A normative framework for learning top-down predictions through synaptic plasticity in apical dendrites

Arjun Rao[*,1], Robert Legenstein[*,1], Anand Subramoney[1,2], and Wolfgang Maass[1,+]

[*]First authors

[1]Institute for Theoretical Computer Science, Graz University of Technology, 8010, Austria
[2]Institute for Neural Computation, Ruhr University Bochum, Germany
[+]Corresponding author: maass@igi.tugraz.at

April 30, 2021

Abstract

Predictive coding has been identified as a key aspect of computation and learning in cortical microcircuits. But we do not know how synaptic plasticity processes install and maintain predictive coding capabilites in these neural circuits. Predictions are inherently uncertain, and learning rules that aim at discriminating linearly separable classes of inputs – such as the perceptron learning rule – do not perform well if the goal is learning to predict. We show that experimental data on synaptic plasticity in apical dendrites of pyramidal cells support another learning rule that is suitable for learning to predict. More precisely, it enables a spike-based approximation to logistic regression, a well-known gold standard for probabilistic prediction. We also show that data-based interactions between apical dendrites support learning of predictions for more complex probability distributions than those that can be handled by single dendrites. The resulting learning theory for top-down inputs to pyramidal cells provides a normative framework for evaluating experimental data, and suggests further experiments for tracking the emergence of predictive coding through synaptic plasticity in apical dendrites.

1 Introduction

A hallmark of brain intelligence is its capability of using internal models, predictions, and internal goals that are maintained in higher brain areas to modulate, and arguably optimize, sensory processing in lower brain areas. This ubiquitous integration of context into sensory processing distinguishes brain computation from most other computing paradigms. Furthermore, we assume that the stereotypical laminar structure of cortical microcircuits supports this particular style of brain computation. In particular, inputs from higher brain areas preferentially arrive in pyramidal cells through synaptic connections in apical dendrites that are located in the upper layers of the laminar microcircuit. Hence in order to understand how context information from higher brain areas shapes sensory processing, we need to understand the plasticity mechanisms in apical dendrites of these pyramidal cells. One intriguing hypothesis is that pyramidal cells learn to predict bottom-up sensory inputs on the basis of information from other brain areas, including motor areas [Leinweber et al., 2017], that reaches these neurons through synapses

on their apical dendrites [Xu et al., 2012, Larkum, 2013, Doron et al., 2020]. This hypothesis implies that plasticity of synapses at distal dendrites plays a key role for the emergence of predictive coding. Interestingly, a number of experimental data suggest that plasticity processes in apical dendrites differ in essential aspects from the more frequently studied correlative Hebbian or spike-timing-dependent plasticity (STDP)-like rules at other synaptic connections to pyramidal cells. In particular two differences have been emphasized: Firstly, the firing of the postsynaptic neurons, i.e., its somatic spikes, do not appear to play an essential role in the induction of this synaptic plasticity [Kampa et al., 2006, Kampa et al., 2007, Gambino et al., 2014]. Secondly, plasticity of these synapses is gated instead by other instructive signals [Magee and Grienberger, 2020], such as Ca^{2+}-spikes [Kampa et al., 2006, Kampa et al., 2007], NMDAR spikes [Gambino et al., 2014], more general forms of plateau potentials within the dendritic arbor of pyramidal cells in area CA1 [Bittner et al., 2015, Bittner et al., 2017, Magee and Grienberger, 2020], or synaptic input from the perirhinal cortex [Doron et al., 2020]. These gating signals for synaptic plasticity have the common property that they are longer lasting events than postsynaptic spikes, extending over at least 50 ms. Hence we will refer to them as instructive signals in line with the terminology of [Magee and Grienberger, 2020].

In the case of Ca^{2+}-spikes as instructive signals for synaptic plasticity, these Ca^{2+}-spikes are typically generated when strong synaptic input to apical dendrites coincides with somatic spikes of a thick tufted pyramidal cell in layer 5. These somatic spikes may be caused by bottom-up sensory inputs, that arrive at synapses located below the dendritic tuft [Larkum, 2013]. Hence, prediction of Ca^{2+}-spike can be seen for these neurons as a proxy for predicting sensory inputs.

A theoretical framework for analyzing these rules for synaptic plasticity at apical dendrites is missing, and a primary goal of this article is to establish such a framework. In addition, we propose a normative model for this form of synaptic plasticity. This normative model is based on the commonly conjectured learning goal for this synaptic plasticity: learning to predict. It takes into account that prediction is inherently uncertain, and can therefore not be learned well with learning rules such as the perceptron learning rule that is designed for learning classification of linearly separable classes of inputs. Furthermore, predictions are most useful if they are accompanied with an estimate of their uncertainty. For that one has to resort to models for learning probabilistic prediction, rather than classification.

We will first present a theoretical framework for learning probabilistic predictions through synaptic plasticity in a single dendritic branch in the apical tuft of a pyramidal cell, ignoring its interaction with other dendritic branches. We then extend the framework to the case where there are several dendritic branches in the apical tuft. There, we consider two types of experimentally reported interactions between dendritic branches, a saturating summation [Naud et al., 2014, Poirazi et al., 2003], as well as a soft XOR for pyramidal cells on layer 2/3 of the human cortex [Gidon et al., 2020]. We include illustrative examples for computational tasks that can be learned with the proposed model for synaptic plasticity in apical dendrites, both for the case of a single branch, and for the case of several interacting branches. For simplicity, we will refer to those dendritic branches in the apical tuft of pyramidal cells in layer 2/3 or 5, to which our learning theory applies, just as apical dendrites.

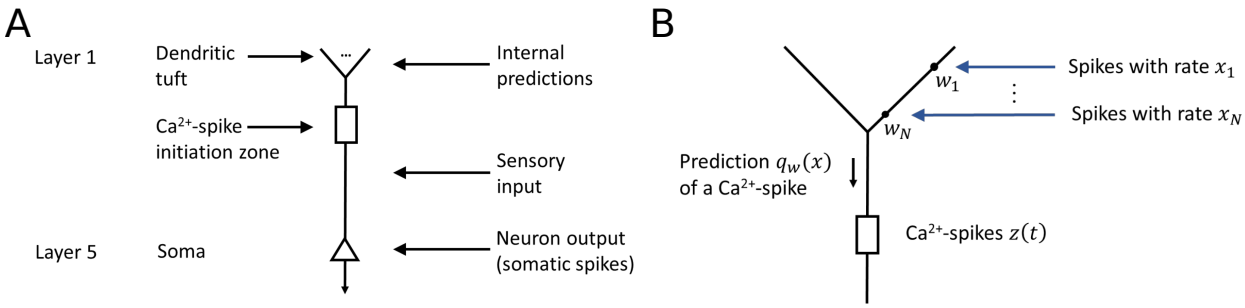

Figure 1: Scheme of a pyramidal cell with apical dendrites and Ca^{2+}-spike initiation zone, and setup of the normative learning model. A) Generic model of a thick tufted pyramidal cell in layer 5 of the stereotypical laminar cortical microcircuit, integrating two different inputs streams, see [Larkum, 2013] for details. A Ca^{2+}-spike, which also propagates backwards into the dendritic tuft, is generated at the Ca^{2+}-spike initiation zone when sufficiently strong depolarization arises from both input streams. This Ca^{2+}-spike serves as the instructive signal in the sense of the plasticity model of [Magee and Grienberger, 2020] **B)** Illustration of synaptic input and output of the normative learning model, for the simplest case of a single dendritic branch. Synaptic input from neurons in higher brain areas with firing rates $x_1, \ldots, x_N$ is integrated with synaptic weights $w_1, \ldots, w_N$ in an apical dendrite. The normative learning model addresses plasticity of these weights, and the resulting depolarization $q_{\mathbf{w}}(\mathbf{x})$ in the dendritic tuft represents the resulting prediction after learning.

2 Results

2.1 A theoretically tractable model for prediction learning through synaptic plasticity in apical dendrites

We are considering a model for synaptic plasticity in apical dendrites that aims at being consistent with the main body of experimental data, and simultaneously at being as simple as possible so that it can be analyzed theoretically. Our model is closely related to the simple model for computations in apical dendrites of layer 5 pyramidal cells (L5Ps) proposed in [Larkum, 2013], see Fig. 1A. We have added to that a simple model for synaptic plasticity in these dendrites, see Fig. 1B.

The vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ represents the current firing rates of N neurons in higher brain areas that provide synaptic input to this dendritic branch. We write $\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$ for the synaptic weights of these synapses. We use the variable z to indicate the presence of a gating signal for synaptic plasticity, with $z(t) = 1$ if synaptic plasticity is enabled at time t, else $z(t) = 0$. This gating signal is referred to as the instructive signal in [Magee and Grienberger, 2020].

Since the analysis of synaptic plasticity in apical dendrites of these neurons [Kampa et al., 2006, Kampa et al., 2007] focused on the Ca^{2+}-spike as a gating factor for synaptic plasticity in L5Ps, and since L5Ps are used as a running example for our analysis, we will refer in the following to this gating factor as a Ca^{2+}-spike. But our analysis also holds if synaptic plasticity in apical dendrites is gated by a different signal, such as input from the perirhinal cortex according to [Doron et al., 2020], or a plateau potential in pyramidal cells of area CA1 that may emerge stochastically or triggered by attention or salience signals [Bittner et al., 2015, Bittner et al., 2017, Magee and Grienberger, 2020]. A common property of all these experimentally reported gating signals is that they last for 50 ms or longer.

We assume that the goal of synaptic plasticity of the weights $\mathbf{w}$ of synapses on the apical dendrite is the prediction of the gating signal z. Thus if this gating signal is triggered for example in pyramidal

Gating signal z	weight update $\Delta\mathbf{w}$
0	$-\ q_{\mathbf{w}}(\mathbf{x})\ x_j$
1	$x_j\ -\ q_{\mathbf{w}}(\mathbf{x})\ x_j$

Table 1: Updates for synaptic weight w_j according to the Dendritic Logistic Regression (DLR) rule: The firing rates of presynaptic neurons in higher brain areas are denoted by $\mathbf{x}$, and z is the gating signal. $q_{\mathbf{w}}(\mathbf{x})$ is a function of the local dendritic membrane potential defined in equation 2.

cells in layer 5 when the synaptic input from higher brain areas to the apical dendrites coincides with sensory inputs, this learning goal amounts to predicting coinciding sensory input to the neuron, see [Larkum, 2013]. Furthermore, we assume that the apical dendrites produce this prediction in the form of the depolarization $q_{\mathbf{w}}(\mathbf{x})$ of the membrane potential that results from the current synaptic inputs $\mathbf{x}$ with weights $\mathbf{w}$ onto these dendrites. If one denotes the joint distribution of the input to the considered dendritic branch by $\mathbf{x}$ and the Ca^{2+}-spike z by $p_{\text{target}}(\mathbf{x}, z)$, then the optimal value of the model estimate $q_{\mathbf{w}}(\mathbf{x})$ is the conditional probability $p_{\text{target}}(z|\mathbf{x})$. An important computational role of the resulting contribution $q_{\mathbf{w}}(\mathbf{x})$ of distal dendrites to the processing of sensory information by a pyramidal cell in layer 5 was elucidated in [Takahashi et al., 2016] and [Takahashi et al., 2020]: it shifts the perceptual threshold.

Since Ca^{2+}-spikes and the other gating signals that we are considering have a duration of 50 ms or more, we are looking for a synaptic plasticity rule which is consistent with this time scale, i.e., with a time scale that is slower than that of single spikes and EPSPs. In particular, we focus on a realization of the prediction $q_{\mathbf{w}}(\mathbf{x})$ on this slower time scale: through the average membrane potential $\bar{u}$ that results from the firing rates $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ of neurons in higher brain areas that have formed synaptic connections on the considered apical dendrite. We approximate the resulting average membrane potential in this dendrite by:

$$\bar{u} = \sum_{j=1}^{N} x_j w_j = \mathbf{x}^T \mathbf{w}. \tag{1}$$

It turns out that the resulting plasticity rule assumes a particularly simple form if one uses the following functional form for $q_{\mathbf{w}}(\mathbf{x})$, which also takes into account a saturation effect:

$$q_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\beta\left(\bar{u} - u_0\right)\right), \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the standard sigmoid function, u_0 is a constant threshold, and β is a constant scaling factor.

In order to optimize $q_{\mathbf{w}}(\mathbf{x})$ as approximation of $p_{\text{target}}(z|\mathbf{x})$, we use the cross entropy loss as a standard measure of the distance between the two distributions. A useful property of this loss function is that it is convex. Gradient descent for this loss function yields the following normative rule for synaptic plasticity of the weights $\mathbf{w}$ of synapses onto the apical dendrite:

$$\Delta w_j \ = \ \eta\left(z - q_{\mathbf{w}}(\mathbf{x})\right) x_j \ , \tag{3}$$

where Δw_j is the change of the weight of synapse j and $\eta > 0$ is a constant scaling factor. We show in Section 3.2, that this weight update performs stochastic gradient descent for the previously described cross entropy loss. Because of the convexity of this loss function, this update is guaranteed

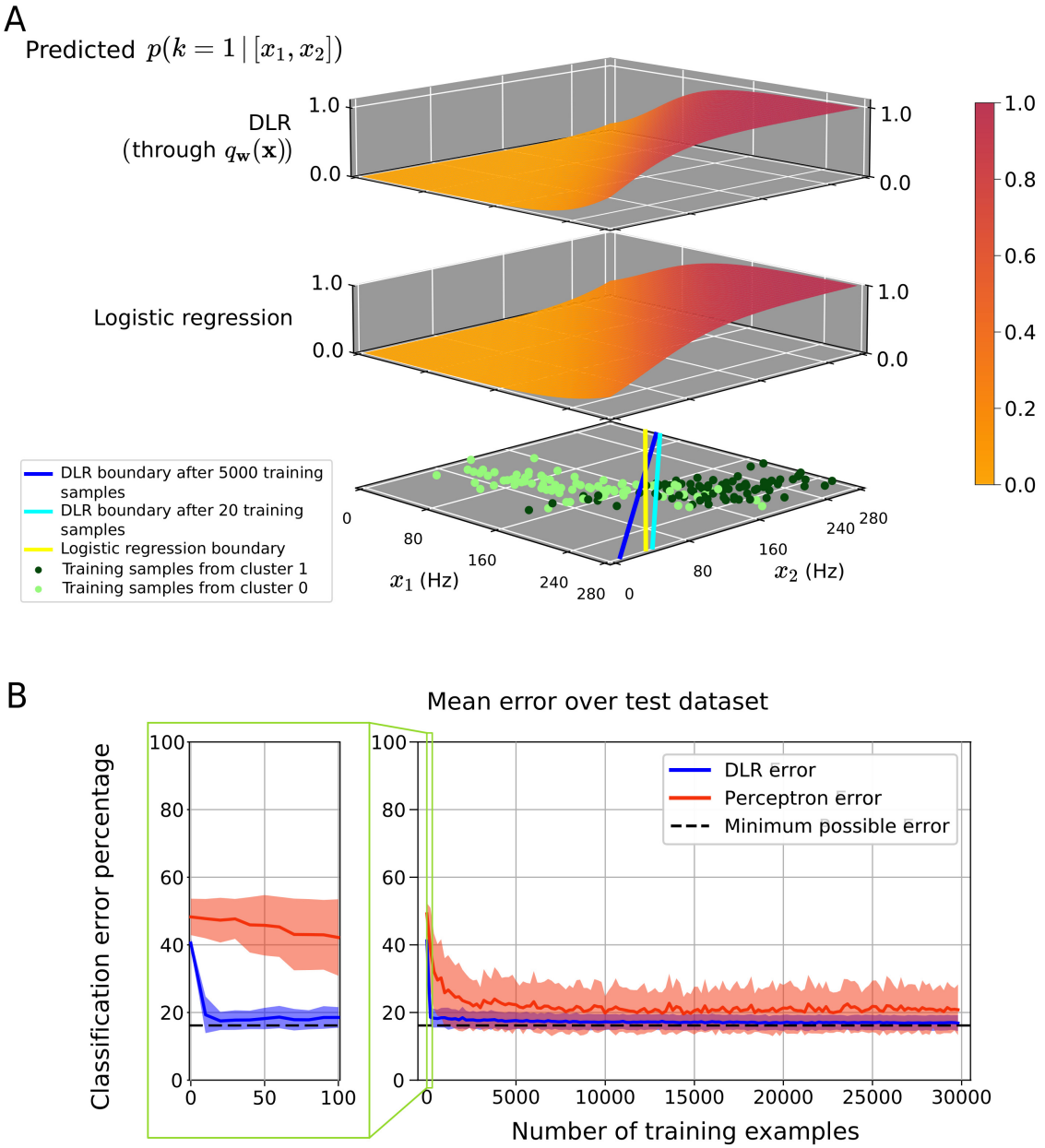

Figure 2: Illustration of emergent prediction capabilities of apical dendrites through DLR. A) Plot of probabilistic predictions for different firing rate patterns $\mathbf{x}$ of neurons in higher brain area that project to the considered apical dendrites, both as learned through DLR (at the top) and theoretically optimal predictions via logistic regression (below). The mean difference between the predicted probabilities is just 0.036. Also the resulting hyperplanes for resulting binary predictions for DLR and theoretically optimal logistic regression are very similar (bottom part). We also see a very rapid convergence of the DLR separation boundary after training with only 20 samples. **B)** Speed and accuracy of learning to guess (classify) from which cluster a presynaptic firing pattern $\mathbf{x}$ was drawn for the DLR learning rule and the perceptron learning rule, as considered in [Moldwin and Segev, 2020]. One sees that DLR enables much faster learning, and that the perceptron learning rule never converges to an optimal prediction performance. Mean and variance of the error are shown for 100 learning episodes with independently drawn initial weights and data points.

to converge to its global optimum. Hence this plasticity rule is in a well-defined sense optimal, and this analysis provides the foundation for our normative model. The rule can equivalently be described in the form of Table 1. Note that it contains a weight decay term that is activated when the apical dendrite is depolarized, i.e., $q_{\mathbf{w}}(\mathbf{x})$) is large, but there is no gating signal, i.e., $z(t) = 0$. Both the LTP and the LTD component of the plasticity rule are proportional to the presynaptic firing rate x_j.

Both the loss function that we minimize and the resulting type of plasticity rule are closely related to a well known optimization method in statistics and machine learning called Logistic Regression, see Methods. Hence we refer to the resulting synaptic plasticity rule as Dendritic Logistic Regression (DLR). For details regarding the implementation of this rule in simulations, see Methods.

2.1.1 A simple illustration of the capability of DLR to enable probabilistic predictions

Assume that there are two clusters of firing patterns $\mathbf{x}$ of neurons in higher brain areas that project to an apical dendrite of a pyramidal cell in layer 5, and that one can model each of these clusters by a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, with parameters $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$. Furthermore, assume that firing patterns $\mathbf{x}$ from cluster $k = 1$ predict that a subsequent sensory input will arrive at the target neuron (thereby causing a Ca^{2+}-spike in the neuron, i.e., a gating signal $z = 1$), and firing patterns from cluster $k = 0$ predict that this sensory input will not arrive (and that $z = 0$). Now assume that these two distributions overlap – as indicated by the clouds of dark green and light green dots in Fig. 2A. This occurs if many firing patterns $\mathbf{x}$ have significant probability under both of these distributions. In order to facilitate visualization, we consider here just two-dimensional firing patterns $\mathbf{x}$, but the same principle applies for input patterns of arbitrary dimension. These two point clouds are not linearly separable, and hence the perceptron learning rule (as investigated in [Moldwin and Segev, 2020]) cannot do a good job of separating them, as indicated by the red trace in Fig. 2B. In contrast, synaptic weights in the apical dendrite can learn this inherently uncertain prediction task very fast through DLR; see the blowup of results for the first few trials in Fig. 2B. This large performance difference between perceptron learning and DLR might seem surprising. The perceptron learning rule is similar to the DLR given in equation 3, where $q_{\mathbf{w}}(\mathbf{x})$ is replaced by $q_{\mathbf{w}}(\mathbf{x}) = 1$ if $\bar{u} \geq u_0$ and $q_{\mathbf{w}}(\mathbf{x}) = 0$ if $\bar{u} < u_0$. The more fine-grained dependence of the DLR on the branch depolarization (equation 2) greatly improves the convergence properties. Moreover the apical dendrites learn through DLR to predict the probability that a spike input pattern $\mathbf{x}$ predicts a subsequent sensory input through the resulting value of $q_{\mathbf{w}}(\mathbf{x})$ not only for those samples $\mathbf{x}$ that occurred during training, but they could generalize this probabilistic predictions to all potential values of presynaptic firing rates $\mathbf{x}$. Furthermore the resulting predicting capability was on the same level as the theoretically optimal logistic regression learning rule, as shown in Fig. 2A. The resulting prediction capability through DLR can be measured through the negative log likelihood (NLL) in nats (lower is better) for the training data. Across a 100 independent training runs, it achieves an average of 0.378 nats, compared to 0.361 nats achieved through logistic regression.

2.1.2 Predicting future sensory input caused by a moving object

Our brains are very good at estimating from the current position, speed, and movement direction of a moving object where this object will be at a later time point T. They can do this although estimates of movement speed and direction are inherently noisy, see the cartoon in Fig. 3B. We present a simple model which shows that this fundamental computational capability of neural circuits in the neocortex can be acquired through synaptic plasticity in apical dendrites. We assume that the receptive fields of a 2D array of neurons cover the movement field, and for simplicity, we represent the moving object

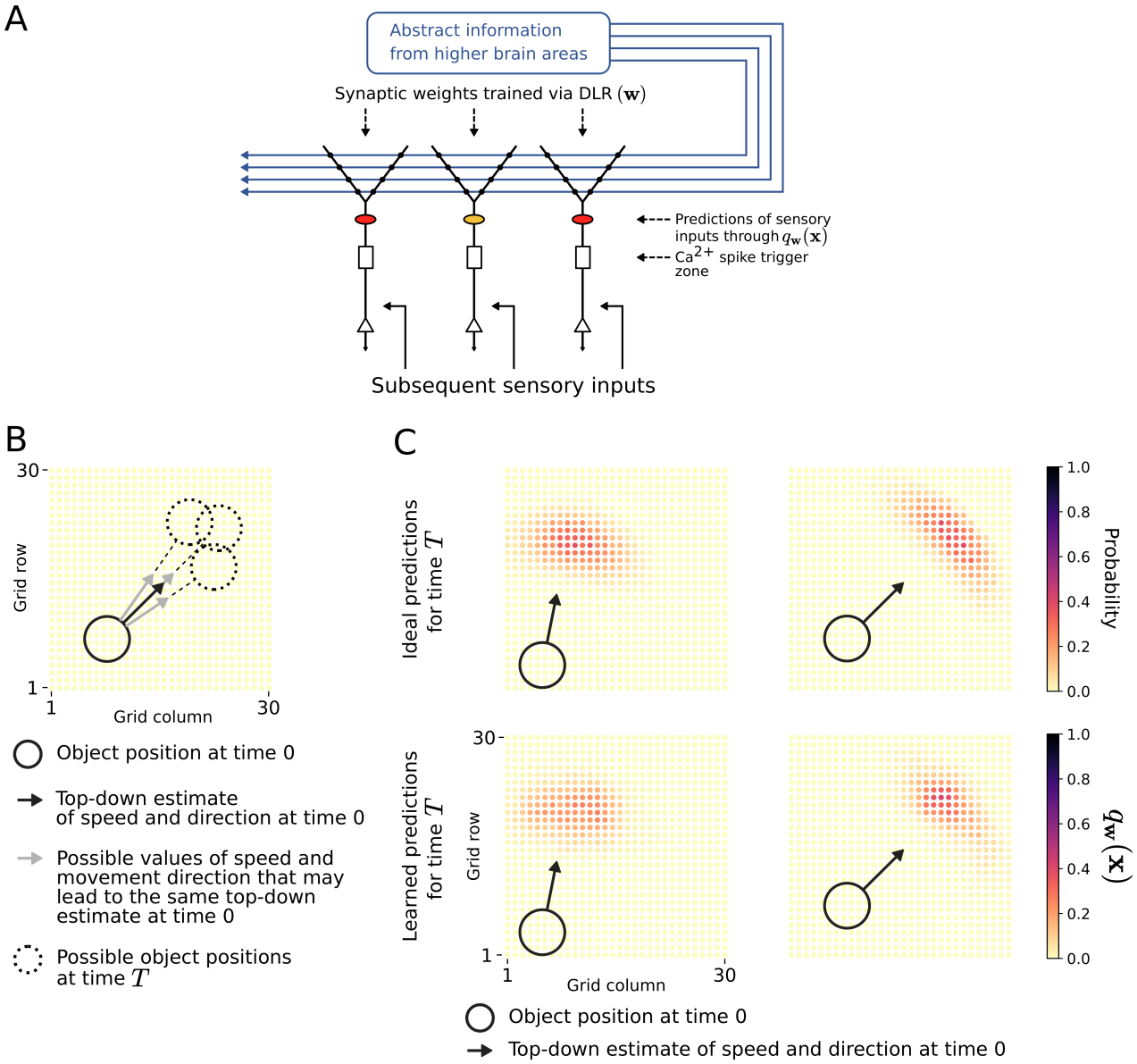

Figure 3: Performance of DLR for self-supervised learning to predict whether the receptive field of a neuron will be covered by a moving object. A) Schema of the simple network architecture that we consider. We assume that the position, speed, and movement direction of a moving object at time 0 are estimated in higher brain areas, and transmitted to the apical dendrites of a sheet of pyramidal cells, whose receptive fields cover the movement range by a 30×30 grid. More precisely, we assume that each of these neurons receives the same estimate. After learning, each apical tuft produces its individual probabilistic prediction, indicated by the same color code as used in panel C. **B)** For a particular top-down estimation of movement speed and direction, an illustration of three different possible trajectories of the moving object that may have resulted in this estimate. **C)** For two instances of this scenario we show in the lower row for each neuron in the 30×30 array its individual probabilistic prediction $q_{\mathbf{w}}(\mathbf{x})$ that it generates in its apical dendrites, after self-supervised learning with DLR. Note that all these neurons receive the same synaptic input $\mathbf{x}$ in their apical dendrites. For comparison, we show in the upper row for each receptive field the actual probability that it will be covered at time T by the moving object, computed with knowledge of the full distribution of noisy estimates for time 0.

by a linearly moving disk. The neurons in this array could for example be neurons in a visual or somatosensory area. We furthermore assume that these neurons receive from higher brain areas the estimated position, speed, and movement direction of a moving object at time 0, see Fig. 3A. We model the uncertainty of these estimates by adding Gaussian noise to an ideal estimate.

During learning, the neurons in this 2D array receive at time T both, these estimates from higher brain areas that arrive at their apical dendrites, and whether the receptive field of the neuron is covered at this time T by the moving object, as strong sensory inputs on their basal dendrites. We assume that the coincidence of both signals causes a Ca^{2+}-spike, thereby triggering in the apical dendrites of the neuron synaptic plasticity via DLR. We show in Fig. 3C that the neurons learn in this way to predict through the membrane potential $q_{\mathbf{w}}(\mathbf{x})$ generated by synaptic inputs $\mathbf{x}$ to their apical dendrites whether their receptive field will be covered at time T by the moving object. This holds whenever the input $\mathbf{x}$ to their apical contains the estimates of movement speed and direction at time 0. Hence when these estimates are provided before time T, the apical dendrites will produce a probabilistic prediction of a future event. These learnt predictions are shown in the lower row of Fig. 3C for two different movement scenarios. A comparison with the probabilistic ground truth in the upper row of Fig. 3C shows that the learnt probabilistic movement predictions are close to optimal. Note that the probabilistic ground truth in the upper row was computed on the basis of full knowledge of the distribution of the noisy estimates of speed and direction at time 0.

2.2 Extensions to the case of multiple dendritic branches

When there are several dendritic branches, the question arises as to how the contributions from different branches are combined for the prediction $q_{\mathbf{w}}(\mathbf{x})$. Here, we denote the weights and inputs into each dendritic branch B by $\mathbf{w}^B = \left[w_1^B, w_2^B, \ldots, w_N^B\right]^T$ and $\mathbf{x}^B = \left[x_1^B, x_2^B, \ldots, x_N^B\right]^T$ respectively, and the weights and inputs across all branches as $\mathbf{w}, \mathbf{x}$ respectively. We denote the predicted distribution by $q_{\mathbf{w}}(\mathbf{x})$ (similarly to equation 2), which estimates the target conditional distribution $p_{\text{target}}(z = 1|\mathbf{x})$.

It was shown in [Naud et al., 2014] that a neuron model with a somatic and dendritic compartment can accurately predict spike responses of cortical pyramidal neurons. Similar two-layer models have been proposed for hippocampal CA1 pyramidal cells [Poirazi et al., 2003] and even for cortical interneurons [Tzilivaki et al., 2019]. In general, it is believed that two-layer models provide a viable model for neocortical and hippocampal pyramidal cells [Jadi et al., 2014]. In this model the activity of each dendritic branch saturates according to a sigmoidal nonlinearity. Dendritic outputs are then summed and another non-linearity is applied to model the neuron output activity. In line with this model, we assume that the contributions from each dendritic branch (f^B) are a function of the mean branch local membrane potential $\bar{u}^B = \sum_i x_i^B w_i^B$ (equation 1) and consider a model of $q_{\mathbf{w}}(\mathbf{x})$ that accumulates contributions from the dendritic branches:

$$q_{\mathbf{w}}(\mathbf{x}) = \Psi\Big(\sum_B f^B(\bar{u}^B)\Big) \triangleq \Psi\left(f_{\text{total}}\right). \tag{4}$$

Consistent with the single-branch model (equation 2), the contribution of each branch is given by a sigmoidal function of its weighted input rate, i.e.

$$f^B = \sigma(\beta\left(\bar{u}^B - u_0\right)). \tag{5}$$

For the function Ψ we consider two possibilities. According to the classical two-layer model, the neuron output activity is given by a saturating nonlinearity applied to the summed dendritic contributions. Hence, as a first possibility, we model Ψ as a sigmoidal function in Subsection 2.2.1 below.

More recently, a more complex relationship was found for single human neurons [Gidon et al., 2020] which will be considered in Subsection 2.2.2.

2.2.1 Saturating model

In the first possibility, taking into consideration the thresholding and saturation dynamics of pyramidal neurons [Naud et al., 2014, Jadi et al., 2014], we chose the sigmoidal shape for Ψ to get

$$\Psi\left(f_{\text{total}}\right) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right), \tag{6}$$

where α and ϑ are suitable constants. Note that by choice of α and ϑ, one can model different types of (soft-) logic operations. A low ϑ lesser than 1 implements a soft OR operation (a minimum of one branch needs to be active to get a high probability output). A ϑ close to the number of branches implements a soft AND, and in-between we obtain a general threshold condition on the number of active branches. See Fig. 4B (left and middle) for an example with two branches.

The goal of the learning rule in this case is to get $q_{\mathbf{w}}(\mathbf{x})$ to predict the probability $p_{\text{target}}(z = 1|\mathbf{x})$ of Ca^{2+}-spikes that occurs during training. Hence a normative model for synaptic plasticity in these dendritic branches should minimize the distance between these two terms, formulated again as cross-entropy error.

The above choices for f^B and Ψ lead to the following weight update rule for the incoming weights to a particular dendritic branch B:

$$\Delta\, w_j^B = \eta\, \left(z - \Psi\left(f_{\text{total}}\right)\right)\, \dot{f}^B(\bar{u}^B)\, x_j^B, \tag{7}$$

where η is a learning rate and $\dot{f}^B$ is the derivative of f^B. This update is similar to the single-dendrite update, with two differences: (1) the term $\dot{f}^B(u^B)$ means that there is a nonlinear dependence on the local branch potential, and (2) the update depends on $\Psi\left(f_{\text{total}}\right)$. This quantity is not local to the dendritic branch. In order to minimize the information about $\Psi\left(f_{\text{total}}\right)$ at the branch, one can consider a gating of updates based on thresholds on $\Psi\left(f_{\text{total}}\right)$ which approximates the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ in the update rule 7. This leads to the following updates of synaptic efficacies:

$$\begin{aligned}
&\text{For } z = 0: \\
&\qquad \Delta\, w_j^B = \begin{cases} -\dot{f}^B(\bar{u}^B) x_j^B & , \quad \text{if } \Psi\left(f_{\text{total}}\right) \geq q^- \\ 0 & , \quad \text{otherwise} \end{cases} \\
&\text{For } z = 1: \\
&\qquad \Delta\, w_j^B = \begin{cases} \dot{f}^B(\bar{u}^B) x_j^B & , \quad \text{if } \Psi\left(f_{\text{total}}\right) \leq q^+ \\ 0 & , \quad \text{otherwise} \end{cases}
\end{aligned} \tag{8}$$

for plasticity thresholds q^- and q^+. To see the correspondence with rule 7, first consider the case without Ca^{2+}-spike ($z = 0$). Here, the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ in equation 7 is between -1 and 0. It becomes very small when $\Psi\left(f_{\text{total}}\right)$ is close to zero. This is approximated by a gating mechanism which blocks plasticity if $\Psi\left(f_{\text{total}}\right)$ is below some threshold q^- close to 0. When there is a Ca^{2+}-spike ($z = 1$), the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ is between 0 and 1. It becomes very small when $\Psi\left(f_{\text{total}}\right)$ is close to one. Hence, plasticity is blocked if $\Psi\left(f_{\text{total}}\right)$ is above some threshold q^+ close to 1 in the approximation 8.

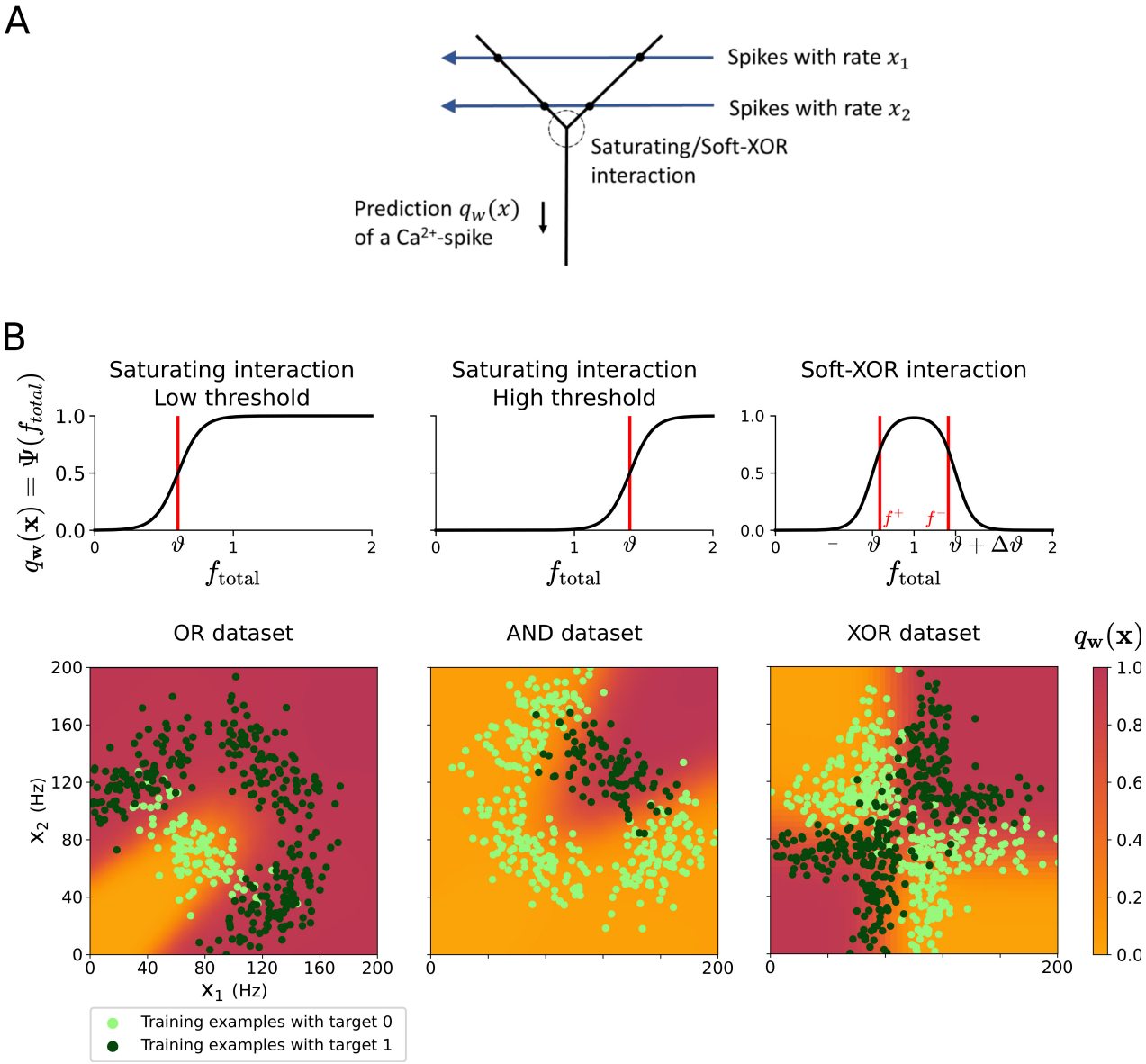

Figure 4: Illustration of emergent prediction capabilities through DLR of apical dendrites in the case of multiple branches in the dendritic tuft. A) We consider here for simplicity the case of two branches of the dendritic tuft that both receive synaptic input from the same two neurons in a higher brain area, for two experimentally reported types of interaction of their contributions: saturating and soft-XOR. **B)** The resulting probabilistic predictions $\Psi(f_{\text{total}})$ of the apical tuft are shown after training on three different datasets. Note that for neither one of these datasets a single-branch probabilistic prediction as shown in Fig. 2 suffices. But both experimentally observed saturating and XOR-like interactions between different dendritic branches enable DLR to produce suitable distributions of probabilities, indicated again through color coding, that separate the two point clouds. Note that like in the case of Fig. 2, a number of these training examples could have been generated by either one of the two distributions that generate dark green and light green datapoints. Hence probabilistic predictions are needed, like in the case of Fig. 2. But with DLR running in parallel in several apical dendrites, also complex distributions of datapoints can be handled (Table 2).

2.2.2 Soft-XOR model

This second choice for Ψ is motivated by recent findings from human L2/3 pyramidal cells. According to [Gidon et al., 2020], Ca^{2+}-spikes are initiated in these cells at a threshold activation ϑ of dendrites, but if dendrites are even more strongly activated, these spikes are not elicited. We can model such behavior by a nonlinearity $\Psi(f_{\text{total}})$ that is close to 1 in a certain activation region $(\vartheta, \vartheta + \Delta\vartheta)$, see Fig. 4B (right) and Methods.

A plasticity rule for this model is derived in Methods, where the plasticity at the synapse depends on f_{total}. Again, we can consider a coarse influence of f_{total} on the local plasticity, where it switches between LTP, LTD, and no update based on three thresholds f^- (low f_{total}), f^{mid} (medium f_{total}), and f^+ (high f_{total}), see Fig. 4B (right). This leads to the following updates of synaptic efficacies:

$$
\begin{aligned}
&\text{For } z = 0: \\
&\Delta w_j^B = \begin{cases} -\dot{f}^B(\bar{u}^B)\, x_j^B & , \text{ if } f^- \leq f_{\text{total}} < f^{\text{mid}} \\ \dot{f}^B(\bar{u}^B)\, x_j^B & , \text{ if } f^{\text{mid}} \leq f_{\text{total}} \leq f^+ \\ 0 & , \text{ otherwise} \end{cases} \\
&\text{For } z = 1: \\
&\Delta w_j^B = \begin{cases} \dot{f}^B(\bar{u}^B)\, x_j^B & , \text{ if } f_{\text{total}} < f^- \\ -\dot{f}^B(\bar{u}^B)\, x_j^B & , \text{ if } f_{\text{total}} > f^+ \\ 0 & , \text{ otherwise} \end{cases}
\end{aligned}
\tag{9}
$$

We can interpret these plasticity dynamics as follows. Assume first the case of a Ca^{2+}-spike $z = 1$. For low f_{total} $(< f^-)$, the activation has to increase in order to increase the probability of the model. Hence active synapses are potentiated. For medium f_{total} $(f^- \leq f_{\text{total}} \leq f^+)$, we have already a high probability for a Ca^{2+}-spike. Hence, no update is necessary. For high f_{total} $(> f^+)$, the activation is too high and depression is needed. Consider now the case of no Ca^{2+}-spike $z = 0$. If the neuron is at medium f_{total}, the probability can be decreased either by decreasing the activation (LTD) or increasing it (LTP). In the upper half of the "high probability bump" $(f^{\text{mid}} < f_{\text{total}} \leq f^+)$, LTP is chosen, and in the lower half $(f^- \leq f_{\text{total}} < f^{\text{mid}})$ LTD is chosen, which shift f_{total} in the direction leading to a decreased probability. Outside of this, i.e. for $f_{\text{total}} > f^+$ and $f_{\text{total}} < f^-$, we already predict a low probability of Ca^{2+}-spike and thus no updates are necessary.

2.2.3 Demonstration of the enhanced probabilistic prediction capabilities that emerge through DLR in the case of several apical dendrites

Similar to as shown in Fig. 2 we assume here that data points with labels $z = 0$ or 1 are drawn from some overlapping probability distributions. But in contrast to Fig. 2 we assume here that these distributions are not just Gaussians that can be separated by an optimal linear decision boundary. Examples for the resulting clouds of data points are shown for three examples of more complex distributions in Fig. 4B.

Using a model with two dendritic branches, we consider three different kinds of interactions between these two branches – a saturating interaction with a low threshold, high threshold, and a soft-XOR interaction. One sees in Fig. 4B that each of these three types of interactions is useful for learning probabilistic predictions for three different types of distributions of data points. Furthermore

one sees that training via DLR enables the dendritic branches to also provide meaningful predictions for presynaptic firing rates $\mathbf{x}$ that never occurred during training.

A detailed quantitative evaluation is provided in Methods, see Table 2 there. Here we summarize these results: When using two dendritic branches, the cases of saturating interaction with low and high value of the strong spike threshold perform very well on the OR and AND tasks respectively. The Soft-XOR interaction performs much better than the saturating interaction in modelling the XOR dataset. This validates the intuitive understanding of the different interactions and how they correspond to different Boolean operations. Adding more branches with a saturating interaction with an intermediate threshold value ϑ, we see that these configurations perform very well across all the three datasets thus showing that increasingly complex distributions can be modelled by a single configuration given enough branches.

3 Methods

3.1 Implementation Details of DLR

Each dendritic branch indexed by B receives top-down input $\mathbf{x}^B$ weighted by input weights $\mathbf{w}^B$. An input vector $\mathbf{x}^B = [x_1^B, \ldots, x_N^B]^T$ is presented to the branch as input spike trains $s_1^B(t), \ldots, s_N^B(t)$ for a fixed duration. Mathematically, we model the spike trains $s_n^B(t)$ as sums of Dirac delta pulses at spike times. For each n, the spike times are generated by a Poisson point process of rate equal to x_n^B.

For the duration of the input presentation, the membrane potential u is given by the weighted sum of the spike trains convolved temporally with a linear postsynaptic potential kernel k_{PSP}

$$u^B(t) = \sum_{j=1}^{N} w_j \left(k_{PSP} * s_j^B\right)(t), \tag{10}$$

where we use a double exponential PSP kernel

$$k_{PSP}(t) = \lambda_{PSP} \left(e^{-\frac{t}{\tau_f}} - e^{-\frac{t}{\tau_r}}\right). \tag{11}$$

In our simulations, we used values $\lambda_{PSP} = 1$, $\tau_f = 2$ ms, and $\tau_r = 10$ ms for the scaling- and time-constants.

We define the quantity $\hat{x}_n^B(t) \triangleq (k_{PSP} * s_n)(t)$ as the un-weighted contribution of incoming synapse n to the dendritic potential in branch B. For any PSP shape, the expectation of $\hat{x}_n^B(t)$ is proportional to the spike input rate x_n^B.

Thus, for the weight update rule in 3, we use the time average of the membrane potential $u^B(t)$ ($\bar{u}^B$) to calculate $q_{\mathbf{w}}(\mathbf{x})$, and the time average of $\hat{x}_n^B(t)$ ($\bar{x}_n^B$) in lieu of x_n^B. This is because $\bar{x}^B$ is a quantity that is local to the incoming dendritic synapse and can thus be used in local synaptic weight updates.

For the case of the single branch SLR, the implemented weight update is thus given by

$$\begin{aligned} \Delta w_j &= \eta \left(z - q_{\mathbf{w}}(\mathbf{x})\right) \bar{x}_j^B \,, \\ &= \eta \left(z - \sigma\left(\beta\left(\bar{u} - u_0\right)\right)\right) \bar{x}_j^B \,, \end{aligned}$$

where we make use of the definition of $q_{\mathbf{w}}(\mathbf{x})$ from equation 2.

For the multi-branch update rules, we use $\bar{x}_n^B$ and $\bar{u}_n^B$ in lieu of x_n^B and u_n^B respectively when implementing the weight update rules corresponding to the saturating and soft-XOR dynamics in equations 8, 9.

3.2 Derivation of DLR for a single branch

In general, each dendritic branch indexed by B receives top-down input $\mathbf{x}^B$ weighted by weights $\mathbf{w}^B$. Here, we consider the case of a single dendritic branch. Hence, to simplify notation, we will in the following suppress the superscripts and simply write $\mathbf{x}$ for the inputs to this branch and $\mathbf{w}$ for the corresponding weights. In our model the input $\mathbf{x}$ represents the vector of firing rates of the neurons presynaptic to the dendritic branch.

The Ca^{2+}-spikes are denoted by z where $z = 1$ denotes the presence of the calcium spike, and $z = 0$ denotes the absence of a Ca^{2+}-spike for the duration of the input. We denote the joint distribution of the input $\mathbf{x}$ and the calcium spikes z as $p_{\text{target}}(\mathbf{x}, z)$. Our regression formulation then seeks to provide an estimator parametrized by $\mathbf{w}$ ($q_{\mathbf{w}}(\mathbf{x})$) which is optimized to approximate $p_{\text{target}}(z = 1|\mathbf{x})$. Namely, we aim to train the weights $\mathbf{w}$ so that $q_{\mathbf{w}}(\mathbf{x})$ accurately predicts the probability of the occurrence of a Ca^{2+}-spike given a particular top-down input $\mathbf{x}$.

The cross entropy loss function that measures the distance between $q_{\mathbf{w}}(\mathbf{x})$ and the data distribution p_{target} is defined by

$$\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[z \log q_{\mathbf{w}}(\mathbf{x}) + (1-z)\log\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)\right], \tag{12}$$

where $\mathbf{x}, z$ are sampled from the data distribution p_{target}.

The corresponding gradient with respect to the parameters $\mathbf{w}$ is given by

$$\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\nabla_{\mathbf{w}}\, q_{\mathbf{w}}(\mathbf{x})}{q_{\mathbf{w}}(\mathbf{x})\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)}\left(z - q_{\mathbf{w}}(\mathbf{x})\right)\right]. \tag{13}$$

In our case, we take into account the fact that $q_{\mathbf{w}}(\mathbf{x})$ is a function of the dendritic membrane potential which is proportional to $\mathbf{w}^T\mathbf{x}$. Hence, if $q_{\mathbf{w}}(\mathbf{x})$ takes the simple form $q(\mathbf{w}^T\mathbf{x})$ for some nonlinearity q, one gets

$$\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\dot{q}(\mathbf{w}^T\mathbf{x})}{q(\mathbf{w}^T\mathbf{x})\left(1 - q(\mathbf{w}^T\mathbf{x})\right)}\left(z - q(\mathbf{w}^T\mathbf{x})\right)\mathbf{x}\right]. \tag{14}$$

A further simplification arises if one uses a logistic sigmoid function as nonlinearity q such that:

$$q_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\beta(\mathbf{w}^T\mathbf{x} - u_0)\right) = \sigma\left(\beta(\bar{u} - u_0)\right), \tag{15}$$

where $\sigma(a) = \frac{1}{1+e^{-a}}$ and β and u_0 are model constants. Hence, in this case, the predictive depolarization that results from synaptic currents at the dendrite is a saturated version of $\bar{u}$ as given in equation 2. Then, using the fact that $\sigma'(a) = \sigma(a)\left(1 - \sigma(a)\right)$, one obtains

$$\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\left(z - q_{\mathbf{w}}(\mathbf{x})\right)\mathbf{x}\right]. \tag{16}$$

When we insert the definition of $q_{\mathbf{w}}(\mathbf{x})$ from equation 15 into the loss function $\mathcal{LL}(\mathbf{w})$, we find that this loss function corresponds to a well known linear regression scheme called Logistic Regression. It is known that this loss function is a convex loss function.

Thus, with $\mathbf{x}$ encoded as the mean input spike rate, and $\bar{u} = \mathbf{w}^T\mathbf{x}$ (equation 1), we see that the weight update rule described in equation 3 performs stochastic descent along the gradient of the convex logistic cross entropy loss function. This guarantees the convergence of the weight update to the global optimum of the loss function.

3.3 Derivation of DLR for the case of multiple branches

Here, $\mathbf{x}^B, \mathbf{w}^B$ represent the weights and input to dendritic branch B, and $\mathbf{x}, \mathbf{w}$ represent the full input and weights. The corresponding cross entropy loss with $q_{\mathbf{w}}(\mathbf{x})$ defined as in equation 4 is given by

$$\begin{aligned}\mathcal{LL}(\mathbf{w}) &= -E_{\mathbf{x},z}\left[z \log q_{\mathbf{w}}(\mathbf{x}) + (1-z)\log\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)\right] \\ &= -E_{\mathbf{x},z}\left[z \log \Psi\left(f_{\text{total}}\right) + (1-z)\log\left(1 - \Psi\left(f_{\text{total}}\right)\right)\right].\end{aligned} \tag{17}$$

The derivative of the above loss with respect to the weights $\mathbf{w}^B$ of a branch B evaluates to

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\dot{\Psi}\left(f_{\text{total}}\right)}{\Psi\left(f_{\text{total}}\right)\left(1 - \Psi\left(f_{\text{total}}\right)\right)}\left(z - \Psi\left(f_{\text{total}}\right)\right)\dot{f}^B(\mathbf{w}^{B^T}\mathbf{x}^B)\,\mathbf{x}^B\right]. \tag{18}$$

3.3.1 Derivation of DLR for the saturating model

In this model, a sigmoidal shape for Ψ is chosen:

$$\Psi\left(f_{\text{total}}\right) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right), \tag{19}$$

where α and θ are suitable constants. Using this in equation 18 we obtain the gradient for the saturating model:

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -\,\alpha\,\left(z - \Psi\left(f_{\text{total}}\right)\right)\,\dot{f}^B(\bar{u}^B)\,\mathbf{x}^B, \tag{20}$$

which leads to the weight update rule in equation 7.

In order to minimize the information about $\Psi\left(f_{\text{total}}\right)$ at the branch, one can consider a gating of updates based on thresholds on $\Psi\left(f_{\text{total}}\right)$ which approximates the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ in the update rule 7. We consider the two cases $z = 0$ and $z = 1$ separately. For $z = 0$, the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ is between -1 and 0. It becomes very small when $\Psi\left(f_{\text{total}}\right)$ is close to zero. A gating mechanism which blocks plasticity if $\Psi\left(f_{\text{total}}\right)$ is below some threshold q^- close to 0 approximates this term. For $z = 1$, the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ is between 0 and 1. It becomes very small when $\Psi\left(f_{\text{total}}\right)$ is close to one. A gating mechanism which blocks plasticity if $\Psi\left(f_{\text{total}}\right)$ is above some threshold q^+ close to 1 approximates this term. Hence, we can formulate the approximate update 8.

3.3.2 Derivation of DLR for the soft-XOR model

In the soft-XOR model, Ψ is given by

$$\Psi(f_{\text{total}}) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right)\sigma\left(\alpha\left(-f_{\text{total}} + \vartheta + \Delta\vartheta\right)\right) \tag{21}$$

$$\equiv \sigma_1(f_{\text{total}})\sigma_2(f_{\text{total}}). \tag{22}$$

Suitable parameters α, θ, and $\Delta\theta$ lead to an XOR-like behavior (high probability in a certain activation region), see Fig. 4B (right). Using this nonlinearity in equation 18, we obtain

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\sigma_2(f_{\text{total}}) - \sigma_1(f_{\text{total}})}{\left(1 - \Psi\left(f_{\text{total}}\right)\right)}\left(z - \Psi\left(f_{\text{total}}\right)\right)\dot{f}^B(\mathbf{w}^{B^T}\mathbf{x}^B)\,\mathbf{x}^B\right]. \tag{23}$$

Considering a coarse influence of the f_{total} on the local plasticity, we can formulate an approximate rule which switches between LTP, LTD, and no update based on three thresholds f^- (close to ϑ), f^{mid} $(\vartheta + \Delta\vartheta/2)$, and f^+ (close to $\vartheta + \Delta\vartheta$), see Fig. 4B. This leads to the updates given in equation 9.

3.4 Simulation Details: Classification of 2D data (Fig. 2)

The training examples for this task consist of points generated from two 2D Gaussian clusters with means and covariances $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for $k \in \{0, 1\}$. All points generated from $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ have the target value $z = k$. The prior probabilities for the classes 0 and 1 were chosen to be equal (0.5). Thus to generate a training sample $(\mathbf{x}_i, z_i)$, we first pick the Gaussian component $k \in \{0, 1\}$ with equal probability i.e. $k \sim \text{bernoulli}(p = 0.5)$. We then pick $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and assign the target $z_i = k$. This gives the following conditional probability distribution $p(z = 1|\mathbf{x})$

$$p(z = 1|\mathbf{x}) = \frac{p(\mathbf{x}|k = 1)}{p(\mathbf{x}|k = 0) + p(\mathbf{x}|k = 1)},$$

where $p(\mathbf{x}|k)$ is the Gaussian probability density function with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. It is the aim of the algorithm to learn this conditional distribution in the quantity $q_{\mathbf{w}}(\mathbf{x})$ in the dendrite (equation 2).

The covariance matrix $\boldsymbol{\Sigma}_k$ can be decomposed as $V_k\, {S_k}^2\, V_k^T$. Here V_k is an orthogonal matrix where the columns are the eigenvectors of $\boldsymbol{\Sigma}_k$ and give the directions along which the 2D Gaussian is aligned. S_k is a diagonal matrix with each diagonal entry representing the standard deviation of the Gaussian along the corresponding eigenvector. The parameters used for this simulation are:

$$\boldsymbol{\mu}_0 = [120,\, 120]\,\text{Hz}, \quad V_0 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}, \quad S_0 = \begin{bmatrix} 60 & 0 \\ 0 & 20 \end{bmatrix} \text{Hz},$$
$$\boldsymbol{\mu}_1 = [200,\, 200]\,\text{Hz}, \quad V_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 60 & 0 \\ 0 & 20 \end{bmatrix} \text{Hz}.$$

For this demonstrative simulation we consider the neuron to be receiving the input on a single dendritic branch, with the DLR being applied to the incoming weights to that branch. Each data point $\mathbf{x}_i$, is padded with an additional component of constant rate of 40Hz (in order to model intercept fitting). Each component of this data point is then given as input to the network encoded as a Poisson spike train with mean rate equal to the value of the component. This spike train is provided to the neuron for a duration of 400 ms. Targets are fed to the network by enabling or disabling Ca^{2+}-spike firing corresponding to $z = 1$ and $z = 0$ respectively. Fig. 1B illustrates this neuron and learning setup.

When training the input weights to the neuron using DLR, the initial weights for each run are initialized to positive values drawn from a normal distribution with mean 9.0 and standard deviation 4.5. The weights are clipped at zero so that they stay positive in order to obey Dale's law.

The standard perceptron classifier trains a set of weights $\mathbf{w}$ so that for any data point $(\mathbf{x}, z)$, we should have $\mathbf{w}^T\mathbf{x} > 0$ if $z = 1$ and $\mathbf{w}^T\mathbf{x} < 0$ if $z = 0$. For each iteration of updating the weights $\mathbf{w}$, we randomly sample a single data point $(\mathbf{x}, z)$ from the target distribution p_{target}, and apply the following standard perceptron update to weights $\mathbf{w}$. Note that we also add here the same additional component of 40Hz to $\mathbf{x}$ to enable the perceptron to train the intercept of the separating line.

$$\Delta\mathbf{w} = \begin{cases} \eta\,\mathbf{x} & \text{if } z = 1 \text{ and } \mathbf{w}^T\mathbf{x} < 0 \\ -\eta\,\mathbf{x} & \text{if } z = 0 \text{ and } \mathbf{w}^T\mathbf{x} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

The learning rates η for both the DLR and perceptron learning (used in equations 3, and 24 respectively) decay harmonically from an initial value of η_0 to a final value of η_{final} over the course of the 30000 iterations that we run them for in Fig. 2. For the DLR we use $\eta_0 = 1.8$ (See Table 3), and for the perceptron learning we use $\eta_0 = 0.01$, where the lower learning rate is used to aid any possible convergence of the perceptron. In both cases $\eta_{final} = \eta_0/6$. Thus we see that even with a lower learning rate and an idealized perceptron learning, the perceptron rule does not converge for this case where the data is not linearly separable.

3.5 Simulation Details: Predicting the future sensory input from a moving object (Fig. 3)

3.5.1 Input encoding and parameters

In our simulation, the moving object is a 2D circle which moves within a 30×30 grid and has a diameter of 5 Grid Units (GU). We consider here the task of predicting the sensory input corresponding to its position after a fixed delay of T. The top-down input corresponding to an instance of a moving object is specified by a four-tuple (r_x, r_y, v, θ) consisting of x and y coordinates r_x, r_y of the initial position, the speed v, and the direction angle θ. This four-tuple is encoded in a population coded manner described below. This set of population coded inputs is connected to a 30×30 grid of L5P neurons (one neuron per grid position). The speed and direction are perturbed by a Gaussian noise to model inaccuracies in top-down measurement. This grid is trained to reproduce the image of the ball at its final position which is determined by the initial position and the true (unperturbed) velocity in the external environment. The incoming weights to each L5P neuron are trained using DLR with the target being the corresponding pixel value in this image.

The top-down inputs specifying the initial position and velocity of the ball are encoded as follows. The initial position (r_x, r_y) can take any value such that both r_x, r_y take values in the range of $S_r = [3, 27]$ GU. The speed v is defined in terms of the distance (in GU) that the moving object covers in the fixed delay T between the top-down and the bottom up inputs. This distance can take any value in the range $S_v = [4, 20]$ GU. The direction is specified by an angle θ to the horizontal which can take any value in the range $[0, 2\pi)$.

In addition, before being given as input, the speed v and the direction angle θ are perturbed by the addition of a Gaussian noise in order to model a noisy estimate of the object velocity. This means that given a particular initial position and velocity as input, the true velocity of the ball and consequently the final position of the ball is uncertain.

The encoding of these values into neuron populations is described below. We consider the sets M_r, M_v and M_θ, which consist of N_r, N_v, and N_θ equally spaced values in the ranges S_r, S_v, and S_θ respectively. Then, corresponding to each four-tuple $\big(\mu_x, \mu_y, \mu_v, \mu_\theta\big) \in M_r \times M_r \times M_v \times M_\theta$, we have a single input neuron that fires according to a Gaussian tuning curve centered at $\big(\mu_x, \mu_y, \mu_v, \mu_\theta\big)$. The neuron encodes the input $\big(r_x, r_y, v, \theta\big)$ by firing at a rate ρ given by

$$\rho = \frac{\rho_{\text{ total}}}{Z} e^{-\frac{(r_x-\mu_x)^2 (r_y-\mu_y)^2 (v-\mu_v)^2 (v-\mu_\theta)^2}{8{\sigma_r}^2 {\sigma_r}^2 {\sigma_v}^2 {\sigma_\theta}^2}}, \tag{25}$$

where Z is a normalization factor calculated so that the total spike rate of all the input neurons is $\rho_{\text{ total}} = 250$ Hz. Here the standard deviation of the response curve corresponding to position σ_r, velocity σ_v, and direction σ_θ are constant across all neurons.

For the simulation whose results are shown in Fig. 3, we use $N_r = 10$, $N_v = 10$, $N_\theta = 24$, $\sigma_r = 2.67$, $\sigma_v = 1.78$, $\sigma_\theta = 0.2618 = 15.0°$. This leads to a total input dimension of ${N_r}^2 N_v N_\theta = 24000$.

The standard deviations of the Gaussian noise perturbations applied to the speed and direction components of the top-down input are equal to the width of the respective receptive fields σ_v, σ_θ. In order to train the network we make use of 168000 data points in the training dataset.

All the weights are initialized to a constant positive value of 9.0 and are clipped at zero during training so that they stay non-negative in accordance with Dale's law.

3.5.2 Calculating actual probability of a neuron's receptive field being covered by the object at time T

The imprecision in the estimate of the object velocity, modeled by the perturbed input speed and direction, leads to the final position of the object being uncertain. Each pyramidal neuron in the 30×30 grid receives as input to the apical dendrites the four-tuple $s^{in} \triangleq (r_x, r_y, v^{in}, \theta^{in})$. Here (r_x, r_y) is the initial position of the object and (v^{in}, θ^{in}) are the noisy estimates of speed and direction of the velocity used by the pyramidal neuron. We denote the true velocity of the object by (v, θ), and define the corresponding four-tuple $s \triangleq (r_x, r_y, v, \theta)$.

Consider a single neuron in the grid. For this neuron, we denote as Z the random variable such that $Z = 1$ if and only if the object covers the receptive field of this neuron at time T, and $Z = 0$ otherwise. Given the noise in the estimate of velocity, we thus wish to calculate $p(Z = 1|s^{in})$, which is the conditional distribution that is estimated by the neuron via $q_{\mathbf{w}}(\mathbf{x})$

$$p(Z = 1|s^{in}) = E_{s|s_{in}} \left[p\left(Z = 1 \mid s\right) \right].$$

Here we use the fact that the final position of the object is fully determined given the true velocity and the initial position s.

To calculate $p(s|s_{in})$ we use the following facts regarding (v, θ) and (v^{in}, θ^{in}) (as in description of input above)

$$\begin{aligned} v &\sim \text{Uniform in } S_v, \\ \theta &\sim \text{Uniform in } S_\theta, \\ v^{in} &= v + \epsilon_v, \text{ where } \epsilon_v \sim N(0, \sigma_v), \\ \theta^{in} &= \theta + \epsilon_\theta, \text{ where } \epsilon_\theta \sim N(0, \sigma_\theta). \end{aligned}$$

Thus, using Bayes' theorem, we can calculate

$$\begin{aligned} p(s|s_{in}) &= \frac{p\left(s_{in} \mid s\right) p\left(s\right)}{p\left(s^{in}\right)} \\ &= \frac{p\left(s_{in} \mid s\right) p\left(s\right)}{\iint_{(v,\theta) \in S_v \times S_{theta}} p\left(s^{in} \mid s\right) p\left(s\right)} \\ &= \frac{p\left(s_{in} \mid s\right)}{\iint_{(v,\theta) \in S_v \times S_{theta}} p\left(s^{in} \mid s\right)}. \end{aligned}$$

The above can be computed using the fact that $p(s_{in}|s)$ is the probability density function of a normal distribution.

3.5.3 Measurement of performance

Here we provide measurements to demonstrate that the network reliably predicts the probability that a neuron receives input from the moving object at time T. We term the position of the center of the moving object as predicted by the noisy estimate of the velocity as the *estimated final position*. We then consider three concentric slices of the neuron grid centered at the estimated final position, with diameters $0-5$ GU, $5-9$ GU, and $9-13$ GU. For the pyramidal neurons within a slice, we compare the mean predicted probability $q_{\mathbf{w}}(\mathbf{x})$ and the mean actual probability, of their receptive fields being covered by the object at time T.

1. Slice of diameter $0-5$ GU: mean $q_{\mathbf{w}}(\mathbf{x}) = 0.3546$, mean actual probability $= 0.4108$
2. Slice of diameter $5-9$ GU: mean $q_{\mathbf{w}}(\mathbf{x}) = 0.1666$, mean actual probability $= 0.1668$
3. Slice of diameter $9-13$ GU: mean $q_{\mathbf{w}}(\mathbf{x}) = 0.0461$, mean actual probability $= 0.0435$

From this we can clearly see that the mean predicted probability $q_{\mathbf{w}}(\mathbf{x})$ drops away from the estimated final position, as well as that the theoretically calculated probabilities are followed closely by the predicted probabilities. The examples in Fig. 3 and the results above are calculated from combinations of initial position and velocity that were not seen during training.

3.6 Simulation Details: Predicting complex distributions with multiple dendritic branches (Fig. 4)

In this experiment, the 2D data is generated from multiple gaussian clusters. The parameters and the class labels assigned to each cluster are chosen to create datasets corresponding to an OR, AND, and XOR operation shown in Fig. 4.

		OR dataset	AND dataset	XOR dataset
Multi-branch DLR:				
Number of Branches	Branch Interaction			
2	Saturating (low ϑ)	0.21	0.60	0.61
2	Saturating (high ϑ)	0.65	0.18	0.60
2	Soft-XOR	0.19	0.33	0.36
4	Saturating	0.20	0.17	0.45
6	Saturating	0.18	0.16	0.36
Logistic Regression:		0.43	0.45	0.69

Table 2: Performance of DLR for the case of several interacting apical dendrites for probabilistic predictions in cases where there are no linear optimal decision boundaries: Mean negative log likelihood (NLL) in nats (lower is better) is given for the training sets of the three different data distributions considered in 4B, and for the three types of interactions among apical dendrites that we consider, as well as for different numbers of apical dendrites. One sees that using a saturating interaction with low threshold for two branches excels at the OR task, while the high threshold scheme excels at the AND task. The Soft-XOR interaction turns out to be substantially more effective for learning with two branches the probabilistic prediction for the case of the XOR data distribution. However, with more branches the saturating interaction supports learning of probabilistic predictions via DLR for all three tasks. Each of the mean NLL values above is calculated from the best 8 out of 14 independent training runs (to avoid outliers which do not converge).

In order to solve this task, we consider a neuron with N_B (two or more) dendritic branches each with their respective incoming synapses with weights $\mathbf{w}^1, \ldots, \mathbf{w}^{N_B}$. All of these branches receive the same 2D input, i.e., $\mathbf{x}^B = [\mathbf{x}; 1] = [x_1, x_2, 1]$ for $B = 1, \ldots, N_B$. This input is provided as in the 2D regression task for a single branch, i.e., as a spike sequence with a spike rate proportional to the input value. The additional input component with a value of 20 Hz is given in order to model the training of the branch baseline activation.

The various configurations in table 2 along with corresponding parameters are below.

1. 2 Branches, saturating interaction, low threshold $\vartheta = 0.6$: This implements the OR operation between the two branches.

2. 2 Branches, saturating interaction, high threshold $\vartheta = 1.2$: This implements the AND operation between the two branches.

3. 2 Branches, soft-XOR interaction, $\vartheta = 0.5$, $\Delta\vartheta = 1.0$. This implements the XOR operation between the two branches.

4. 4 Branches, saturating interaction, $\vartheta = 1.4$.

5. 6 Branches, saturating interaction, $\vartheta = 2.1$.

For all the saturating interactions, we use a scale factor $\alpha = 12$. The parameters q^+, q^- used in the learning rule in 8 are set to $q^+ = 0.9$ and $q^- = 0.1$. For the Soft-XOR interaction, we use $\alpha = 8.0$ and set $f^- = \vartheta = 0.5$, $f^+ = \vartheta + \Delta\vartheta = 1.5$ and $f^{\text{mid}} = \vartheta + \Delta\vartheta/2 = 1.0$.

The initial weights for each branch B are initialized in the following manner. We pick a point $\mathbf{s}^B \triangleq \left[s_1^B, s_2^B\right]$, where both s_1^B, s_2^B are randomly picked uniformly from the interval $[110, 130]$ Hz. We then pick a random direction vector $\mathbf{h}^B$ sampled uniformly from a unit circle. Using this pair of $\mathbf{s}^B, \mathbf{h}^B$, we set the initial weights $\mathbf{w}_0^B$ so that the separating plane is a plane that passes through the point $\mathbf{s}^B$, with the direction of the plane determined by the normal vector $\mathbf{h}^B$. The weights in this case are unconstrained in sign to allow the model to fit the complex distributions in the case of the low dimensional inputs.

Parameter	Single-branch DLR		Multi-branch DLR	
	2-D regression	Predicting input from moving object	Saturating interaction	Soft-XOR interaction
η_0	1.8	40.0	4.0	0.3
η_{final}	0.3	6.67	4.0	0.3
β	0.5	0.5	0.1	1.0
u_0	20.0	15.0	7.0	7.0
λ_{PSP}	1.0	1.0	1.0	1.0
τ_r	2.0 ms	2.0 ms	2.0 ms	2.0 ms
τ_f	10.0 ms	10.0 ms	10.0 ms	10.0 ms

Table 3: Parameters related to DLR that are used for each experiment. η_0, η_{final} are the initial and final learning rate used. β and u_0 are the scale and threshold for the sigmoid function used to calculate $q_{\mathbf{w}}(\mathbf{x})$ and f^B in equations 2 and 5 respectively.

3.7 Common Parameters for DLR:

Some of the common parameters and the values for the DLR rule for the three experiments are given in Table 3.

4 Discussion

We have presented a mathematical framework and learning theory for the arguably most important synaptic connection for the integration of top-down predictions from higher brain areas into computational processing of sensory input in lower areas: Synapses from axons of neurons in higher brain areas onto apical dendrites of pyramidal cells in lower areas. Experimental data show that plasticity at these synapses follows rules that differ from rules for correlated Hebbian learning, because they are gated by instructive signals [Magee and Grienberger, 2020]. Furthermore these instructive signals differ from commonly considered global third factors for synaptic plasticity, such as dopamine, that typically signal reward. Instead, these instructive signals either emerge locally within the neuron — such as Ca^{2+}-spikes [Kampa et al., 2006, Kampa et al., 2007], NMDAR spikes [Gambino et al., 2014], or other types of plateau potentials [Bittner et al., 2015, Bittner et al., 2017, Magee and Grienberger, 2020] – or arrive in the dendritic tuft through synaptic connections from brain areas such as the perirhinal cortex [Doron et al., 2020] or higher order thalamus [Aru et al., 2020]. We have focused in our illustrative example in Fig. 1 and in the text on the case where these instructive signals, denoted by z in our mathematical models, take the form of Ca^{2+}-spikes. But our theoretical framework and normative learning theory also applies to other forms of instructive signals. The only aspect on which this learning theory is based is that these instructive signals last for 50-100 ms or longer, so that the firing rates of presynaptic neurons become salient, rather than single spikes.

Apart from these neurophysiological differences, the functional impact of plasticity at these synapses on apical dendrites has been shown to differ from that of other synaptic connections. For example, the input from apical dendrites has been shown to shift the perceptual threshold in a graded manner [Takahashi et al., 2016]. In addition, if these synapses learn to predict sensory inputs on the basis of inputs from higher brain areas, such predictions are inherently uncertain, and therefore would be functionally most useful if they aim at learning probabilities, rather than categorical decisions. We have shown that this learning goal can be achieved through simple rules for LTP and LTD, gated by instructive signals z, at synapses on apical dendrites. In fact, a suitable balancing of LTP and LTD can produce a depolarization of the dendritic tuft that approximates logistic regression, i.e., a simple form of probabilistic prediction that is in a well-defined sense optimal. Hence we refer to this normative model for synaptic plasticity in the dendritic tuft as dendritic logistic regression (DLR). We have shown in illustrative examples that DLR enables the dendritic tuft to make meaningful probabilistic predictions in the case where the same prediction is sometimes correct and sometimes not (Fig. 2A). Furthermore these probabilistic predictions can be learnt really fast through DLR (Fig. 2B). In particular, they provide substantially better performance than the perceptron learning rule [Moldwin and Segev, 2020]. The reason is that the latter is guaranteed to work well only if the training examples are linearly separable, in particular if they do not contain contradicting examples that commonly occur in the case of probabilistic predictions. We have also demonstrated in Fig. 3 that the DLR rule enables distal dendrites to predict future sensory inputs caused by a moving object.

A concrete experimentally testable prediction of this normative model is that, after learning, the depolarization in a dendritic branch of the tuft is correlated with the probability that the current

synaptic input to the dendritic tuft coincides with an instructive signal, for example with a Ca^{2+}-spike. Another experimentally testable prediction of our model is that the current depolarization of the dendritic branch scales the amplitude of LTP and LTD during the induction of plasticity. In particular, if synaptic input to an apical dendrite coincides with a plateau potential, the expression of LTP is predicted to be weaker if the dendrite is already strongly depolarized. Furthermore, our normative model predicts that synaptic input to an apical dendrite causes LTD in the absence of a plateau potential, and the impact of LTD is stronger if the apical dendrite is already stronger depolarized. Our model also raises the question whether LTP of synapses on distal dendrites of pyramidal cells in other brain areas besides the neocortex, such as area CA1 [Bittner et al., 2015, Bittner et al., 2017], is also accompanied by a corresponding forms of LTD, and whether LTP and LTD combine to generate probabilistic predictions of other events.

We have extended our learning theory in the second part to the case of several interacting branches in the dendritic tuft. Two different types of interactions have been found in experiments: Summation of depolarization [Jadi et al., 2014, Poirazi et al., 2003], and competition between depolarized branches [Gidon et al., 2020]. We have characterized in Fig. 4 additional capabilities of probabilistic prediction that are enabled by DLR for these two types of interactions between dendritic branches. In both cases, probabilistic predictions emerge that can learn predictions for substantially more complex distributions than for the simplified case of a single dendritic branch. In this sense, our learning theory also throws new light on likely computational contributions of the dendritic arbor.

Acknowledgments

This work was supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreements No. 785907 (Human Brain Project) and No. 899265 (ADOPD).

References

[Aru et al., 2020] Aru, J., Suzuki, M., and Larkum, M. E. (2020). Cellular mechanisms of conscious processing. *Trends in Cognitive Sciences*.

[Bittner et al., 2015] Bittner, K. C., Grienberger, C., Vaidya, S. P., Milstein, A. D., Macklin, J. J., Suh, J., Tonegawa, S., and Magee, J. C. (2015). Conjunctive input processing drives feature selectivity in hippocampal CA1 neurons. *Nature neuroscience*, 18(8):1133–1142.

[Bittner et al., 2017] Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S., and Magee, J. C. (2017). Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science*, 357(6355):1033–1036.

[Doron et al., 2020] Doron, G., Shin, J. N., Takahashi, N., Drüke, M., Bocklisch, C., Skenderi, S., de Mont, L., Toumazou, M., Ledderose, J., Brecht, M., et al. (2020). Perirhinal input to neocortical layer 1 controls learning. *Science*, 370(6523).

[Gambino et al., 2014] Gambino, F., Pagès, S., Kehayas, V., Baptista, D., Tatti, R., Carleton, A., and Holtmaat, A. (2014). Sensory-evoked LTP driven by dendritic plateau potentials in vivo. *Nature*, 515(7525):116–119.

[Gidon et al., 2020] Gidon, A., Zolnik, T. A., Fidzinski, P., Bolduan, F., Papoutsi, A., Poirazi, P., Holtkamp, M., Vida, I., and Larkum, M. E. (2020). Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 367(6473):83–87.

[Jadi et al., 2014] Jadi, M. P., Behabadi, B. F., Poleg-Polsky, A., Schiller, J., and Mel, B. W. (2014). An augmented two-layer model captures nonlinear analog spatial integration effects in pyramidal neuron dendrites. *Proceedings of the IEEE*, 102(5):782–798.

[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.

[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.

[Leinweber et al., 2017] Leinweber, M., Ward, D. R., Sobczak, J. M., Attinger, A., and Keller, G. B. (2017). A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron*, 95(6):1420–1432.

[Magee and Grienberger, 2020] Magee, J. C. and Grienberger, C. (2020). Synaptic plasticity forms and functions. *Annual review of neuroscience*, 43:95–117.

[Moldwin and Segev, 2020] Moldwin, T. and Segev, I. (2020). Perceptron learning and classification in a modeled cortical pyramidal cell. *Frontiers in Computational Neuroscience*, 14.

[Naud et al., 2014] Naud, R., Bathellier, B., and Gerstner, W. (2014). Spike-timing prediction in cortical neurons with active dendrites. *Frontiers in computational neuroscience*, 8:90.

[Poirazi et al., 2003] Poirazi, P., Brannon, T., and Mel, B. W. (2003). Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999.

[Takahashi et al., 2020] Takahashi, N., Ebner, C., Sigl-Glöckner, J., Moberg, S., Nierwetberg, S., and Larkum, M. E. (2020). Active dendritic currents gate descending cortical outputs in perception. *Nature Neuroscience*, 23(10):1277–1285.

[Takahashi et al., 2016] Takahashi, N., Oertner, T. G., Hegemann, P., and Larkum, M. E. (2016). Active cortical dendrites modulate perception. *Science*, 354(6319):1587–1590.

[Tzilivaki et al., 2019] Tzilivaki, A., Kastellakis, G., and Poirazi, P. (2019). Challenging the point neuron dogma: FS basket cells as 2-stage nonlinear integrators. *Nature communications*, 10(1):1–14.

[Xu et al., 2012] Xu, N.-l., Harnett, M. T., Williams, S. R., Huber, D., O'Connor, D. H., Svoboda, K., and Magee, J. C. (2012). Nonlinear dendritic integration of sensory and motor input during an active sensing task. *Nature*, 492(7428):247–251.