}(t),\, y}\left[E_{s_{\text{nmda}}(t)|\ \mathbf{x}(t),\, y}\left[\eta\left(\frac{p\left(z(t) = 1 | s_{\text{nmda}} = 1\right)}{q(t)} - 1\right) s_{\text{nmda}}(t)\right] \mathbf{x}(t)\right]$$

Where $p\left(z(t) = 1 | s_{\text{nmda}} = 1\right)$ is the probability that a particular NMDA spike causes a Ca^{2+} spike through conicidence with a somatic spike, or is generated during an existing Ca^{2+} spike. This probability depends on the NMDA spike rate (which depends on $\mathbf{w}^T\mathbf{x}(t)$) as well as the somatic spike rate $\rho_s(t)$. However we note that we are dealing with binary targets $y \in \{0, 1\}$. Thus, we assume that when $y = 0$, The somatic spiking rate $\rho_{target,low}$ is so low that the probability of an NMDA spike being coincident with a somatic spike, and consequently the probability of a calcium spike being generated is near zero. That is, we assume that when $y = 0$, $p\left(z(t) = 1 | s_{\text{nmda}} = 1\right) \approx 0$. Conversely, we assume that when $y = 1$, $\rho_{target,high}$ is large enough that $p\left(z(t) = 1 | s_{\text{nmda}} = 1\right) \approx 1$. What this assumption means is that we can approximate $p\left(z(t) = 1 | s_{\text{nmda}} = 1\right) \approx y$. We also use the fact that $E_{s_{\text{nmda}}(t)|\ \mathbf{x}(t), y}\ \left[s_{\text{nmda}}(t)\right] = \rho(t)$.

$$
\begin{aligned}
E_t\left[\Delta \mathbf{w}\right] &= E_{\mathbf{x}(t),\, y}\left[E_{s_{\text{nmda}}(t)|\,\mathbf{x}(t),\, y}\left[\eta\left(\frac{y}{q(t)}-1\right)s_{\text{nmda}}(t)\right]\mathbf{x}(t)\right] \\
&= E_{\mathbf{x}(t),\, y}\left[\eta\left(\frac{y}{q(t)}-1\right)\rho(t)\mathbf{x}(t)\right] \\
&= E_{\mathbf{x}(t),\, y}\left[\eta\rho_{max}\left(y-q(t)\right)\ \mathbf{x}(t)\right] \\
&= E_{\mathbf{x}(t),\, y}\left[\eta\rho_{max}\left(y-\sigma\left(\beta\left(\mathbf{w}^T\mathbf{x}(t)-u_0\right)\right)\right)\ \mathbf{x}(t)\right] \qquad (11)
\end{aligned}
$$

Here we have used the definition of $q(t)$ in equation 5 to simplify.

One recognizes the update in equation 11 to be the gradient descent update of the logistic loss between the independent $\mathbf{x}(t)$, and targets y (i.e. equation 6). We also see that $q(t)$ learns to estimate the likelihood of the target being 1 given the input. Since $\mathbf{x}(t)$ is proportional to the actual inputs x, we conclude that the weight update algorithm performs logistic regression on inputs $\mathbf{x}$ with target y.

4.2 MNIST digit classification

The network structure is described in figure 1B. The input is binarized MNIST digits that are rate coded – one input neuron per pixel of the MNIST image is used and each neuron spikes with a low firing rate ($\rho_{inp,low}$ = 2Hz) or high firing rate ($\rho_{inp,high}$ = 100Hz) depending on whether the corresponding pixel is 0 or 1 respectively. There are 784 (28×28) input neurons, one per pixel, each one with a firing rate determined by the value of the particular pixel from the input image. The classifier consists of 10 L5P-neurons with each neuron responsible for predicting the likelihood of the input being a particular digit. Each classifier is trained to distinguish one digit class from the rest. That is, a particular L5P neuron only receives target signal as 1 if the label corresponding to the current input is the digit it is being trained to distinguish. The somatic rate is clamped at $\rho_{target,high}$ if the target is 1 and to $\rho_{target,low}$ otherwise, where $\rho_{target,high} = 100$ Hz, and $\rho_{target,low} = 2$ Hz. The incoming weights to the L5P neurons are trained using the SLR learning rule. During testing the predicted label is inferred as the label corresponding to the classifier with the highest firing rate at the distal dendrite.

4.3 Reconstruction of MNIST digits

In this experiment we have 3 layers of neurons:

The Input Layer: This layer corresponds to the sensory input going into the cortical laminae. This consists of 2 sets of 784 (28×28) neurons that fire spikes in a population coded manner. There are 2 neurons per pixel, one of which fires at rate $\rho_{inp,high}$ if the input pixel is 1 and $\rho_{inp,low}$ if the input pixel is 0, the other neuron is the inverse, where $\rho_{inp,high} = 100$ Hz, and $\rho_{inp,low} = 2$ Hz. Each such pair of neurons have a constant total firing rate $\rho_{inp} = \rho_{inp,low} + \rho_{inp,high}$. This is necessary to form the population coded input as required by the Spike-based Expectation Maximization algorithm described in [Nessler et al., 2013].

The WTA Layer: This layer corresponds to L2/3, with lateral inhibition. It is implemented using multiple winner take-all (WTA) circuits where each WTA consists of 100 neurons, at its input connections are learned independently using a local learning rule that does stochastic expectation maximization (SEM) [Nessler et al., 2013]. Specifically, in the SEM learning rule, the weight between input neuron i and WTA neuron k is updated with:

$$\Delta w_{ki} = \left(\, y_i\, c\, e^{-w_{ki}} - 1\right) h_k$$

where y_i is the state of the input neuron, c is a constant, and h_k is the state of the WTA neuron. The input to the WTAs come from the input layer.

The L5P Layer: This is the decoding layer of the simulation containing the L5P neurons described above. It consists of one L5P neuron per pixel, that are trained as classifiers. Each input data point of the classifier is the spike response of the WTA Layer to a specific MNIST digit. The corresponding target is the value of that pixel in the target image.

The network setup is shown in figure 1C. Connections between any two layers are all-to-all.

Training: During training, we first train each WTA independently using SEM as described above. Then we feed the input images to the trained WTA network, and feed the spikes of the WTA layer as input to the L5P Layer. The corresponding target image is used to train the L5P neuron pixel classifiers. Training is only done for pixels which take both values 0 and 1 across the set of target images. The learning rate decays proportional to $1/t$ in accordance to the theoretically optimum decay rate for stochastic gradient descent.

Testing: With both sets of weights trained test the generation of the target images upon providing test MNIST images. The test set of images of all experiments contains 100 images that have no overlap with the training set.

In order to determine the value of a pixel in the L5P Layer we compare the average value of firing rate ρ over the duration of presentation of the input image. If $\rho > \rho_0$ then it is assigned 1 else 0. Note that this average is calculated not using the spikes but directly averaging the value of ρ over the relevant duration.

4.4 Sequence Prediction of rotating bars

In this experiment we have 3 layers of neurons.

The Input Layer: The input consists of 2 sequences of images of oriented bars, one rotating clockwise and another anti-clockwise. The input images that are fed into the input layer are 17×17 pixeled images of a centered bar in different particular orientation The bar is of dimensions 15×5. Each pixel holds a binary value of zero or one, where pixels correponding to positions within the bar are one.

The input layer corresponds to the sensory input going into the cortical laminae. It consists of a set of 289 (17×17) neurons, one per pixel. Each input neuron fire spikes in a rate coded manner, with firing rate $\rho_{inp,high} = 40$Hz when the pixel is 1, and rate $\rho_{inp,low} = 0$ Hz when the pixel is 0. Each image in the sequence is provided as input for a duration of $80ms$.

The WTA Layer: This layer corresponds to L2/3, with lateral inhibition. It consists of a WTA circuit with 100 neurons and with recurrent connections as described in [Kappel et al., 2014]. The input connections are trained using the same SEM learning rule as described in section 4.3. The recurrent weight v_{kj} between WTA neurons j and k is trained using the following rule from [Kappel et al., 2014]:

$$\Delta v_{kj} = \left(h_j \, c \, e^{-v_{kj}} - 1 \right) h_k$$

where h_j and h_k are the states of two distinct WTA neurons, and c is a constant. The input to the WTAs come from the input layer.

After training using the SEM algorithm, it is seen that each neuron in the WTA layer specializes to respond to a particular state of a hidden markov model that encodes the input sequences.

The L5P Layer: This is the decoding layer of the simulation containing the L5P neurons described above, and is constructed identically to the layer used to reconstruct binarized MNIST digits. It consists of one L5P neuron per pixel, which are trained as classifiers for the pixel value.

Each input data point of the classifier is the spike response of the WTA Layer to a specific frame of the rotating bar. The corresponding target for a L5P neuron is the value of the corresponding pixel in the target frame.

The network used here is shown in figure 1D, with connections between any two layers being all-to-all.

Training: We train the input weights to the WTA as described above. Then we feed the input images to the trained WTA network, and feed the spikes of the WTA layer as input to the L5P Layer. The corresponding target image is used to train the L5P neuron pixel classifiers.

Testing: With both sets of weights trained, the reconstruction of the input is tested where only half the sequence of input is presented to the network. The values of pixes are determined in the same way as in section 4.3.

4.5 Parameters Used

Neuron Model: The parameters for the L5Py neurons are consistent across the experiments in this paper. The Input PSP's are calculated using a double-exponential PSP kernel with a rise time constant $\tau_{rise} = 2ms$ and a falling time constant $\tau_{fall} = 10ms$. The parameters used to calculate the spike rate of NMDA spikes $\rho(t)$ (via equation 1) are: $\rho_{max} = 400$Hz $\beta = 0.1$, $u_0 = 6.0$.

Parameters for SLR: The parameters for the SLR rule for the above three experiments are given in table 3

	Experiment		
Parameter	MNIST Classification	MNIST Reconstruction	Sequence Prediction
τ_{Ca}	100.0	100.0	50.0
τ_c	20.0	15.0	50.0
η_0	0.08	0.5	0.9
η_{final}	0.002	0.05	0.09
$\rho_{target,high}$	50 Hz	50 Hz	50 Hz
$\rho_{target,low}$	1 Hz	1 Hz	1 Hz

Table 3: Parameters related to SLR that are used for each experiment

5 Acknowledgement

The research for this paper was partially supported by the Human Brain Project, grant 785907, of the European Union.

References

[Amitai et al., 1993] Amitai, Y., Friedman, A., Connors, B., and Gutnick, M. (1993). Regenerative activity in apical dendrites of pyramidal cells in neocortex. *Cerebral cortex*, 3(1):26–38.

[Binzegger et al., 2004] Binzegger, T., Douglas, R. J., and Martin, K. A. (2004). A quantitative map of the circuit of cat primary visual cortex. *Journal of Neuroscience*, 24(39):8441–8453.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[Kampa et al., 2006] Kampa, B. M., Letzkus, J. J., and Stuart, G. J. (2006). Requirement of dendritic calcium spikes for induction of spike-timing-dependent synaptic plasticity. *The Journal of physiology*, 574(1):283–290.

[Kampa et al., 2007] Kampa, B. M., Letzkus, J. J., and Stuart, G. J. (2007). Dendritic mechanisms controlling spike-timing-dependent synaptic plasticity. *Trends in Neurosciences*, 30(9):456–463.

[Kappel et al., 2014] Kappel, D., Nessler, B., and Maass, W. (2014). STDP Installs in Winner-Take-All Circuits an Online Approximation to Hidden Markov Model Learning. *PLOS Computational Biology*, 10(3):e1003511.

[Kuhn et al., 2008] Kuhn, B., Denk, W., and Bruno, R. M. (2008). In vivo two-photon voltage-sensitive dye imaging reveals top-down control of cortical layers 1 and 2 during wakefulness. *Proceedings of the National Academy of Sciences*, 105(21):7588–7593.

[Larkum, 2013] Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151.

[Larkum et al., 2009] Larkum, M. E., Nevian, T., Sandler, M., Polsky, A., and Schiller, J. (2009). Synaptic Integration in Tuft Dendrites of Layer 5 Pyramidal Neurons: A New Unifying Principle. *Science*, 325(5941):756–760.

[Larkum and Zhu, 2002] Larkum, M. E. and Zhu, J. J. (2002). Signaling of layer 1 and whisker-evoked ca2+ and na+ action potentials in distal and terminal dendrites of rat neocortical pyramidal neurons in vitro and in vivo. *Journal of neuroscience*, 22(16):6991–7005.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Major et al., 2013] Major, G., Larkum, M. E., and Schiller, J. (2013). Active Properties of Neocortical Pyramidal Neuron Dendrites. *Annual Review of Neuroscience*, 36(1):1–24.

[Nessler et al., 2013] Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity. *PLOS Computational Biology*, 9(4):e1003037.

[Nieuwenhuys, 1994] Nieuwenhuys, R. (1994). The neocortex. *Anatomy and embryology*, 190(4):307–337.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Schiller et al., 1997] Schiller, J., Schiller, Y., Stuart, G., and Sakmann, B. (1997). Calcium action potentials restricted to distal apical dendrites of rat neocortical pyramidal neurons. *The Journal of physiology*, 505(3):605–616.

[Yuste et al., 1994] Yuste, R., Gutnick, M. J., Saar, D., Delaney, K. R., and Tank, D. W. (1994). Ca2+ accumulations in dendrites of neocortical pyramidal neurons: an apical band and evidence for two functional compartments. *Neuron*, 13(1):23–43.