Learning in a Simplified Neural Compartment Model

Fabian Schubert

October 9, 2020

1 Dendritic Computation for Sequence Prediction

Shai et al. proposed a phenomenological model describing neural activity as a function of basal and distal synaptic input in pyramidal neurons [1], see Fig. 1. We further simplified this model to the following form:

$$y\left(I_p, I_d\right) = \sigma\left(I_p - \theta^{\mathrm{p}}_1\right)\sigma\left(I_d - \theta^{\mathrm{d}}\right) \; + \alpha\sigma\left(I_p - \theta^{\mathrm{p}}_0\right)\left[1 - \sigma\left(I_d - \theta^{\mathrm{d}}\right)\right] \tag{1}$$

$$\sigma\left(x\right) = \frac{1}{1 + \exp(-4g \cdot x)} \tag{2}$$

See Fig. 2 for a visualization of the relevant parameters.

We expected the nonlinearity of the neuronal output to be selective for correlated proximal and distal input, such that a Hebbian learning rule allows the neuron to be selective to synaptic input generating the most coherence between distal and proximal input. We tested this hypothesis with a single neuron using the setup illustrated in Fig. 3. The neuron receives $n = 10$ proximal input signals $(y^{\mathrm{p}}_1(t), ... y^{\mathrm{p}}_n(t)$ and a single distal input signal $y^{\mathrm{d}}(t)$. Proximal inputs were generated by simulating a chaotic random network and recording the activity of a randomly chosen unit. Each simulation run corresponds to a single proximal input signal to prevent possible correlations. The distal input was generated as a linear combination of the proximal input streams. In particular, we chose it to be an exact copy of $y^{\mathrm{p}}_1(t)$ as the most simple case for initial testing. Proximal weights were subject to Hebbian plasticity and weight normalization. The exact mathematical description of the setup is given in (3) – (8) and Table 1.

$$I_{\mathrm{p}}(t) = \sum_{k=1}^{n} w^{\mathrm{p}}_k(t) y^{\mathrm{p}}_k(t) \tag{3}$$

$$y^{\mathrm{d}}(t) = \sum_{k=1}^{n} a_k x^{\mathrm{p}}_k(t), \; \sum_{k=1}^{n} a_k = 1 \tag{4}$$

$$I_{\mathrm{d}}(t) = w^{\mathrm{d}} x^{\mathrm{d}}(t) \tag{5}$$

$$y(t) = \sigma\left(I_p(t) - \theta^{\mathrm{p}}_1\right)\sigma\left(I_d(t) - \theta^{\mathrm{d}}\right) \; + \alpha\sigma\left(I_p(t) - \theta^{\mathrm{p}}_0\right)\left[1 - \sigma\left(I_d(t) - \theta^{\mathrm{d}}\right)\right] \tag{6}$$

$$\Delta w^{\mathrm{p}}_i(t) = \epsilon_w\left(y(t) - \langle y\rangle\right)\left(y^{\mathrm{p}}_i(t) - \langle y^{\mathrm{p}}_i\rangle\right) \tag{7}$$

$$w^{\mathrm{p}}_i(t+1) = w^{\mathrm{p}}_{\mathrm{total}} \frac{w^{\mathrm{p}}_i(t) + \Delta w^{\mathrm{p}}_i(t)}{\sum_{k=1}^{n} w^{\mathrm{p}}_k(t) + \Delta w^{\mathrm{p}}_k(t)} \tag{8}$$

A First result is shown in Fig. 4. The correct proximal weight is chosen such that the proximal total input follows the distal input signal. However, when testing the

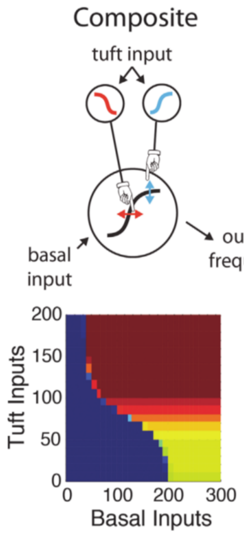

Figure 1: Firing rate as a function of distal and proximal input of the rate model proposed in [1].

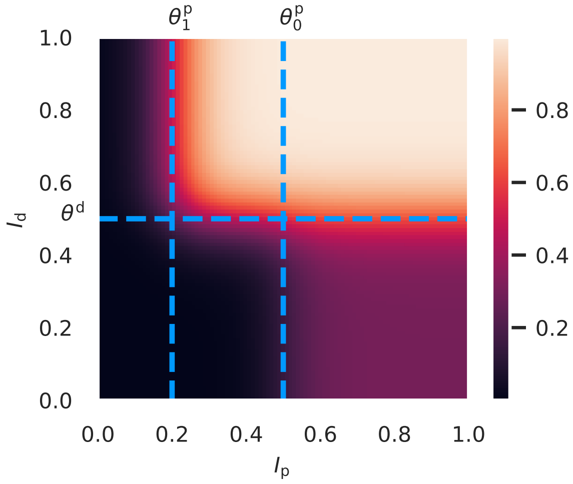

Figure 2: Output firing rate as a function of proximal and distal input as given by (1)

Table 1: Parameter settings for the setup described in (3) – (8).

w^{d}	1
θ_0^{p}	0.5
θ_1^{p}	0.2
θ^{d}	0.5
α	0.3
g	5
ϵ_w	10^{-3}
$w_{\mathrm{total}}^{\mathrm{p}}$	1

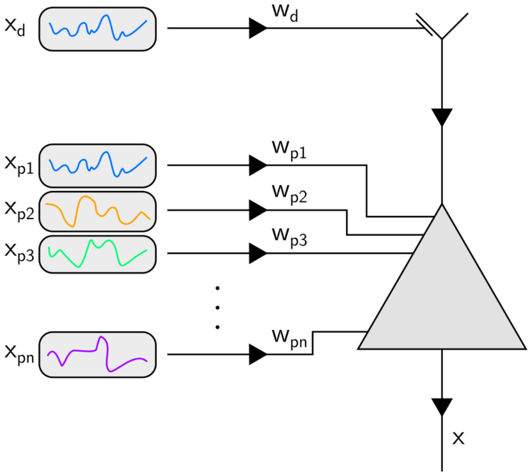

Figure 3: A single neuron receiving multiple proximal inputs and a single distal signal.

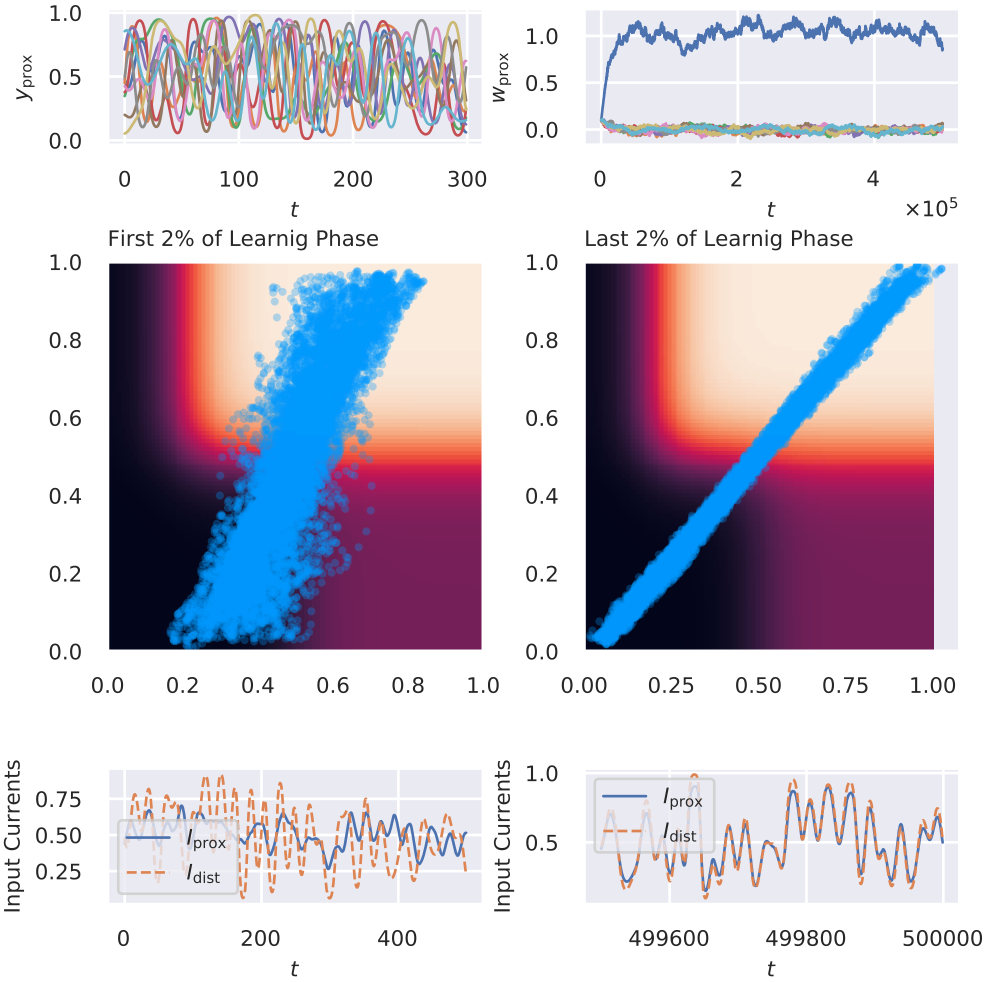

Figure 4: Weights and input signals before and after Hebbian learning of proximal weights, using nonlinear proximal-distal interaction.

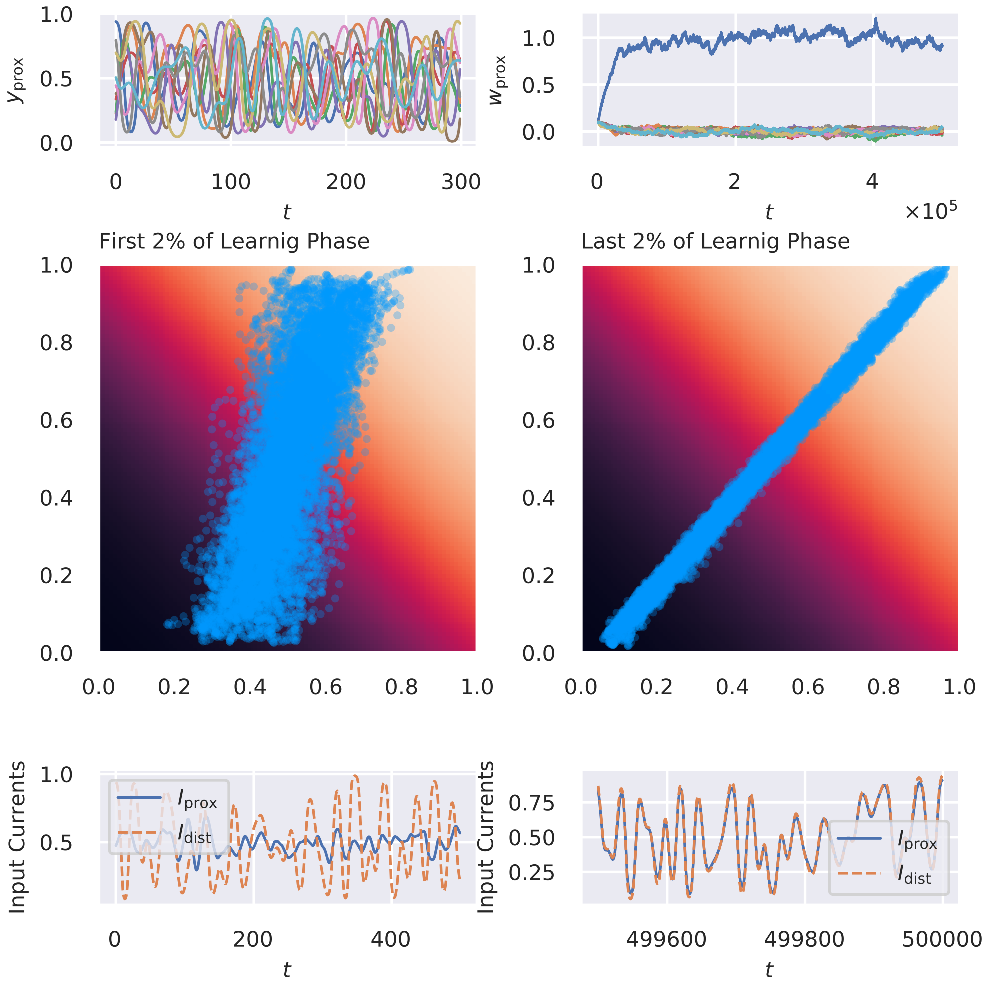

Figure 5: Weights and input signals before and after Hebbian learning of proximal weights, using linear proximal-distal summation.

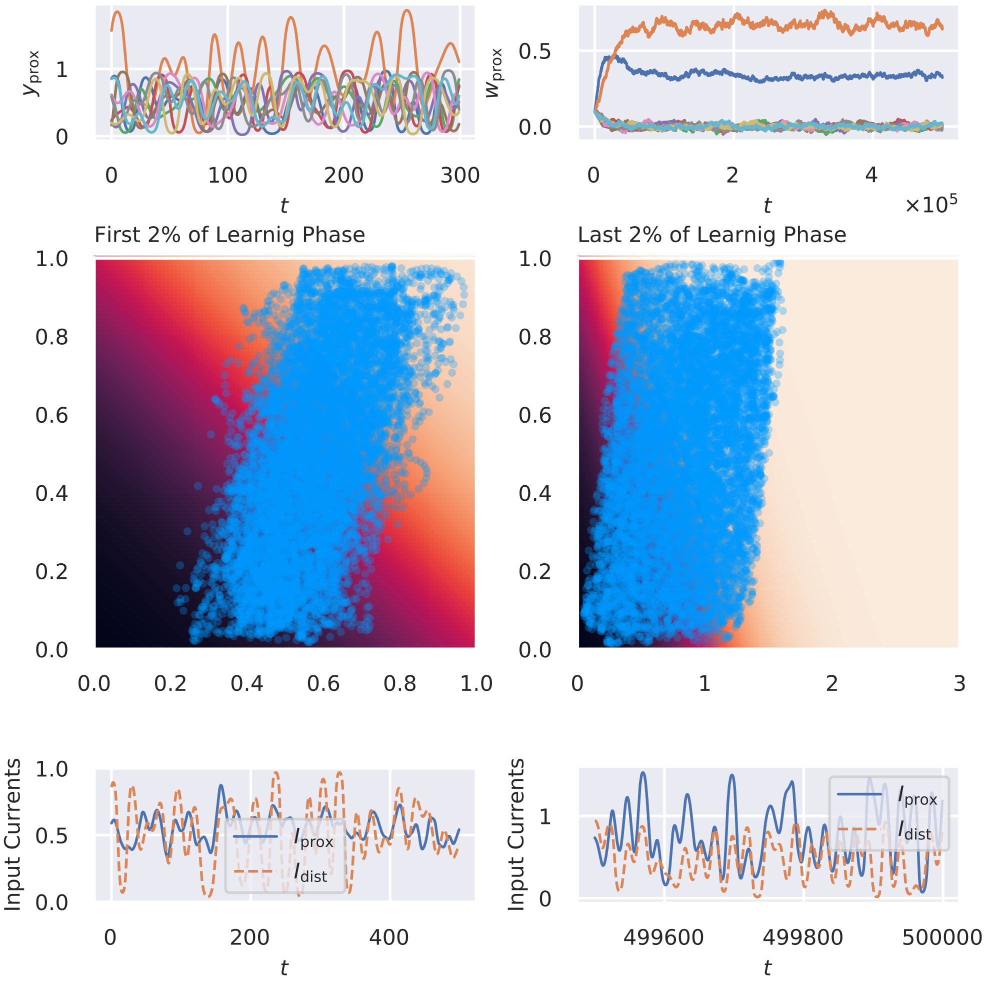

Figure 6: Same setup as in Fig. 5, but with the signal of the second proximal input channel scaled by a factor of two.

same setup except for a point-neuron summation of both proximal and distal inputs, the same result could be achieved, as shown in Fig. 5.

As a further test, we doubled the standard deviation of the second proximal input, thereby making it preferential for the classic scheme of Hebbian learning combined with linear superposition of inputs. This revealed a difference between the proximal-distal activation function and the point neuron: While the proximal-distal scheme was able to select the proximal input that would maximize correlation between proximal and distal signals, the point neuron selected the principal component of the proximal input, in this case being the second input signal. These results are shown in Fig. 6 and 7.

1.1 Output Dynamics After Learning

In the given case that after learning, the proximal input as aligned to the distal input, one might consider the case where this alignment is temporarily broken. Which input conveys more information about the output? Of course, different scenarios might be

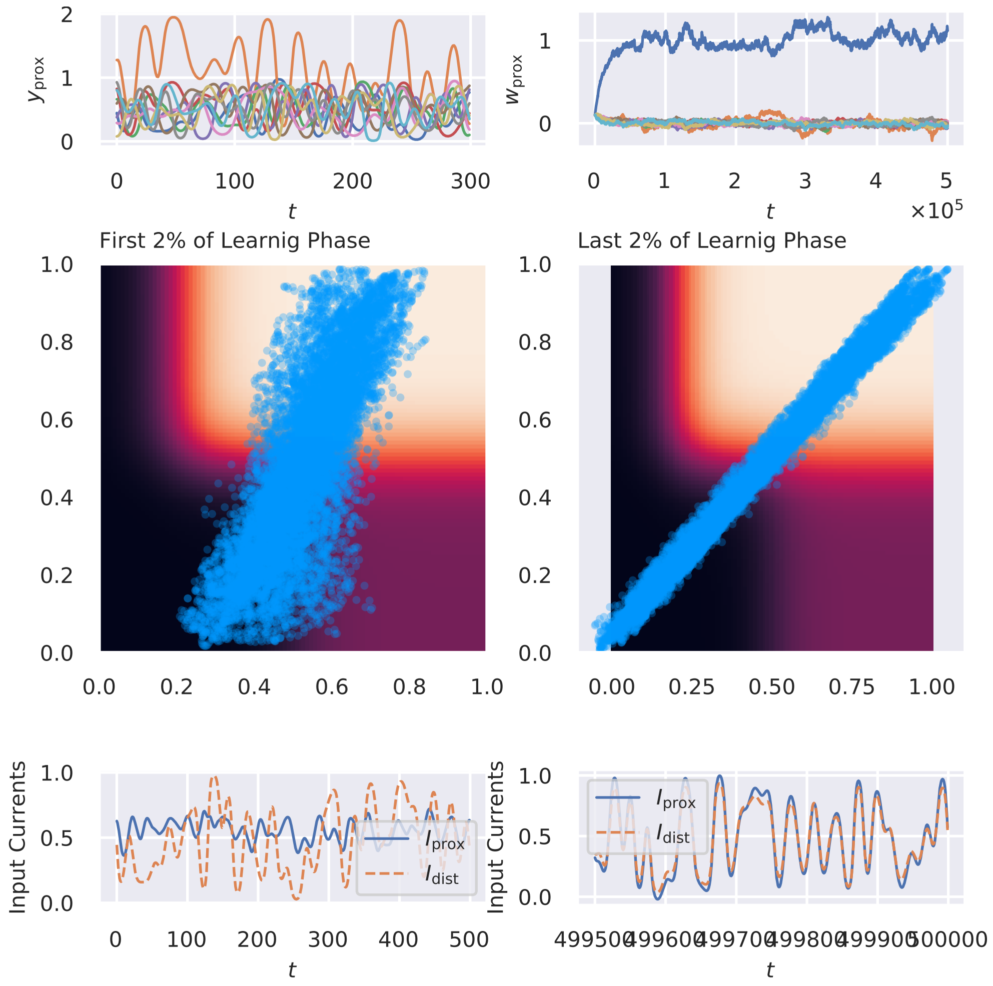

Figure 7: Same setup as in Fig. 4, but with the signal of the second proximal input channel scaled by a factor of two.

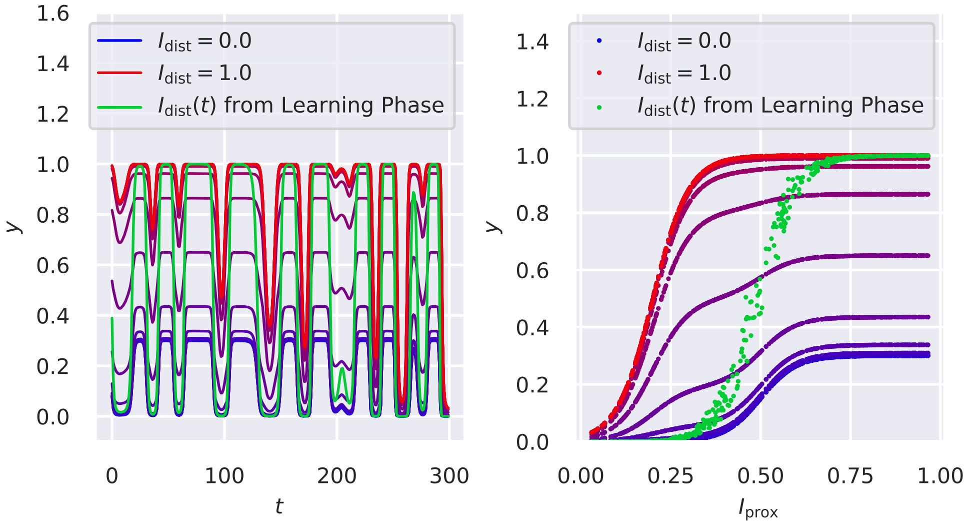

Figure 8: Output activity for a set of constant distal inputs (red to blue) applied after learning. Green line corresponds to the distal input also used for the learning phase.

considered here. First, we could consider the case where no input, or a constant level of input arrives at the distal compartment. This case is shown in Fig. 8. Changes in the distal input has a modulatory effect on the maximal output, but also shifts the bias. Speaking in terms of information transmission, this relates to two different effects: While changing the maximal output alters the overall impact of this neuron's efferent connections, changing the bias changes the actual gating for proximal input. In the next scenario, we presented uncorrelated, fluctuating input to the distal compartment. In this case, we gradually increased the maximum value for the distal input from zero to one, see Fig. 9. Obviously, increased fluctuations in I_{d} made it harder to actually extract information about I_{p} from the output. Therefore, one might wonder which input stream conveys more information to the output activity, given that both are uncorrelated and have the same overall strength. This is illustrated in Fig. 10. Although the projection onto the distal input space resulted in a wider spread of activity for strong driving, a numerical estimate of the mutual information resulted in $I(I_{\mathrm{d}}; y) \approx 2.75$ bits and $I(I_{\mathrm{p}}; y) \approx 1.87$ bits, respectively. This indicates that in the described scenario, the distal input actually conveys more information about the output as compared to the proximal input.

1.2 Analytic Approximation of Weight Dynamics

Assuming that inputs to (1) practically never reach a region where θ_{p1} becomes relevant, we remove this threshold, resulting in:

$$x\left(I_p, I_d\right) = \sigma\left(I_d - \theta_d\right) \; + \alpha\sigma\left(I_p - \theta_{p0}\right)\sigma\left(-\left(I_d - \theta_d\right)\right) \; . \tag{9}$$

We can set both thresholds to zero without loss of generality by assuming that $\langle I_p \rangle = 0$ and $\langle I_d \rangle = 0$.

To approximate the weight dynamics, we expand the activation function around the mean input (being zero by assumption) to first order. This gives

$$y\left(I_{\mathrm{p}}, I_{\mathrm{d}}\right) = y_0 + p \cdot g\alpha/2 + d \cdot g\left(1 - \alpha/2\right) + \mathcal{O}(I_{\mathrm{p}}^2, I_{\mathrm{d}}^2) \; . \tag{10}$$

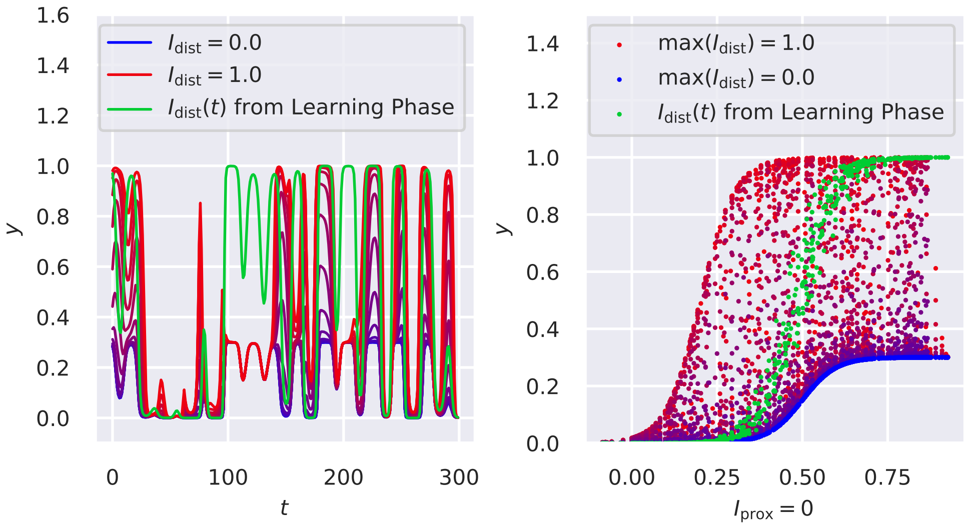

Figure 9: Output activity for a set of fluctuating distal inputs (red to blue) that are uncorrelated with the proximal input (applied after learning). Green line corresponds to the distal input also used for the learning phase.

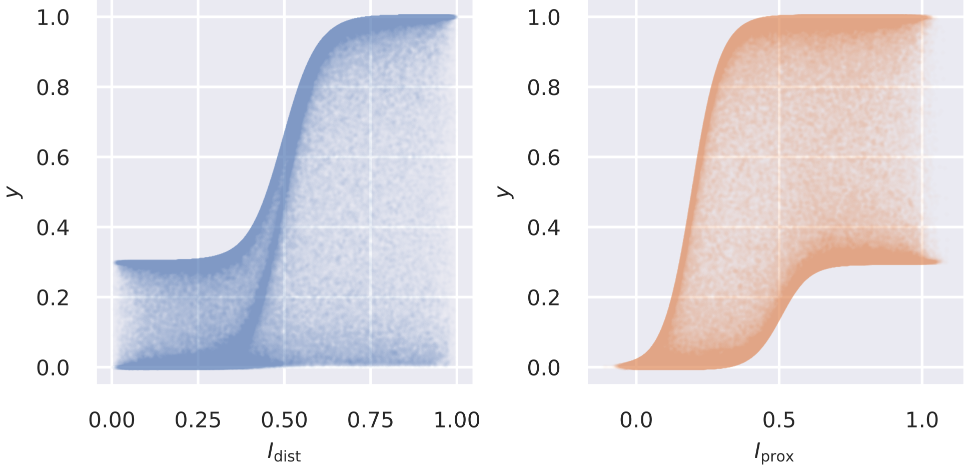

Figure 10: Activity for independently fluctuating values of I_{d} and I_{p} both ranging from 0 to 1, projected onto I_{d} (left) and I_{p} (right).

Plugging this approximation into (7), we get

$$\Delta w_i^{\mathrm{p}} \approx \epsilon_w g \left(I_{\mathrm{p}}\alpha/2 + I_{\mathrm{d}}(1-\alpha/2) - \langle I_{\mathrm{p}}\alpha/2 + I_{\mathrm{p}}(1-\alpha/2)\rangle\right)\left(x_i^{\mathrm{p}}(t) - \langle x_i^{\mathrm{p}}\rangle\right) \tag{11}$$

Further simplification of this equation leads to

$$\Delta w_i^{\mathrm{p}} \approx \epsilon'_w \left[\alpha \sum_j C_{ij}^{yy} w_j^{\mathrm{p}} + (2-\alpha) C_i^{dy}\right] \tag{12}$$

$$\epsilon'_w \equiv \epsilon_w g/2 \tag{13}$$

$$C_{ij}^{yy} \equiv \langle (y_i^{\mathrm{p}} - \langle y_i^{\mathrm{p}}\rangle)\left(y_j^{\mathrm{p}} - \langle y_j^{\mathrm{p}}\rangle\right)\rangle \tag{14}$$

$$C_i^{dy} \equiv \langle (d - \langle d\rangle)\left(y_i^{\mathrm{p}} - \langle y_i^{\mathrm{p}}\rangle\right)\rangle \tag{15}$$

We can compare (12) with the expression that we get if we calculate the negative derivative of the mean squared error between proximal and distal input, scaled by some arbitrary proportionality factor γ:

$$\Delta w_i^{\mathrm{p}} \propto -\partial_{w_i^{\mathrm{p}}} \frac{1}{2}\langle (I_{\mathrm{p}} - \gamma I_{\mathrm{d}})^2\rangle = -\sum_j C_{ij}^{yy} w_j^{\mathrm{p}} + \gamma C_i^{dy} \ . \tag{16}$$

Note that if we set α to negative values, this equation would have the same mathematical form as (12).

What has not been included in this analysis yet is synaptic normalization. In this respect, it is important to note that the fixed point that one would get from $\Delta w_i^{\mathrm{p}} = 0,\ \forall i$ is not the fixed point that is actually attained during the learning process since it is generally not compatible with the normalization condition $\|\mathbf{w}\|_1 = 1$. Rather, stable weights are achieved by a positive feedback on the weights which is constantly canceled by the synaptic normalization. By defining

$$\widehat{A} \equiv \alpha \widehat{C}^{yy} \tag{17}$$

$$\mathbf{b} \equiv (2-\alpha)\mathbf{C}^{dy} \tag{18}$$

and omitting the p-superscript in the proximal weights, the actual stationarity condition is given by

$$\frac{\left(1 + \epsilon'_w \widehat{A}\right)\mathbf{w} + \epsilon'_w \mathbf{b}}{\left\|\left(1 + \epsilon'_w \widehat{A}\right)\mathbf{w} + \epsilon'_w \mathbf{b}\right\|_1} = \mathbf{w} \tag{19}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm.

Expanding this expression to first order gives

$$\mathbf{w} + \epsilon'_w \left[\frac{\widehat{A}\mathbf{w} + \mathbf{b}}{\|\mathbf{w}\|_1} - \frac{\mathbf{w}}{\|\mathbf{w}\|_1^2}\left\|\widehat{A}\mathbf{w} + \mathbf{b}\right\|_1\right] = \mathbf{w} \tag{20}$$

$$\widehat{A}\mathbf{w} + \mathbf{b} - \mathbf{w}\left\|\widehat{A}\mathbf{w} + \mathbf{b}\right\|_1 = 0 \ , \tag{21}$$

using the fact that $\|\mathbf{w}\|_1 = 1$ by construction. In the special case of $\widehat{A}$ being simply a rescaled identity matrix, the solution would be $\mathbf{w} = \mathbf{b}/\|\mathbf{b}\|_1$. Moreover, the fixpoint of (16) would also be aligned with $\mathbf{w} = \mathbf{b}$. An alternative case where $\mathbf{b}/\|\mathbf{b}\|_1$ would be an approximately correct solution of (19) is when $\widehat{C}^{yy}$ is weak compared to $\mathbf{C}^{dy}$. However, decreasing the overall scale $\widehat{C}^{yy}$ does not affect the direction of $\mathbf{w}$ corresponding to the fixpoint of (16). Therefore, differences in the alignment between the solutions of (16) and (19) are to be expected.

It should be noted that $\mathbf{w} = \mathbf{b} / \left\|\mathbf{b}\right\|_1$ is also the stationary solution of a learning rule trying to maximize the covariance between I_{p} and I_{d} under a multiplicative normalization constraint, since (omitting means)

$$\partial_{\mathbf{w}^{\mathrm{p}}} \left\langle I_{\mathrm{p}} I_{\mathrm{d}} \right\rangle = \left\langle \mathbf{y} I_{\mathrm{d}} \right\rangle = \mathbf{C}^{dy} \ . \tag{22}$$

However, maximizing covariance is not necessarily the same as maximizing correlation:

$$\partial_{\mathbf{w}^{\mathrm{p}}} \rho \left(I_{\mathrm{p}}, I_{\mathrm{p}}\right) \propto \mathbf{C}^{dy} - \left[\mathbf{w}^{\mathrm{p}T} \widehat{C}^{yy} \mathbf{w}^{\mathrm{p}}\right] \left[\mathbf{w}^{\mathrm{p}T} \mathbf{C}^{dy}\right] \widehat{C}^{yy} \mathbf{w}^{\mathrm{p}} \tag{23}$$

.

1.3 Effect of "Distraction" Components in the Proximal Input

As already seen previously, using a compartment model forces proximal inputs to a configuration that aligns with the presented distal input, even if the proximal input has a direction of larger variance that does not align with the weight configuration that matches the distal input. We wanted to further quantify this effect systematically. To do so, we devised the following setup: A random sequence for the distal input was generated using the construction given by (4), that is, a random linear combination of the sequences also presented to the proximal input. However, the proximal input was also subject to a distracting scaling operation perpendicular to the vector $\mathbf{a}$ that defined the linear combination for the distal input. Since the number of proximal input $N_p = 10$ was larger than 2, this was done by picking a random unit vector in the hyperplane that is orthogonal to $\mathbf{a}$ and then stretching the N_p-dimensional input by a factor s along this direction. Therefore, this procedure simulated a "worst-case scenario" where the proximal input has its greatest variance in a direction perfectly orthogonal to the desired alignment of the weights. For a given scaling factor s, we generated multiple trials of this simulation and then calculated the resulting normalized mean squared error

$$MSE = \frac{\left\langle \left|\mathbf{a} - \mathbf{w}^{\mathrm{p}}\right|^2 \right\rangle}{\left\langle \left|\mathbf{a}\right|^2 \right\rangle} \ . \tag{24}$$

This was done over a range of values of s. The results are shown in Fig. 11. The compartment model preserves a much better alignment for larger values of s, with practically zero error up to a factor of approximately 2.5.

An alternative protocol was used in Fig. 12: Here, $N_p = 20$ and the distal input was composed of the average of 10 randomly picked sequences. As a distraction for the proximal input, N proximal inputs *other* than those chosen to construct the distal input were elevated by a factor of 2. Apart from the fact that, like in the previous protocol, the point neuron falls short in aligning the proximal weights, it is interesting to note that increasing the number of distracting inputs decreased its mean squared error. Presumably, this is can be explained by noting that an increased number of elevated inputs makes for a less clearly pronounced first principal component as compared to all other components.

1.4 Pattern Classification Task

We wanted to test the ability of our compartment model to learn to binary classify proximal input patterns. During the learning phase, activity patterns of dimension 500 were presented to the proximal input while a single distal input encoded the binary classification of these patterns. Before actually simulating the learning, P

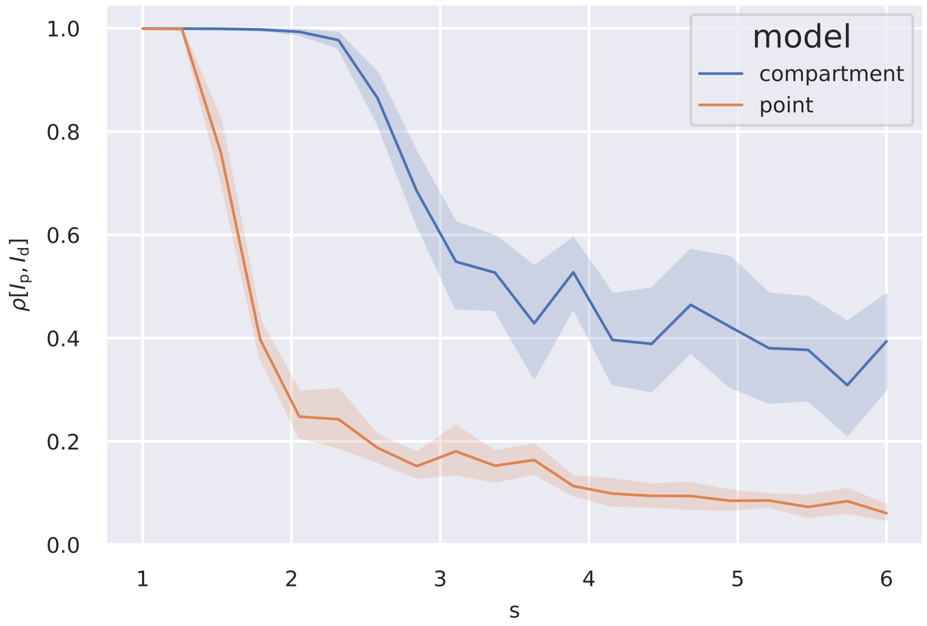

Figure 11: Pearson correlation between proximal and distal input after learning for different values of the distraction scaling parameter s.

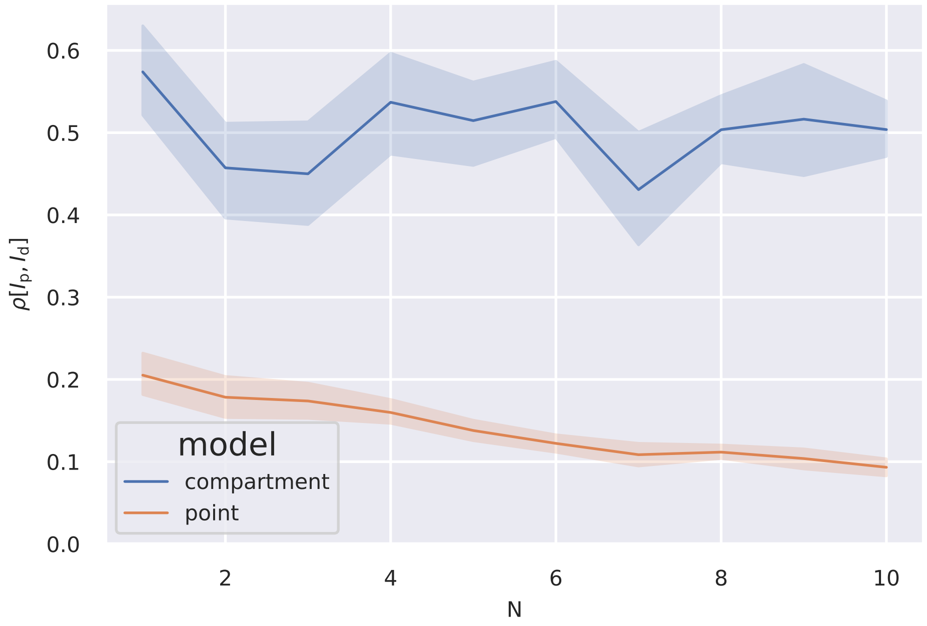

Figure 12: Pearson correlation between proximal and distal input after learning for a different number of distraction dimensions orthogonal to $\mathbf{a}$, which defines the linear combination of proximal inputs for generating I_p. Distraction scaling s was set to 2.

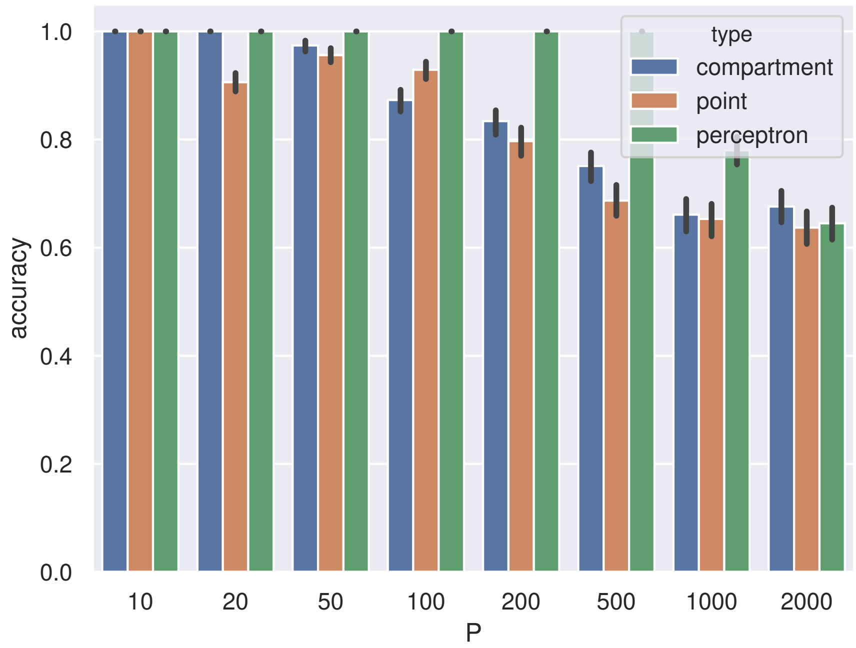

Figure 13: Binary classification accuracy for different amounts of training patterns and different neuron models.

patterns were generated by randomly setting 10% of the proximal inputs to 1, leaving the rest at 0. We randomly assigned each of those patterns to either of the two classes with equal chance (after checking that all random patterns were unique). The actual training sequence was then generated by repeatedly drawing samples from this predefined set of patterns and presenting them to the proximal input, as well as the feeding the respective 0/1 activity encoding the binary classification into to the distal input. We tested this procedure both with the compartment model as well as a simple point neuron where $y(I_\mathrm{p}, I_\mathrm{d}) = \sigma(I_\mathrm{p} + I_\mathrm{d} - \theta)$, with $\theta = 1$. Furthermore, we gradually increased the size of the set of patterns P. To test the actual classification performance, another random sequence from the pattern set was generated, however, this time switching off the distal input. A pattern was then classified as class 1 if the activity exceeded $\alpha/2$, that is, half of the maximally achievable activity in the absence of distal input. The resulting accuracy shown in Fig. 13 was estimated by the fraction of correctly classified input patterns. Apparently, a point neuron can achieve similar accuracy levels under the given experimental setup. It should be noted though that both models fall short to a perceptron model with a simple perceptron learning rule, which essentially represents the upper accuracy limit for this linear classification task. One reason why the two other models performed worse could be due to the enforced positivity of the weights, which was done for reasons of biological plausibility. Removing this constraint indeed resulted in better performance in the compartment and point model, as as shown in Fig. 14. However, substantial differences between these two models could still not be observed.

Another issue not yet addressed is the fact that biases were adjusted to fit the

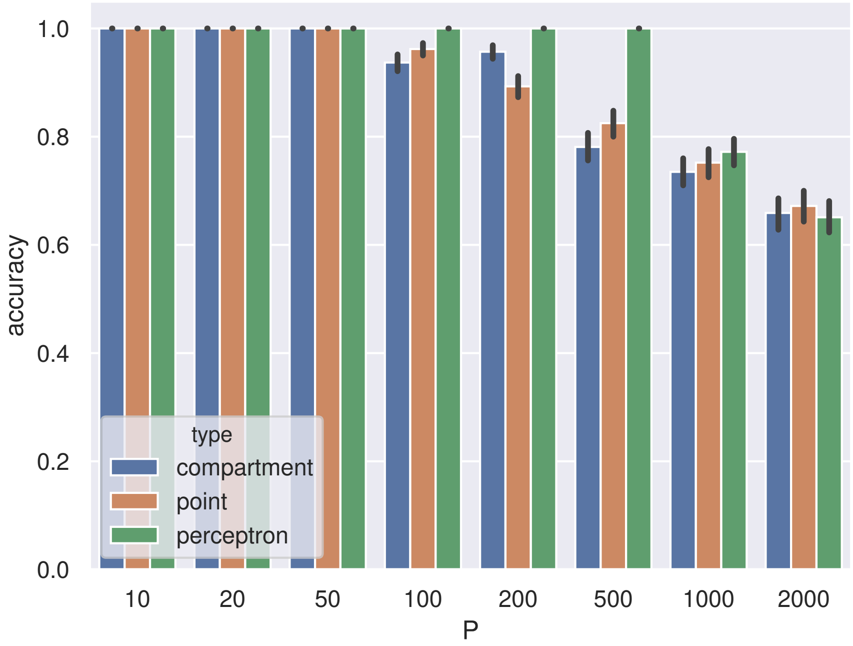

Figure 14: Binary classification accuracy for different amounts of training patterns and different neuron models. Same setup as in Fig. 13, but allowing for negative weights in the point and compartment model

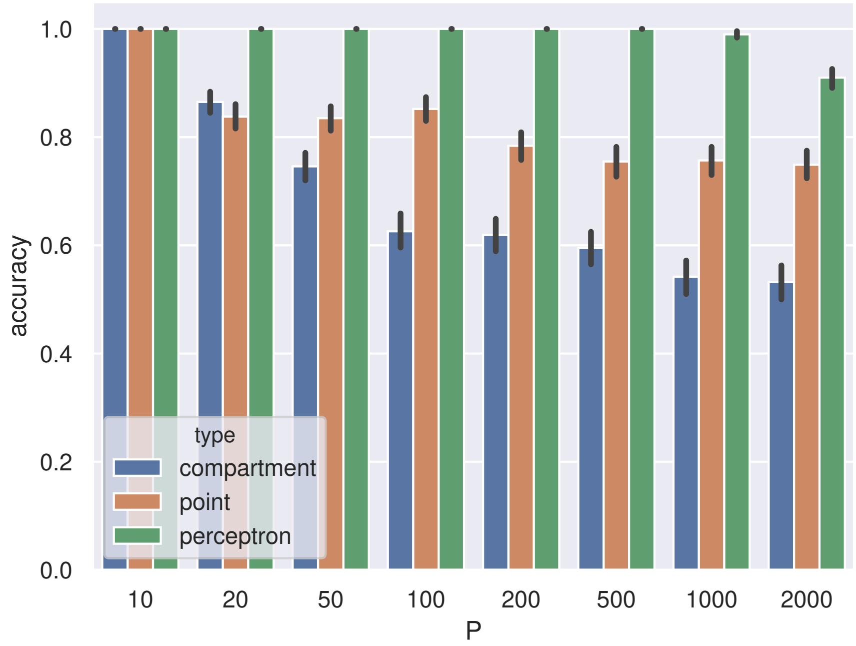

Figure 15: Binary classification accuracy for different amounts of training patterns and different neuron models. Same setup as in Fig. 13, but with class 0 appearing with 90% probability (as opposed to a 50/50 chance as in the previous setups).

average appearance of the two classes, that is, with equal probabilities. If we change this so that e.g. class 0 has a 90% chance of being present, we should expect lower performance under the same set of parameters. This case actually corresponds to a more realistic scenario where cells are sensitive to a more specific set of patterns, thus being activated more sparsely. The results are shown in Fig. 15. As expected, Hebbian learning of the weights can not account for this imbalance, whereas perceptron learning is unaffected. Of course, one could easily fix this by simply introducing some homeostatic mechanisms that adjusts the average activity to fit a given prior probability of finding the patterns that the neuron is supposed to react to. However, this is a ad-hoc solution rather than a biological explanation, except if we argue that there is ac certain heterogeneity among average neural activities and that eventually, neurons will develop receptive fields for patterns whose prior probabilities fit the neurons' given average activity.

Similar to what was investigated in Fig. 12, we also tested the classification accuracy against distraction patterns with different amounts of orthogonal distraction directions. However, since in this setup we did not explicitly define a target weight direction that could be used to construct orthogonal distraction directions, we chose to use the following procedure: Before any of the three learning schemes was used, we trained a perceptron on an unperturbed set of input patterns, which gave us a reference weight vector that could be used to rescale the input along different numbers of (random) orthogonal directions for the testing of the different neural learning schemes. Prior probabilities of the binary classification were kept at 50% in this case

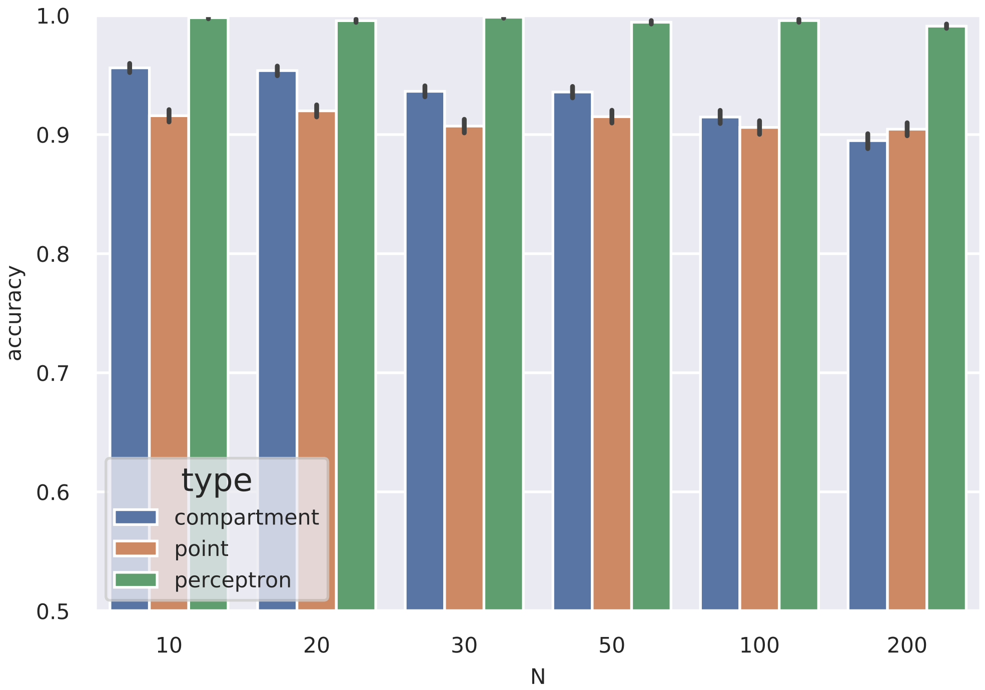

Figure 16: Classification accuracy for different neuron models and a different number of active nodes in the “distraction patterns” during the learning phase. Distraction input was turned off during testing. Averaged over 10 simulation runs.

and the number of different classification patterns was set to 100. The rescaling factor for the orthogonal directions was set to 3. The distal input still simply encoded the class associated with the respective classification pattern. Ideally, the neuron would learn to filter out the proximal information relevant for the classification, despite an increasing impact of the distraction patterns on the overall output activity. The results shown in Fig. 16 were acquired by testing the performance after learning without the distraction patterns. Though not being very prominent, one can observe a slight advantage of the compartment model over the point neuron. This is consistent with our interpretation of the plasticity in the compartment model being mostly driven by coincidence detection between the proximal and distal input, while the proximal plasticity in the point neuron is more affected by the direction of largest variance in the proximal input space.

When testing the performance after learning with the distraction patterns being present, we found the same overall trend where the compartment model performed slightly better than the point neuron, see Fig. 17.

References

[1] A. S. Shai, C. A. Anastassiou, M. E. Larkum, and C. Koch. Physiology of Layer 5 Pyramidal Neurons in Mouse Primary Visual Cortex: Coincidence Detection through Bursting. *PLOS Computational Biology*, 11(3), 2015.

[2] R. Urbanczik and W. Senn. Learning by the Dendritic Prediction of Somatic Spiking. *Neuron*, 81(3):521–528, feb 2014.

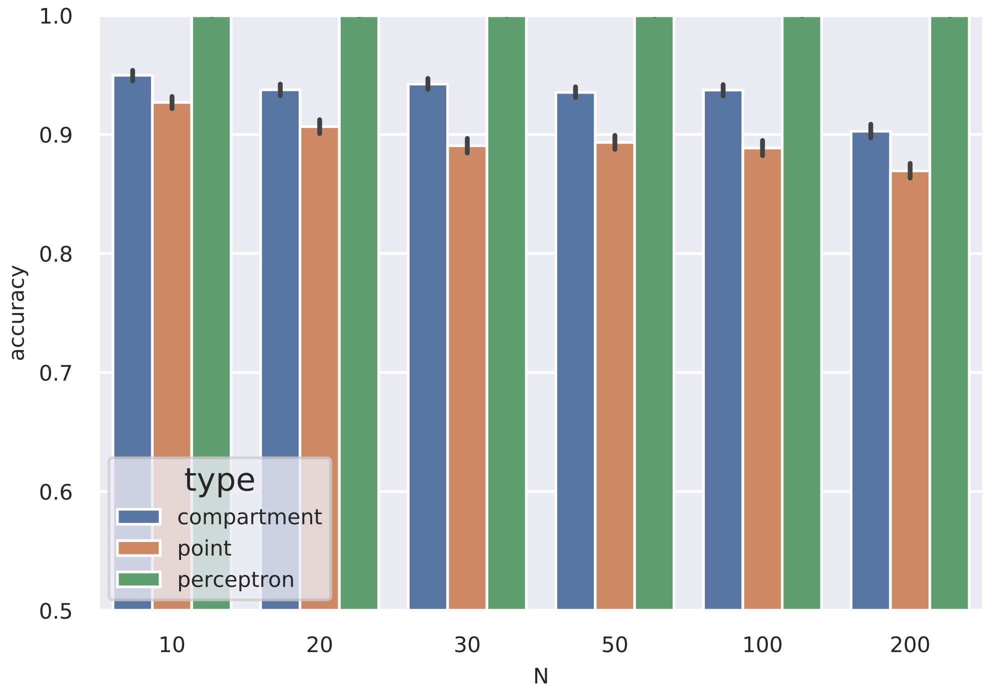

Figure 17: Classification accuracy for different neuron models and a different number of active nodes in the "distraction patterns" during the learning phase. Distraction input was turned off during testing. Averaged over 10 simulation runs.

[3] H. Barbas. General Cortical and Special Prefrontal Connections: Principles from Structure to Function. *Annual Review of Neuroscience*, 38(1):269–289, jul 2015.

[4] Matthew E. Larkum, Lucy S. Petro, Robert N. S. Sachdev, and Lars Muckli. A perspective on cortical layering and layer-spanning neuronal elements. *Frontiers in Neuroanatomy*, 12:56, 2018.

[5] Andre M. Bastos, W. Martin Usrey, Rick A. Adams, George R. Mangun, Pascal Fries, and Karl J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, Nov 2012.

[6] Hanno S. Meyer, Daniel Schwarz, Verena C. Wimmer, Arno C. Schmitt, Jason N. D. Kerr, Bert Sakmann, and Moritz Helmstaedter. Inhibitory interneurons in a cortical column form hot zones of inhibition in layers 2 and 5a. *Proceedings of the National Academy of Sciences*, 108(40):16807–16812, 2011.

[7] P. J. Sjöström and M. Häusser. A Cooperative Switch Determines the Sign of Synaptic Plasticity in Distal Dendrites of Neocortical Pyramidal Neurons. *Neuron*, 51(2):227–238, jul 2006.

[8] Johannes J. Letzkus, Björn M. Kampa, and Greg J. Stuart. Learning Rules for Spike Timing-Dependent Plasticity Depend on Dendritic Synapse Location. *Journal of Neuroscience*, 26(41):10420–10429, 2006.

[9] Frédéric Gambino, Stéphane Pagès, Vassilis Kehayas, Daniela Baptista, Roberta Tatti, Alan Carleton, and Anthony Holtmaat. Sensory-evoked ltp driven by dendritic plateau potentials in vivo. *Nature*, 515(7525):116–119, Nov 2014.

[10] Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information. *CoRR*, abs/1004.2515, 2010.

[11] Michael Wibral, Viola Priesemann, Jim W. Kay, Joseph T. Lizier, and William A. Phillips. Partial information decomposition as a unified approach to the specification of neural goal functions. *Brain and Cognition*, 112:25 – 38, 2017. Perspectives on Human Probabilistic Inferences and the 'Bayesian Brain'.

[12] Jim W. Kay, W. A. Phillips, Jaan Aru, Bruce P. Graham, and Matthew E. Larkum. Bayesian modeling of bac firing as a mechanism for apical amplification in neocortical pyramidal neurons. *bioRxiv*, 2019.

[13] William J. McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, Jun 1954.

[14] Toviah Moldwin and Idan Segev. Perceptron learning and classification in a modeled cortical pyramidal cell. *Frontiers in Computational Neuroscience*, 14:33, 2020.