Self-supervised learning of probabilistic prediction through synaptic plasticity in apical dendrites: A normative model

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

June 25, 2022

Abstract

Sensory information is processed by the brain not in a simple feedforward fashion. Rather, bottom-up inputs are combined in pyramidal cells of sensory cortices with top-down information from higher brain areas that arrives through synapses in apical dendrites. The exact functional role of these top-down inputs has remained unknown. A promising abstract model posits that they provide probabilistic priors for bottom-up sensory inputs. We show that this hypothesis is consistent with a large number of experimental about synaptic plasticity in apical dendrites, in particular with the prominent role of NMDA-spikes. We identify conditions under which this synaptic plasticity could approximate the gold standard for self-supervised learning of probabilistic priors: logistic regression. Furthermore, this perspective suggests an additional functional role for the complex structure of the dendritic arborization plays: It enables the neuron to learn substantially more complex landscapes of probabilistic priors.

1 Introduction

A key open problem in neuroscience and cognitive science is understanding how bottom-up sensory information and internally generated top-down information from higher brain areas are integrated in the computations of laminar cortical microcircuits [Vezoli et al., 2021, Larkum, 2013]. Several abstract theoretical frameworks have been proposed for understanding the organization of these computations, for example that top-down information from higher areas produces probabilistic (Bayesian) priors for bottom-up information [Lee and Mumford, 2003], or internal reconstructions (predictions) of sensory inputs [Rao and Ballard, 1999]. But it remains difficult to relate these abstract theories to experimentally observed processes in cortical microcircuits. Experimental data clarify where top-down and bottom-up information meet in generic cortical microcircuits: in pyramidal cells on layers 2/3 and 5. More precisely, top-down inputs reach these cells primarily through synapses on their apical dendrites in superficial layers, especially in layer 1 [Larkum, 2022], whereas bottom-up information primarily targets basal dendrites. Hence we need to understand the plasticity processes which shape the impact and information content of this top-down synaptic input. It has been proposed by [Shin et al., 2021] that two sequential phases of learning should be distinguished: During the first

phase, which takes substantial time and primarly involves synaptic plasticity in basal dendrites, these pyramidal cells are grounded in the most prevalent statistical features of the environment. During the second phase the activity of these neurons is linked to internal models, using the putative predictive role of apical dendrites. Experimental data show in fact that plasticity rules for synapses in apical dendrites differ from the more commonly considered Hebbian and STDP-like rules for basal dendrites. In particular, local NMDA spikes in the apical tuft of pyramidal cells play an essential role for this plasticity, see e.g. [Augusto and Gambino, 2019, Stuyt et al., 2021, Larkum, 2022] for recent reviews. Related to the important role that NMDA spikes, which typically last 100ms and longer, play for synaptic plasticity in apical dendrites, unlike STDP for basal dendrites the precise temporal relation between these NMDA spikes and other salient events, appears to be less relevant for inducing synaptic plasticity. But also the functional goal of these plasticity processes in apical dendrites remains unclear. Hence we are addressing here the question to what extent these processes, and in particular dendritic spikes, could support self-supervised learning of a prediction of bottom-up inputs to the neuron.

Self-supervised learning of probabilistic predictions requires rules for synaptic plasticity that differ from the more commonly considered rules for classification learning: Predictions are inherently uncertain, i.e., they turn out to be sometimes correct and sometimes not. Most rules for synaptic plasticity that are suitable for classification learning, such as the perceptron learning rule [Moldwin and Segev, 2020], do not perform well if the labels of training examples are inconsistent -which is the generic scenario for learning probabilistic predictions of uncertain binary events. Furthermore, for learning probabilistic predictions, a plasticity rule has to converge to a probability, rather than to a guessed classification. A gold standard for such learning is logistic regression [Bishop, 2006], since this method learns a probabilistic prediction that is, under some constraint on the mathematical form of the prediction, theoretically optimal. We examine to what extent rules for synaptic plasticity in apical dendrites that are consistent with experimental data can approximate logistic regression. More precisely, we derive an arguably optimal approximation to logistic regression through synaptic plasticity in apical dendrites that we term Dendritic Logistic Regression (DLR). Numerous experimental data on synaptic plasticity in apical dendrites are consistent with DLR. In particular, dendritic spikes (NMDA spikes) play a key role for DLR.

Our theoretical framework holds for any number of top-down inputs, however in two of our demonstrations (in Fig. 2, 4), we consider just the case of 2-dimensional top-down inputs $\mathbf{x}$, because learnt landscapes of probabilistic priors in the space of top-down inputs can easily be visualized in this case. We first present the theoretical framework for the case of a single dendritic branch in the apical tuft of a pyramidal cell. The extension to the case of several dendritic branches is presented subsequently. Since experimental data point to two types of interactions between dendritic branches in the apical tuft of different types of neurons, a saturating summation [Naud et al., 2014, Poirazi et al., 2003, Gooch et al., 2022], as well as a soft XOR for some types of pyramidal cells on layer 2/3 of the human cortex [Gidon et al., 2020], as predicted by the model of [Gorski et al., 2017], we analyze functional consequences of DLR for both cases. It turns out that both types of neuron models support DLR, but facilitate learning of different types of probability landscapes. Since either type of probability landscape is useful in some scenarios and less in others, the resulting learning theory for apical dendrites supports the co-existence of different subtypes of pyramidal cells in layer 2/3 [Planert et al., 2021] with different interactions between apical dendrites, since this guarantees that good probabilistic predictions can be learnt in each scenario by at least some of the neurons.

For simplicity, we will refer to the dendritic branches in the apical tuft of pyramidal cells in layer 2/3 or 5 just as apical dendrites.

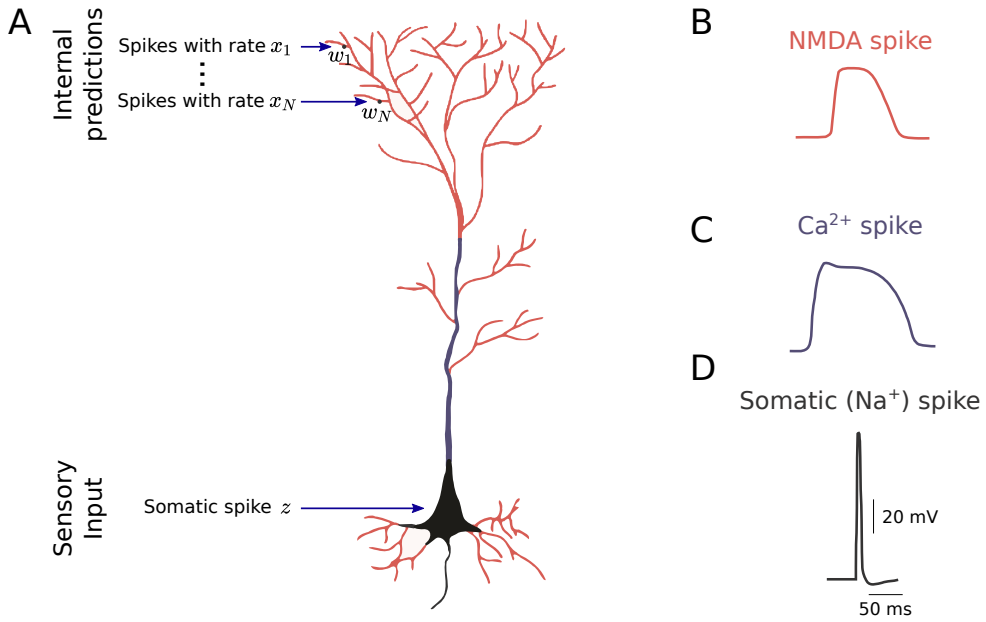

Figure 1: Scheme of a pyramidal cell with apical dendrites, dendritic spikes, and somatic spikes. A) Generic model of a thick tufted pyramidal cell in layer 2/3 or layer 5 of a generic laminar cortical microcircuit. It integrates two different inputs streams, a top-down stream that arrives in its apical dendrites (shown at the top), and a bottom-up stream that arrives in basal dendrites and . influences the generation of somatic spikes. see [Larkum, 2013, Augusto and Gambino, 2019, Larkum et al., 2022]. Synaptic inputs from neurons in higher brain areas with firing rates $x_1, \ldots, x_N$ are integrated with synaptic weights $w_1, \ldots, w_N$ in the apical tuft. We propose a normative learning model for synaptic plasticity of these weights that aims at learning the probability of a co-occurring somatic spike. **B)** NMDA spikes in apical dendrites play a major rule in this normative model. **C)** In view of the larger distance between apical dendrites and the soma for a oyramidal cell on layer 5, information about a somatic spikes reaches apical dendrites through an intermediate signal, a Ca^{2+}-spikes, which is generated in the dendritic trunk (blue region in panel A). **D)** A somatic spike. This figure is adapted from [Augusto and Gambino, 2019].

2 Results

2.1 A normative model for self-supervised learning of probabilistic predictions through synaptic plasticity in an apical dendrite

The main principle of the common hierarchical Bayesian inference model [Lee and Mumford, 2003] for integrating top-down information from higher brain areas into processing of bottom-up input streams in the cortex is the following: The top-down information $\mathbf{x}$ is used on each level of the cortical hierarchy to produce a prior $p(z|\mathbf{x})$ for the bottom-up input to that level. Anatomical data show that these two information streams meet in pyramidal cells on layers 2/3 and 5 of the generic cortical microcircuit. More precisely, the top-down information arrives primarily on apical dendrites and bottom-up information primarily on basal dendrites of pyramidal cells. For an individual pyramidal cell, the Bayesian inference model of [Lee and Mumford, 2003] amounts to the challenge of estimating the probability that input from basal dendrites causes a somatic spike, given the firing rates $\mathbf{x}$ of neurons in higher cortical areas that form synapses on its apical dendrites. We denote the presence or absence of a somatic spike (within some suitable time-window) by a binary variable z. Obviously, the synaptic weights $\mathbf{w} = [w_1, w_2, \ldots, w_N]$ in apical dendrites are essential for transforming firing

NMDA spike y	Somatic spike z	weight update $\Delta\mathbf{w}$
0	0	0
0	1	0
1	0	$-\ x_j$
1	1	$\frac{x_j}{q_{\mathbf{w}}(\mathbf{x})} - x_j$

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 represents a somatic spike (or a global plateau potential in some types of neurons).

activity in higher brain areas into a prediction of somatic spikes of an individual neuron. Logistic regression [Bishop, 2006], see Methods, provides a gold standard for this probabilistic prediction, since it is theoretically optimal among all predictions that apply the sigmoid activatation function in order to map the weighted sum $\mathbf{w} \cdot \mathbf{x} = \sum_{j=1}^{N} w_j x_j$ into the range of probabilities, i.e., to values between 0 and 1.. We examine to what extent this gold standard can be approximated by synaptic plasticity rules for apical dendrites.

Experimental data show that local NMDA spikes are essential for synaptic plasticity in apical dendrites [Gambino et al., 2014]. Full knowledge of the underlying molecular processes is not yet available. But it has been shown in [Stein et al., 2021] that non-ionotropic NMDA receptor signaling plays a key role. Furthermore, it was shown there that the presence or absence of coincident Ca^{2+}-influx decides between LTP and LTD. We use the binary variable y to denote the presence or absence of an NMDA spike in the apical dendrite. We denote the probability of an NMDA spike in an apical dendrite, in dependence on the synaptic input $\mathbf{x}$ that it receives, by $q_{\mathbf{w}}(\mathbf{x})$. We examine whether this term can be tuned through biologically plausible synaptic plasticity rules to provide an estimate of the probability $p_{\text{target}}(z = 1|\mathbf{x})$ of a somatic spike of the neuron. We use for the term $q_{\mathbf{w}}(\mathbf{x})$ the simple formula

$$q_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\beta\left(\mathbf{w} \cdot \mathbf{x} - u_0\right)\right), \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the standard sigmoid function, and β is a constant scaling factor. The threshold u_0 can be interpreted in biophysical terms as the minimal membrane voltage that is required to remove the Mg^{2+} block of NMDA receptors. The term $q_{\mathbf{w}}(\mathbf{x})$ can then be related to the dendritic excitability of an apical dendrite, more precisely to the probability of generating an NMDA spike in response to top-down input $\mathbf{x}$. According to the experimental data reviewed in [Stuyt et al., 2021, Gonzalez et al., 2022] this dendritic excitability changes during learning, and is an important factor in the induction of LTP. Furthermore, the resulting modified dendritic excitability plays an important computational role: it tunes the perceptual threshold [Takahashi et al., 2016, Takahashi et al., 2020].

In order to derive a plasticity rule for apical dendrites from the goal of predicting $p_{\text{target}}(z|\mathbf{x})$ through the probability of an NMDA spike in the dendrite, we use the common cross entropy loss as a measure for the distance between the ground truth probability and the estimate $q_{\mathbf{w}}(\mathbf{x})$. This loss function is convex, and hence can be optimized through stochastic gradient descent. This stochastic gradient descent yields the following rule for synaptic plasticity for the weights $\mathbf{w}$ of synapses onto the apical dendrite:

$$\Delta w_j \ = \ \eta \left(\frac{z}{q_{\mathbf{w}}(\mathbf{x})} - 1 \right) y x_j \ , \tag{2}$$

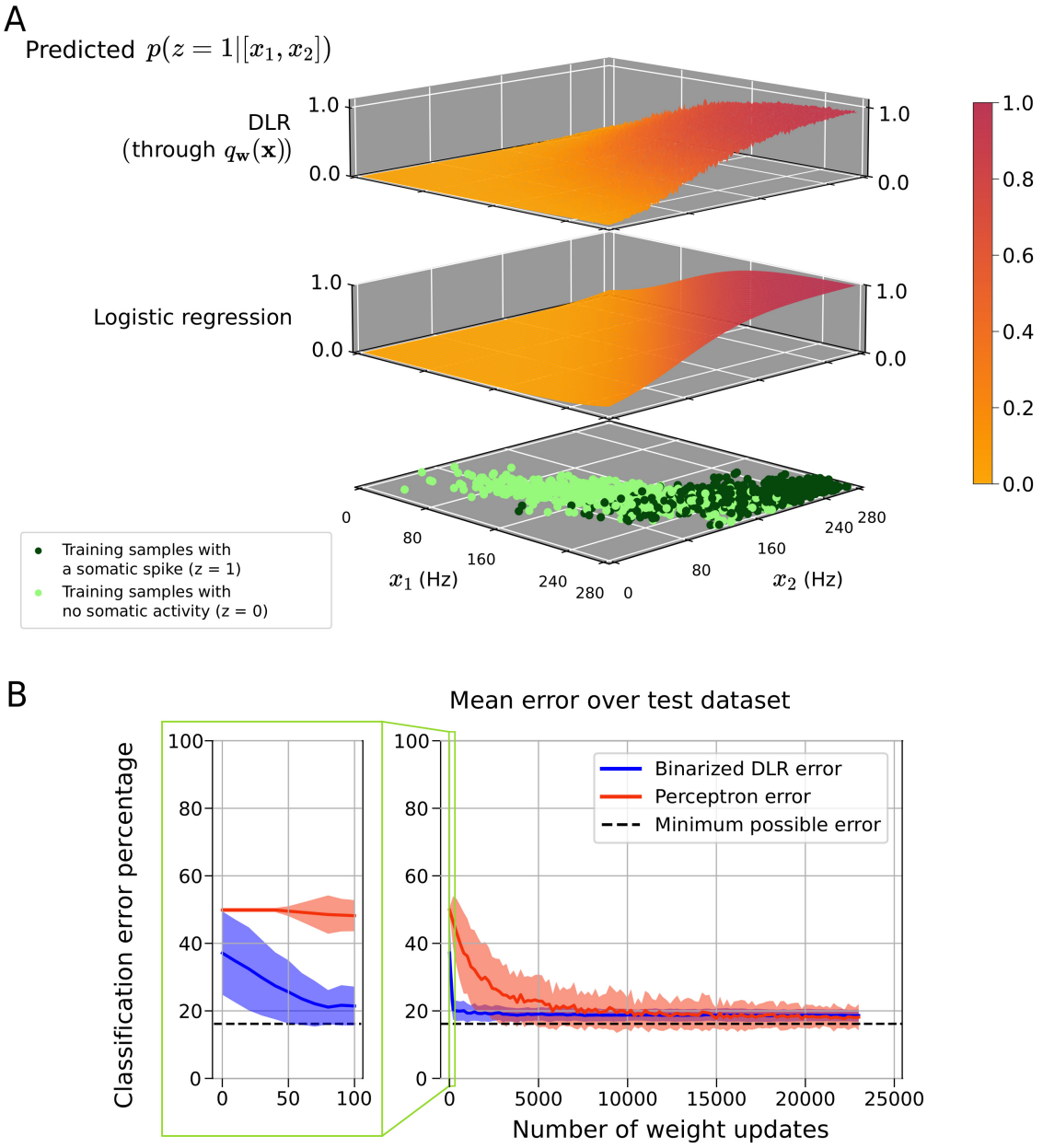

Figure 2: Illustration of the challenge to learn probabilistic predictions, and performance of the gold standard logistic regression as well as a biologically plausible approximation (Dendritic Logistic Regression: DLR) on a simple benchmark task. A) Bottom row: Top-down inputs $\mathbf{x}$ are generated by two overlapping Gaussians. The $\mathbf{x}$ from one of the Gaussians are accompanied by somatic spikes (dark green points denote some samples), the other Gaussian generates $\mathbf{x}$ that are not accompanied by somatic spikes (like green sample points). Middle row: Probability that a top-down input $\mathbf{x}$ is accompanied by a somatic spike, estimated through logistic regression. Top row: DLR produces almost equally good estimates for the same task (mean difference 0.045). **B)** The perceptron learning rule, as considered in [Moldwin and Segev, 2020], can only learn binary classifications, no probabilities. Hence the performance of the preceptron learning rule (red curve) and DLR (blue curve) can only be compared if one ignores the additional information that DLR learning provides through in the form of a probability, and rounds its learnt probabilities to 0 or 1 (Binarized DLR). An interesting difference between these two approaches for learning classifications is that DLR converges in much fewer weight updates to provide close-to-optimal classifications. Mean and variance of the error are shown for 100 learning episodes with independently drawn initial weights and data points.

where Δw_j is the change of the weight of synapse j and $\eta > 0$ is a constant scaling factor. The rule can equivalently be described in the form of Table 1. We show in Section 3.2, that this weight update performs stochastic gradient descent for the previously described cross entropy loss. Since this stochastic gradient descent is guaranteed to converge to the minimal value of this loss function. Furthermore, since this minimal value is the same as the value that can be achieved through theoretically optimal logistic regression for approximating the target value $p_{\text{target}}(z|\mathbf{x})$, it gives rise to a normative model. We refer to this normative plasticity rule as Dendritic Logistic Regression (DLR).

DLR is consistent with the experimental data of [Gambino et al., 2014, Stein et al., 2021] insofar as synaptic plasticity only occurs in the presence of an NMDA spike, i.e., $y = 1$. Furthermore, consistent with the results of [Stein et al., 2021], the decision between LTP and LTD depends on the presence of a signal that causes Ca^{2+}-influx in temporal proximity, which can be caused in L2/3 pyramids by a backpropagating action potential, and in L5 pyramids by a Ca^{2+}-spike that is triggered when significant depolarization in the apical tuft coincides with a somatic spike [Larkum, 2013, Larkum et al., 2022]. In addition, DLR predicts that the amplitude of LTP is scaled by the term $q_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\beta\left(\mathbf{w}\cdot\mathbf{x} - u_0\right)\right)$, i.e., by the current excitability of the dendrite for NMDA spikes, reducing the amplitude of LTP if this excitability is already fairly high.

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

We consider two clusters of firing patterns $\mathbf{x}$ of neurons in higher brain areas that project to an apical dendrite of a pyramidal cell. One of them models activity patterns $\mathbf{x}$ that were accompanied by a somatic spike (dark green points at the bottom of in the bottom row of panel A in Fig. 2), and the other models activity patterns $\mathbf{x}$ that were not accompanied by a somatic spike (light green points in the bottom row of panel A in Fig. 2). Importantly, in order to model the inherent uncertainty of predictions we consider the case that these two clusters of firing patterns $\mathbf{x}$ of neurons in higher brain areas overlap, i.e., the same $\mathbf{x}$ is sometimes accompanied by a somatic spike and sometimes not. More precisely, each cluster is generated by a Gaussian $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, with parameters $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, and these two Gaussians overlap. In order to facilitate visualization, we consider here just two-dimensional firing patterns $\mathbf{x}$, but the same principle applies for inputs of arbitrary dimension from higher brain areas.

Fig. 2A shows that the apical dendrites learn through DLR to predict via $q_{\mathbf{w}}(\mathbf{x})$ quite precisely the probability that an input pattern $\mathbf{x}$ is accompanied by a somatic spike. In fact, one sees that the resulting prediction performance was very similar to that of theoretically optimal logistic regression. The probabilistic prediction capability that results from 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 achieved an average of 0.384 nats, compared to 0.358 nats achieved through logistic regression. We show in Fig. S1 of the Supplement that DLR also learns close to optimal prediction probabilities if these approach nowhere the extreme values of 0 or 1.

A direct comparison of convergence properties of the perceptron learning rule and DLR requires us to binarize the probabilistic prediction that is learnt through DLR, since the perceptron learning rule can only learn to produce binary classifications. However, the two point clouds are not linearly separable due to the overlap of the Gaussians which generate them, and hence the perceptron learning rule (as investigated in [Moldwin and Segev, 2020] struggles with separating them, due to the inconsistencies of classifications in the training set, as indicated by the red trace in Fig. 2B. In contrast, the blue trace in Fig. 2B shows that binarized DLR converges quite fast to an optimal linear separation of the two point clouds. This large performance difference between perceptron learning and DLR might seem surprising. The perceptron weight update rule (see equation 21) has

dependencies only based on the binary condition of whether $\mathbf{w} \cdot \mathbf{x}$ is greater or lesser than u_0. In comparison, the more fine-grained dependence of the DLR on dendritic excitability $q_{\mathbf{w}}(\mathbf{x})$ (equation 1) greatly improves the convergence properties.

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 the probability that this object will be at a given later time point T at a particular position. This is a non-trivial prediction task since estimates of initial position, movement speed and direction are inherently noisy, see the cartoon in Fig. 3B. We present a simple model which shows that these probabilistic predictions can be acquired through DLR. We assume that the receptive fields of a 2D array of neurons cover the movement field, and for simplicity, we represent the moving object 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 of current speed and movement direction.

The underlying self-supervised learning task is specified as follows. Each neuron in this 2D array receives the estimates of position and velocity at time 0, encoded as spike rates $\mathbf{x}$, that arrive at its apical dendrites from higher brain areas. Furthermore, each neuron fires a somatic spike z when its receptive field is covered by the moving object at time T. Hence we can interpret the target distribution $p_{\text{target}}(z = 1|\mathbf{x})$ as the probability that the object will be positioned over the receptive field of a neuron at time T, given the position and velocity at time 0.

According to the underlying theory, once the weights have been learned via DLR, $q_{\mathbf{w}}(\mathbf{x})$ approximates for each neuron in the 2D grid $p_{\text{target}}(z = 1|\mathbf{x})$. The learnt probabilities 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 these learnt predictions are close to optimal for this task. 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 Extension to the case of synaptic plasticity in multiple apical dendrites

When there are several dendritic branches, the question arises as to how the contributions from different branches are combined for the prediction $q_{\mathbf{w}}(\mathbf{x})$, that is in our model the probability of a dendritic spike arising from apical dendrites. Here, we denote the weights and presynaptic firing rates for synapses in each dendritic branch B by $\mathbf{w}^B = \left[w_1^B, w_2^B, \ldots, w_N^B\right]$ and $\mathbf{x}^B = \left[x_1^B, x_2^B, \ldots, x_N^B\right]$ respectively, and the weights and presynaptic firing rates across all branches as $\mathbf{w}, \mathbf{x}$ respectively. As in the single dendrite case, the goal of synaptic plasticity in apical dendrites is in our model that $q_{\mathbf{w}}(\mathbf{x})$ learns to become a good estimate of the target probability $p_{\text{target}}(z = 1|\mathbf{x})$.

In order to arrive at a model for the probability $q_{\mathbf{w}}(\mathbf{x})$ of a dendritic spike arising from the apical dendrite, we assume that contributions f^B from each branch B are summed and followed by a nonlinearity Ψ:

$$q_{\mathbf{w}}(\mathbf{x}) = \Psi\Big(\sum_B f^B\Big) \triangleq \Psi\left(f_{\text{total}}\right). \tag{3}$$

Consistent with the single-branch model (equation 1), the contribution of each branch is given by a

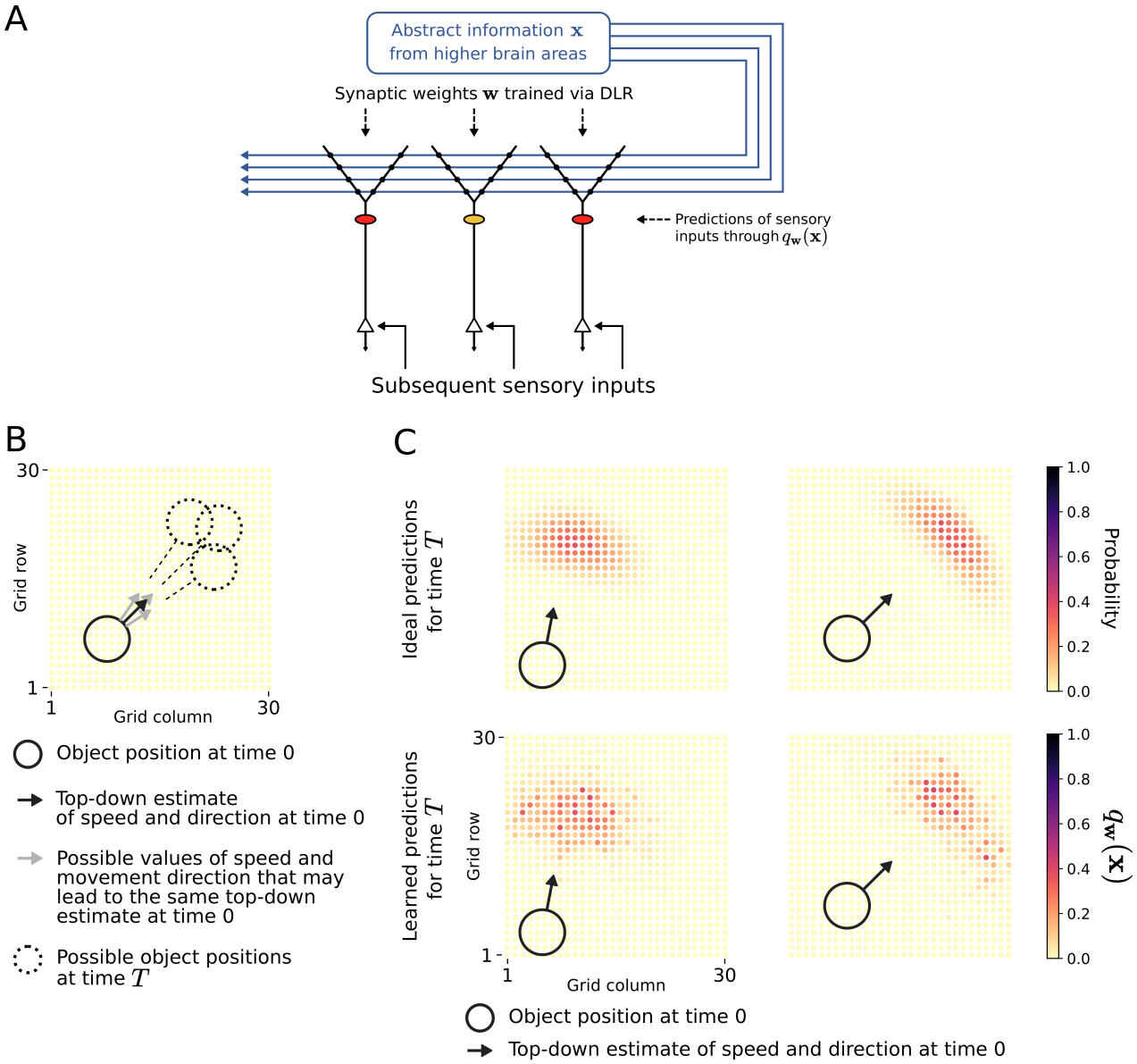

Figure 3: Performance of DLR for self-supervised learning to predict whether the receptive field of a neuron will be covered by a moving object. A) Schema of the simple network architecture, a 30×30 neuron sheet of neurons with apical dendrites that each have their own receptive field, but receive the same top-down input $\mathbf{x}$. $\mathbf{x}$ denotes in this case estimates of the initial position, speed and movement direction of some object. When the object covers at a later time T, the prediction span, the receptive field of a neuron it causes a somatic spike. Each neuron learns through synaptic plasticity in its apical dendrites as estimate of the probability $q_{\mathbf{w}}(\mathbf{x})$ that this will happen (color code for individual predictions as in panel C. **B)** Because of uncertainty in the estimates of the initial position, speed, and movement direction of the object, this top-down input can only give rise to probabilistic estimates of the position of the object at time T. **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. For comparison, in the upper row, we show for each grid position, the actual probability that it will be covered at time T by the moving object, computed using the knowledge of the full distribution of noisy estimates for time 0 (see Section 3.6.3, Methods).

sigmoidal function of its weighted input rate, i.e.

$$f^B = \sigma(\beta\,(\sum_i x_i^B w_i^B - u_0)). \tag{4}$$

For the nonlinearity Ψ we consider two possibilities. According to the classical view [Poirazi et al., 2003, Tzilivaki et al., 2019, Jadi et al., 2014], the dendritic activity is given by a saturating nonlinearity. 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

$$\Psi\left(f_{\text{total}}\right) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right), \tag{5}$$

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 value of ϑ less than 1 implements a soft OR operation (a minimum of one branch needs to be active to get a high probability output). A value of 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 somatic spikes. 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:

$$\Delta\, w_j^B = \eta\,\left(z - \Psi\left(f_{\text{total}}\right)\right)\,\dot{f}^B\, x_j^B, \tag{6}$$

where η is a learning rate and $\dot{f}^B$ is the derivative of f^B at the current value of $\mathbf{w}^B \cdot \mathbf{x}^B$. We mention two differences to the single-dendrite update: (1) the term $\dot{f}^B$ means that there is a nonlinear dependence on the local weighted input, 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 6. This leads to the following updates of synaptic efficacies:

$$\begin{aligned}
&\text{For } z = 0: \\
&\qquad \Delta\, w_j^B = \begin{cases} -\dot{f}^B x_j^B & , \quad \text{if } \Psi\left(f_{\text{total}}\right) \geq q^- \\ 0 & , \quad \text{otherwise} \end{cases} \\
&\text{For } z = 1: \\
&\qquad \Delta\, w_j^B = \begin{cases} \dot{f}^B x_j^B & , \quad \text{if } \Psi\left(f_{\text{total}}\right) \leq q^+ \\ 0 & , \quad \text{otherwise} \end{cases}
\end{aligned} \tag{7}$$

for plasticity thresholds q^- and q^+. To see the correspondence with rule 6, first consider the case without somatic spike ($z = 0$). Here, the term $\left(z - \Psi\left(f_{\text{total}}\right)\right)$ in equation 6 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 somatic 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 7.

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:

$$
\begin{aligned}
&\text{For } z=0: \\
&\qquad \Delta\, w_j^B = \begin{cases} -\dot{f}^B\, x_j^B & , \quad \text{if } f^{-} \leq f_{\text{total}} < f^{\text{mid}} \\ \dot{f}^B\, x_j^B & , \quad \text{if } f^{\text{mid}} \leq f_{\text{total}} \leq f^{+} \\ 0 & , \quad \text{otherwise} \end{cases} \\
&\text{For } z=1: \\
&\qquad \Delta\, w_j^B = \begin{cases} \dot{f}^B\, x_j^B & , \quad \text{if } f_{\text{total}} < f^{-} \\ -\dot{f}^B\, x_j^B & , \quad \text{if } f_{\text{total}} > f^{+} \\ 0 & , \quad \text{otherwise} \end{cases}
\end{aligned}
\tag{8}
$$

We can interpret these plasticity dynamics as follows. Consider first the case of a somatic 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 dendritic 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 somatic 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 range, i.e. for $f_{\text{total}} > f^{+}$ and $f_{\text{total}} < f^{-}$, the model already predicts a low probability for a somatic 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 $z=0$ and $z=1$ are generated by overlapping probability distributions. But in contrast to Fig. 2, we assume here that each of these distributions is a mixture of Gaussians, rather than single Gaussians that can be

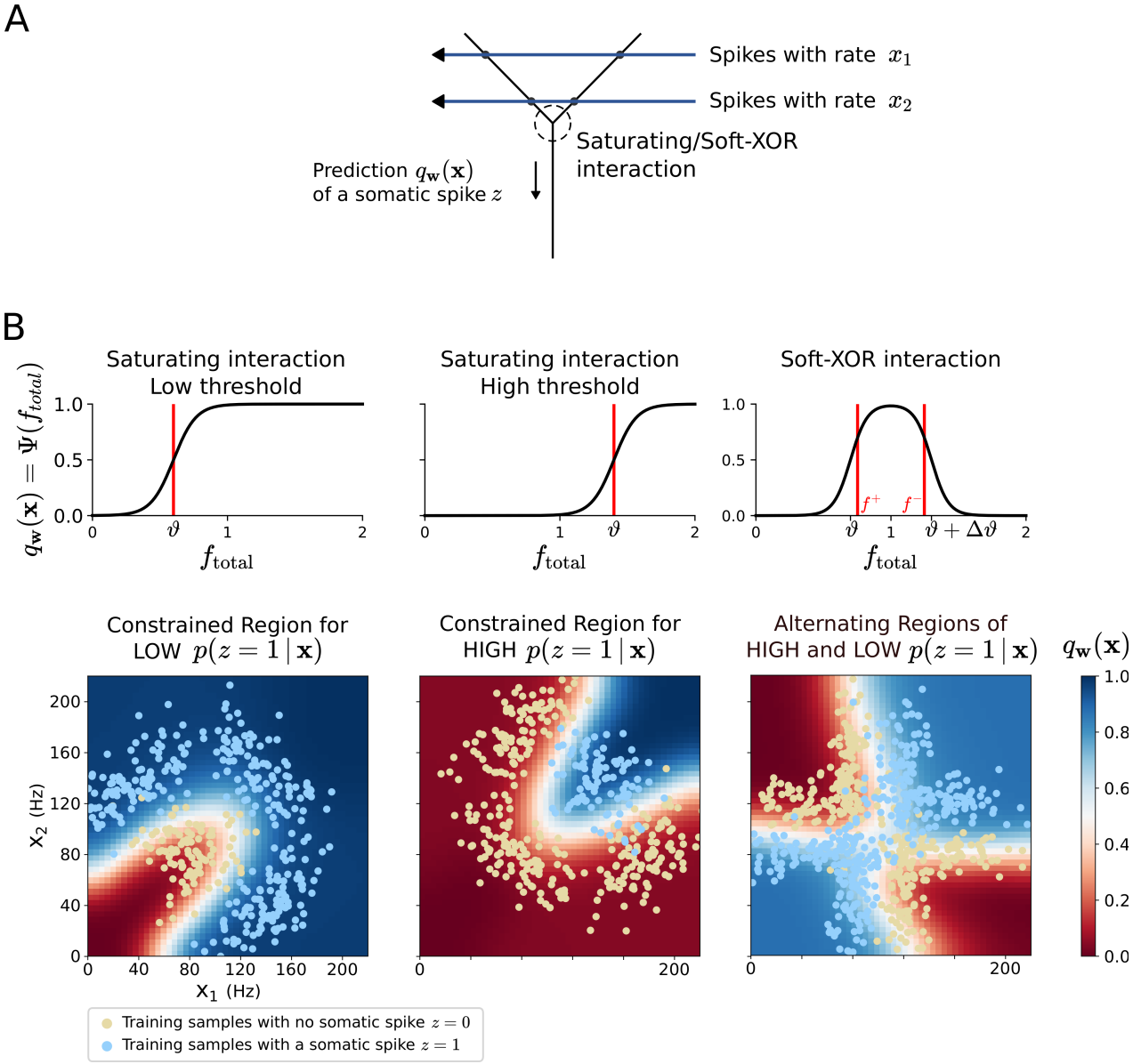

Figure 4: Illustration of complex probability landscapes that can be learnt in the apical dendrites of a neuron if it has more than a single branch. A) We consider for simplicity the case of just two branches in the dendritic tuft, but the theory also covers the case of any number of branches. We consider two different experimentally reported types of interaction between these branches: 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. These were generated through mixture of Gaussians (4–8 depending on the dataset, details in Supplement). A single apical dendrite, that produces probability landscapes of the type shown in Fig. 2, cannot produce good probabilistic predictions for them. Saturating (see first two columns) and XOR-like interactions (see 3rd column) between different dendritic branches enable DLR to produce different types of probability landscapes for predicting the probability of a somatic spike. See (Table 2) for quantitative data on approximation performance.

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 ($\vartheta = 0.6$; Fig. 4B top left), a saturating interaction with a high threshold ($\vartheta = 1.2$; Fig. 4B top middle), and a soft-XOR interaction (Fig. 4B top right). 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: Each task has regions where $p(z = 1|\mathbf{x})$ is close to one (HIGH-p regions) and regions where it is close to zero (LOW-p regions). We tested different configurations on the following three tasks: constrained LOW-p region, constrained HIGH-p region, and alternating HIGH and LOW-p regions (see Fig. 4). When using two dendritic branches, a saturating interaction with low or high threshold performs well on tasks with a constrained LOW-p or HIGH-p region respectively. On the other hand, for the task with alternating LOW and HIGH-p regions, the Soft-XOR interaction performs much better than the saturating interaction. This demonstrates the variety of distributions that can be modeled by different configurations.

We next tested whether the addition of more dendritic branches could improve the capabilities of the apical branches to predict complex distributions. To this end, we performed simulations with four and six branches using saturating interactions with intermediate threshold values ($\vartheta = 1.4$ and $\vartheta = 2.1$ for 4 and 6 branches respectively). We found that these configurations perform very well across all the three datasets, see Table 2. This shows that increasingly complex distributions can be modelled by a single configuration given enough branches.

3 Methods

3.1 Logistic regression

We are referring to Section 4.3.2 of [Bishop, 2006] for an introduction to logistic regression. A more informal account is given on wikipedia. It is explained there that in statistics, the (binary) logistic model is a statistical model that models the probability of one event (out of two alternatives) taking place by having the log-odds (the logarithm of the odds) for the event be a linear combination of one or more independent variables ("predictors"). In regression analysis, logistic regression is estimating the parameters of a logistic model (the coefficients in the linear combination). Formally, in binary logistic regression there is a single binary dependent z, while the independent variables can each be a binary variable or a continuous variable (any real value). The parameters of a logistic regression are most commonly estimated by maximum-likelihood estimation. This does not have a closed-form expression, unlike linear least squares.

Thus logistic regression provides a gold standard for predicting the events where a binary variable z assumes value 1, but it requires off-line computation for a batch of data. However, we show in the following subsection that the online plasticity rule DLR converges through stochastic gradient descent to the same weight vector $\mathbf{w}$.

3.2 Derivation of DLR for a single apical dendrite

We denote the joint distribution of the top-down input $\mathbf{x}$ and the somatic spikes z as $p_{\text{target}}(\mathbf{x}, z)$. Our regression formulation then seeks to provide an estimator $q_{\mathbf{w}}(\mathbf{x})$, parametrized by $\mathbf{w}$, where we aim to train the weights $\mathbf{w}$ so that $q_{\mathbf{w}}(\mathbf{x})$ approximates $p_{\text{target}}(z = 1|\mathbf{x})$, that is, the probability of a somatic spike z 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 $p_{\text{target}}(z|\mathbf{x})$ is defined by

$$\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[z \log q_{\mathbf{w}}(\mathbf{x}) + (1-z)\log\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)\right], \tag{9}$$

where $\mathbf{x}$, z are sampled from $p_{\text{target}}(z, \mathbf{x})$.

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

$$\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\nabla_{\mathbf{w}}\, q_{\mathbf{w}}(\mathbf{x})}{q_{\mathbf{w}}(\mathbf{x})\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)}\left(z - q_{\mathbf{w}}(\mathbf{x})\right)\right]. \tag{10}$$

If $q_{\mathbf{w}}(\mathbf{x})$ has the form $q(\mathbf{w}\cdot\mathbf{x})$ for some nonlinearity q, one gets

$$\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\dot{q}(\mathbf{w}\cdot\mathbf{x})}{q(\mathbf{w}\cdot\mathbf{x})\left(1 - q(\mathbf{w}\cdot\mathbf{x})\right)}\left(z - q(\mathbf{w}\cdot\mathbf{x})\right)\mathbf{x}\right]. \tag{11}$$

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

$$q_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\beta(\mathbf{w}\cdot\mathbf{x} - u_0)\right), \tag{12}$$

where $\sigma(a) = \frac{1}{1+e^{-a}}$ and β and u_0 are model constants. The probability of an NMDA spike $q_{\mathbf{w}}(\mathbf{x})$ is thus a saturated function of the input to the dendrite. Then, using the fact that $\sigma'(a) = \sigma(a)\left(1 - \sigma(a)\right)$, one obtains

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

Now, $q_{\mathbf{w}}(\mathbf{x})$ is the probability of a dendritic spike y. Thus we have $q_{\mathbf{w}}(\mathbf{x}) = p(y = 1|\mathbf{x}, \mathbf{w}) = E_{y|\mathbf{x},\mathbf{w}}[y]$. Hence we can rewrite the above rule as

$$\begin{aligned}
\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w}) &= -E_{\mathbf{x},z}\left[\left(z - q_{\mathbf{w}}(\mathbf{x})\right)\mathbf{x}\right] \\
&= -E_{\mathbf{x},z}\left[\left(\frac{z}{q_{\mathbf{w}}(\mathbf{x})} - 1\right) q_{\mathbf{w}}(\mathbf{x})\,\mathbf{x}\right] \\
&= -E_{\mathbf{x},z}\left[E_{y|\mathbf{x}}\left[\left(\frac{z}{q_{\mathbf{w}}(\mathbf{x})} - 1\right) y\,\mathbf{x}\right]\right] \\
&= \boxed{-E_{\mathbf{x},z,y}\left[\left(\frac{z}{q_{\mathbf{w}}(\mathbf{x})} - 1\right) y\,\mathbf{x}\right]}.
\end{aligned}$$

Setting the weight update as $\Delta w \propto -\nabla_{\mathbf{w}}\, \mathcal{LL}(\mathbf{w})$ completes the derivation of the weight update for DLR described in equation 2.

This derivation proves that DLR performs stochastic descent along the gradient of the convex logistic cross entropy loss function. The convexity of this loss function guarantees stochastic convergence of the weights through DLR 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 top-down input and weights. The corresponding cross entropy loss with $q_{\mathbf{w}}(\mathbf{x})$ defined as in equation 3 is given by

$$\begin{aligned}\mathcal{LL}(\mathbf{w}) &= -E_{\mathbf{x},z}\left[z \log q_{\mathbf{w}}(\mathbf{x}) + (1-z)\log\left(1 - q_{\mathbf{w}}(\mathbf{x})\right)\right] \\ &= -E_{\mathbf{x},z}\left[z \log \Psi\left(f_{\text{total}}\right) + (1-z)\log\left(1 - \Psi\left(f_{\text{total}}\right)\right)\right]. \end{aligned} \tag{14}$$

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

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\dot{\Psi}\left(f_{\text{total}}\right)}{\Psi\left(f_{\text{total}}\right)\left(1-\Psi\left(f_{\text{total}}\right)\right)}\left(z - \Psi\left(f_{\text{total}}\right)\right)\dot{f}^B(\mathbf{w}^B\cdot\mathbf{x}^B)\,\mathbf{x}^B\right]. \tag{15}$$

3.3.1 Derivation of DLR for the saturating model

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

$$\Psi\left(f_{\text{total}}\right) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right), \tag{16}$$

where α and θ are suitable constants. Using this in equation 15 we obtain the gradient for the saturating model:

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\alpha\,\left(z - \Psi\left(f_{\text{total}}\right)\right)\,\dot{f}^B(\mathbf{w}^B\cdot\mathbf{x}^B)\,\mathbf{x}^B\right], \tag{17}$$

which leads to the weight update rule in equation 6.

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 6. 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 7.

3.3.2 Derivation of DLR for the soft-XOR model

In the soft-XOR model, Ψ is given by

$$\Psi(f_{\text{total}}) = \sigma\left(\alpha\left(f_{\text{total}} - \vartheta\right)\right)\sigma\left(\alpha\left(-f_{\text{total}} + \vartheta + \Delta\vartheta\right)\right) \tag{18}$$

$$\equiv \sigma_1(f_{\text{total}})\sigma_2(f_{\text{total}}). \tag{19}$$

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 15, we obtain

$$\nabla_{\mathbf{w}^B}\mathcal{LL}(\mathbf{w}) = -E_{\mathbf{x},z}\left[\frac{\sigma_2(f_{\text{total}}) - \sigma_1(f_{\text{total}})}{\left(1-\Psi\left(f_{\text{total}}\right)\right)}\left(z - \Psi\left(f_{\text{total}}\right)\right)\dot{f}^B(\mathbf{w}^B\cdot\mathbf{x}^B)\,\mathbf{x}^B\right]. \tag{20}$$

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 8.

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

3.5 Implementation Details of DLR

For the case of a single dendritic branch, it receives top-down input spike rates $\mathbf{x}$ weighted by input weights $\mathbf{w}$. An input vector $\mathbf{x} = [x_1, \ldots, x_N]^T$ is presented to the branch as input spike trains $s_1(t), \ldots, s_N(t)$ for a fixed duration T_{sim}. Mathematically, we model the spike trains $s_n(t)$ 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. We then define the quantity

$$\hat{x}_n(t) \triangleq \left(k_{input} * s_n\right)(t),$$

as the un-weighted input contribution through synapse n. where the input spikes are smoothed by the following double exponential input kernel :

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

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

It is then seen that the time average of $\hat{x}_n(t)$ ($\bar{x}_n$), taken over the interval of presentation T_{sim}, is proportional to the input spike rate $\mathbf{x}$. Thus, for the weight update rule in 2, we use the above time average $\bar{\mathbf{x}}$ as a proxy for $\mathbf{x}$ to calculate $q_{\mathbf{w}}(\mathbf{x})$. This is because $\bar{x}$ 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

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

where we make use of the definition of $q_{\mathbf{w}}(\mathbf{x})$ from equation 1.

Similarly, in the case of multi-branch update rules, for each branch B, we calculate using a spike trace $\bar{\mathbf{x}}^B$ as an estimate proportional to $\mathbf{x}^B$ and use $\bar{\mathbf{x}}^B$ when implementing the weight update rules corresponding to the saturating and soft-XOR dynamics in equations 7, 8.

3.5.1 The target distribution $p_{\text{target}}(\mathbf{x}, z)$

We first describe the distribution of the training data. 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 $p_{\text{target}}(z = 1|\mathbf{x})$

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

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 1).

The covariance matrix $\mathbf{\Sigma}_k$ 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 $\mathbf{\Sigma}_k$ and give the directions along which the 2D Gaussian is aligned. V_k^T denotes the transpose of the matrix. 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:

$$\boldsymbol{\mu}_0 = [160,\, 120]\,\text{Hz}, \quad V_0 = \frac{1}{\sqrt{5}}\begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}, \quad S_0 = \begin{bmatrix} 60 & 0 \\ 0 & 20 \end{bmatrix}\text{Hz},$$
$$\boldsymbol{\mu}_1 = [240,\, 200]\,\text{Hz}, \quad V_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 60 & 0 \\ 0 & 20 \end{bmatrix}\text{Hz}.$$

3.5.2 Input encoding and training

The neuron received 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 $T_{\text{sim}} = 400ms$. Targets are fed to the network by inducing or preventing somatic spikes 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 $\mathbf{w} \cdot \mathbf{x} > 0$ if $z = 1$ and $\mathbf{w} \cdot \mathbf{x} < 0$ 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. The perceptron update rule is provided below

$$\Delta\mathbf{w} = \begin{cases} \eta\,\mathbf{x} & \text{if } z = 1 \text{ and } \mathbf{w}\cdot\mathbf{x} < 0 \\ -\eta\,\mathbf{x} & \text{if } z = 0 \text{ and } \mathbf{w}\cdot\mathbf{x} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

The learning rates η for both the DLR and perceptron learning (used in equations 2, and 21 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.

Note that incoming weights are initialized to random positive values and the sign of the weight is fixed in accordance to Dale's law

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

3.6.1 Input distribution

We consider here the following task. Given estimates of the position and velocity of a moving object at time 0, predict the sensory input expected from the object at its new position after a fixed delay of T.

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). More specifically, we have a 30×30 grid of pyramidal neurons, each of which correspond to one position in the aforementioned grid on which the object moves. The soma of each neuron spikes (i.e. $z = 1$) iff the object covers the corresponding position on the grid.

The information that forms the dendritic input is given by the 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. 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 this setup each data point is generated as follows. We sample a four-tuple $(r_x, r_y, v_{real}, \theta_{real})$ picking each parameter uniformly from the above ranges. This four-tuple represents the initial position and real velocity of the object at the initial time point $t = 0$. In order to model the noise introduced by the higher layers, we only feed a noisy perturbation of the real velocity which is calculated as follows

$$v = v_{real} + \epsilon_v, \text{ where } \epsilon_v \sim N(0, \sigma_v),$$
$$\theta = \theta_{real} + \epsilon_\theta, \text{ where } \epsilon_\theta \sim N(0, \sigma_\theta).$$

The input spike rates $\mathbf{x}$ is the population-coded represenation of (r_x, r_y, v, θ), where the population coding is described below. For each pyramidal neuron, the presence of a somatic spike z is determined by the image of the object on the grid at time $t = T$, which is calculated using $(r_x, r_y, v_{real}, \theta_{real})$. A pyramidal neuron will have a somatic spike ($z = 1$) iff in this image, the object covers the corresponding grid position. With this, the incoming weights to each pyramidal neuron are trained using DLR with the target being the corresponding pixel value in this image.

Given the perturbed top-down estimate of initial position and velocity at time 0 as input, the true initial velocity of the object, and consequently the final position of the object at time T is uncertain. The calculation of the corresponding target distribution $p_{\text{target}}(z = 1|\mathbf{x})$ is described below in Section 3.6.3.

3.6.2 Input encoding and parameters

The top-down inputs specifying the initial position and velocity of the object are encoded into neuron populations as follows. 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

$$\rho_{\mu_x,\mu_y,\mu_v,\mu_\theta}\left(r_x, r_y, v, \theta\right) = \frac{\rho_{\text{ total}}}{Z} e^{-\frac{(r_x-\mu_x)^2(r_y-\mu_y)^2(v-\mu_v)^2(v-\mu_\theta)^2}{8{\sigma_r}^2{\sigma_r}^2{\sigma_v}^2{\sigma_\theta}^2}}, \tag{22}$$

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. $\mathbf{x}$ is the set of all such input 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 $\mathbf{x}$ having a total 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.

3.6.3 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 $s \triangleq (r_x, r_y, v, \theta)$. Here (r_x, r_y) is the initial position of the object and (v, θ) 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 $(v_{real}, \theta_{real})$, and define the corresponding four-tuple $s_{real} \triangleq (r_x, r_y, v_{real}, \theta_{real})$.

Consider a single neuron in the grid. For this neuron, the soma spikes ($z = 1$) 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 $p_{\text{target}}(z = 1|\mathbf{x})$, which is the conditional distribution that is estimated by the neuron via $q_{\mathbf{w}}(\mathbf{x})$

$$p_{\text{target}}(z = 1|\mathbf{x}) = p_{\text{target}}(z = 1|s) = E_{s_{real}|s}\left[p\left(z = 1 \mid s_{real}\right)\right].$$

Here we use the fact that the final position of the object is fully determined given the true velocity and the initial position s_{real}.

To calculate $p(s_{real}|s)$ we then use Bayes' theorem:

$$\begin{aligned} p(s_{real}|s) &= \frac{p\left(s \mid s_{real}\right) p\left(s_{real}\right)}{p\left(s\right)} \\ &= \frac{p\left(s \mid s_{real}\right) p\left(s_{real}\right)}{\iint_{(\lambda_v,\lambda_\theta)\in S_v \times S_{theta}} p\left(s \mid s_{real} = (r_x, r_y, \lambda_v, \lambda_\theta)\right) p\left(s_{real} = (r_x, r_y, \lambda_v, \lambda_\theta)\right) d\lambda_v \, d\lambda_\theta}. \end{aligned}$$

Since s_{real} is sampled uniformly, $p(s_{real})$ is a constant, hence it simplifies to

$$p(s_{real}|s) = \frac{p\left(s \mid s_{real}\right)}{\iint_{(\lambda_v,\lambda_\theta)\in S_v \times S_{theta}} p\left(s \mid s_{real} = (r_x, r_y, \lambda_v, \lambda_\theta)\right) d\lambda_v \, d\lambda_\theta}.$$

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

3.6.4 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 $q_{\mathbf{w}}(\mathbf{x}) = 0.1806$, mean actual probability $= 0.1668$
3. Slice of diameter $9-13$ GU: mean $q_{\mathbf{w}}(\mathbf{x}) = 0.0638$, 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.

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

3.7.1 Target distribution $p_{\mathbf{target}}(\mathbf{x}, z)$

In this experiment, the 2D data is generated from multiple gaussian clusters. Each gaussian cluster k has a mean $\boldsymbol{\mu}_k$, covariance $\boldsymbol{\Sigma}_k$, and a corresponding value of the somatic spike $\zeta_k \in \{0, 1\}$. To generate a data point, we first pick a cluster $k \in 1, \ldots, K$ at random, sample $\mathbf{x}$ from it, and assign the somatic spike $z = \zeta_k$ corresponding to the cluster. The resultant joint target distribution is

$$p_{\text{target}}(\mathbf{x}, z) = \frac{1}{K} \sum_{\substack{k \in 1,\ldots,K \\ :: \zeta_k = z}} p\left(\mathbf{x} | \, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right),$$

where $p\left(\mathbf{x} | \, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ is the gaussian probability density function with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$.

The parameters and the somatic spiking corresponding to the different clusters are chosen so as to create three distributions which have respectively a constrained LOW-p region, constrained HIGH-p region, and alternating HIGH and LOW-p regions (see Fig. 4). Here LOW and HIGH-p refer to regions of LOW and HIGH $p_{\text{target}}(z = 1|\mathbf{x})$ respectively. The precise parameters of the component gaussian distributions are provided in the Supplement.

3.7.2 Input encoding and training parameters

In order to model the above distributions, 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 poisson spike sequence with spike rates given by $\mathbf{x}^B$. 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.

		Constrained LOW-p region	Constrained HIGH-p region	Alternating regions of HIGH and LOW-p
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 Fig. 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 task with a constrained LOW-p region, while the high threshold scheme excels at the task with a constrained HIGH-p region. The Soft-XOR interaction turns out to be substantially more effective for learning with two branches the probabilistic prediction for the case with alternating HIGH and LOW-p regions. 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).

1. 2 Branches, saturating interaction, low threshold $\vartheta = 0.6$: This implements a soft-OR operation between the two branches.
2. 2 Branches, saturating interaction, high threshold $\vartheta = 1.2$: This implements a soft-AND operation between the two branches.
3. 2 Branches, soft-XOR interaction, $\vartheta = 0.5$, $\Delta\vartheta = 1.0$. This implements a soft-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 7 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
T_{sim}	400.0 ms	200.0 ms	400.0 ms	400.0 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 1 and 4 respectively.

3.8 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 provided a link between the probabilistic theory of [Lee and Mumford, 2003] for computations in hierarchical cortical networks and synaptic plasticity in apical dendrties of pyramidal cells on layers 2/3 and 5 of generic cortical microcircuits. It has recently been proposed [Shin et al., 2021] that these pyramidal cells become selective for specific bottom-up and top-down input features in two separate learning phases: During the first learning phase these neurons become selective to specific features of sensory inputs, or more generally to specific features of the bottom-up input stream which reaches them primarily through synapses on basal dendrites. Synaptic plasticity on apical dendrites enables these neurons during the second learning phase to integrate their bottom-up processing into concrete contexts and goals. We have presented here a learning theory for this second learning phase. More precisely, we have shown that the theory of [Lee and Mumford, 2003], which postulates a concrete computational goal as outcome of this second learning phase -top-down generation of probabilistic priors for bottom-up information streams, is in many aspects consistent with known experimental data on synaptic plasticity in apical dendrites. More precisely, we have presented a normative model for these plasticity processes based on the hypothesis that their functional goal is self-supervised learning of probabilistic priors from the statistics of coincidences and non-coincidences of particular configurations of firing patterns in the top-down and bottom-up input streams that converge on a neuron. This normative model builds on a key result from statistical learning theory [Bishop, 2006]: That logistic regression defines the optimal weight vector for producing such probabilistic prediction through a weighted sum of top-down inputs $\mathbf{x}$, provided a sigmoid function is used to map the weighted sum into probabilities, i.e., values between 0 and 1. We have identified a concrete online synaptic plasticity rule, called Dendritic Logistic Regression (DLR), that

approximates this gold standard. This plasticity rule is consistent with a number of known details of plasticity rules in apical dendrites, in particular with the prominent role of NMDA spikes for this plasticity [Gambino et al., 2014, Stein et al., 2021]. However many details of synaptic plasticity in apical dendrites remain unknown, especially with regard to LTD.

Understanding learning of such priors, i.e., probabilities, requires to step away from the more familiar set-up of learning in a deterministic setting, where the predicted "labels" are provided by a generally consistent supervisor, rather than through a stochastic process. Although one is tempted to think also about synaptic plasticity in apical dendrites in terms of well-known learning rules such as the perceptron learning rule, a closer look (see also Fig. 2 and S1) shows that such learning rule is not suitable for learning the probability of a stochastically occuring event (such as a somatic spike). The key point is that such somatic spike will sometimes occur for a given top-down input $\mathbf{x}$ and sometimes not. We have demonstrated in Fig. 2B that DLR is well-suited for coping with this challenge: It can even learn for data points that are generated by two overlapping Gaussians the probability that a data point was generated by a specific one of them. In fact DLR can learn this probability as well as logistic regression. In Fig. 3 we have scaled up self-supervised learning through DLR to a 2D sheet of pyramidal cells, and have shown that it enables them to learn a close to optimal 2D probability landscape for estimating the future position of a moving object.

A long-standing open question is whether dendrites offer to neurons a substantial functional advantage, in particular, a conceptual advantage that goes beyond integrating computational capabilities into single neurons for which otherwise a larger neural network would be needed [Larkum, 2022]. Our results provide an interesting new answer to this question: Dendritic arborization enables neurons to learn internal models (probability landscapes) that are substantially more complex, and therefore can provide substantially better fits to target distributions that are generated by their environment, see Fig. 4. We are not aware of a biologically plausible method that would enable even substantially larger networks of point neurons to achieve that, because it requires an integration of several non-binary signals that are readily available within a single neuron, but which are apparently not transmitted between neurons. For the sake of better visualization we have demonstrated in Fig. 4 this qualitative jump in the complexity of probability landscapes that can be learnt by neurons with several apical dendrites only for the case where the top-down input $\mathbf{x}$ is 2-dimensional. But the underlying theory that we have presented holds for any dimension of $\mathbf{x}$. Obviously, a further qualitative jump in the complexity of internal probabilistic models that can be learnt arises when one applies DLR to neurons that receive 100- or 1000 dimensional top-down inputs $\mathbf{x}$.

There exists a staggering diversity of subtypes of pyramidal cells, especially in the human cortex [Berg et al., 2021]. In addition there exists substantial diversity among subtypes of NMDA receptors [Stein et al., 2021] and of NMDA spikes [Stuyt et al., 2021, Larkum et al., 2022]. Hence it would be quite optimistic to expect that all resulting plasticity processes in apical dendrites of pyramidal cells can be described by a simple formula, such as DLR. However, a comparison of postulates of DLR with experimental data on synaptic plasticity processes in a concrete apical dendrite will provide definite insight into the question whether this plasticity process is likely to have the functional goal to learn the probability of a somatic spike. Since DLR is approximately optimal for learning this probability, significant deviations from DLR suggest that this learning process has a different functional goal.

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), the project

No. 899265 (ADOPD), and by a grant from Intel.

References

[Augusto and Gambino, 2019] Augusto, E. and Gambino, F. (2019). Can nmda spikes dictate computations of local networks and behavior? *Frontiers in Molecular Neuroscience*, page 238.

[Berg et al., 2021] Berg, J., Sorensen, S. A., Ting, J. T., Miller, J. A., Chartrand, T., Buchin, A., Bakken, T. E., Budzillo, A., Dee, N., Ding, S.-L., et al. (2021). Human neocortical expansion involves glutamatergic neuron diversification. *Nature*, 598(7879):151–158.

[Bishop, 2006] Bishop, C. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[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.

[Gonzalez et al., 2022] Gonzalez, K. C., Losonczy, A., and Negrean, A. (2022). Dendritic excitability and synaptic plasticity in vitro and in vivo. *Neuroscience*, pages S0306–4522.

[Gooch et al., 2022] Gooch, H. M., Bluett, T., Perumal, M. B., Vo, H. D., Fletcher, L. N., Papacostas, J., Jeffree, R. L., Wood, M., Colditz, M. J., McMillen, J., et al. (2022). High-fidelity dendritic sodium spike generation in human layer 2/3 neocortical pyramidal neurons. *bioRxiv*.

[Gorski et al., 2017] Gorski, T., Veltz, R., Galtier, M., Fragnaud, H., Teleńczuk, B., and Destexhe, A. (2017). Inverse correlation processing by neurons with active dendrites. *bioRxiv*, page 137984.

[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.

[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, 2022] Larkum, M. E. (2022). Are dendrites conceptually useful? *Neuroscience*.

[Larkum et al., 2022] Larkum, M. E., Wu, J., Duverdin, S. A., and Gidon, A. (2022). The guide to dendritic spikes of the mammalian cortex in vitro and in vivo. *Neuroscience*.

[Lee and Mumford, 2003] Lee, T. S. and Mumford, D. (2003). Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448.

[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.

[Planert et al., 2021] Planert, H., Mittermaier, F. X., Grosser, S., Fidzinski, P., Schneider, U. C., Radbruch, H., Onken, J., Holtkamp, M., Schmitz, D., Alle, H., et al. (2021). Intra-individual physiomic landscape of pyramidal neurons in the human neocortex. *bioRxiv*.

[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.

[Rao and Ballard, 1999] Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

[Shin et al., 2021] Shin, J. N., Doron, G., and Larkum, M. E. (2021). Memories off the top of your head. *Science*, 374(6567):538–539.

[Stein et al., 2021] Stein, I. S., Park, D. K., Claiborne, N., and Zito, K. (2021). Non-ionotropic nmda receptor signaling gates bidirectional structural plasticity of dendritic spines. *Cell reports*, 34(4):108664.

[Stuyt et al., 2021] Stuyt, G., Godenzini, L., and Palmer, L. M. (2021). Local and global dynamics of dendritic activity in the pyramidal neuron. *Neuroscience*.

[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.

[Vezoli et al., 2021] Vezoli, J., Magrou, L., Goebel, R., Wang, X.-J., Knoblauch, K., Vinck, M., and Kennedy, H. (2021). Cortical hierarchy, dual counterstream architecture and the importance of top-down generative networks. *Neuroimage*, 225:117479.