Chapter 1

Neural model of spatiotemporal prediction for object recognition

1.1   Introduction

TODO

1.2   Methods

1.2.1   Model architecture

The model consisted of four layers (Figure **??**) whose parameters are described in detail in the following paragraphs. Two of the layers contained columnar substructure which was necessary for implementing learning using the LeabraTI framework (Chapter **??**). To simplify the overall LeabraTI computation, only superficial (Layers 2 and 3) and deep (Layer 6) neuron subtypes were explicitly modeled. Projections between these neuron populations correspond to the descending Layer 5 $\rightarrow$ Layer 6 synapses in the brain, which are assumed to be plastic, and the ascending Layer 6 $\rightarrow$ Layer 4 transthalamic synapses which are assumed to be relatively stable and nonplastic.

Retina and V1 preprocessing: Input was provided to the model via a 24x24 topographic filter bank that preprocessed images offline from the model proper. This preprocessing step is
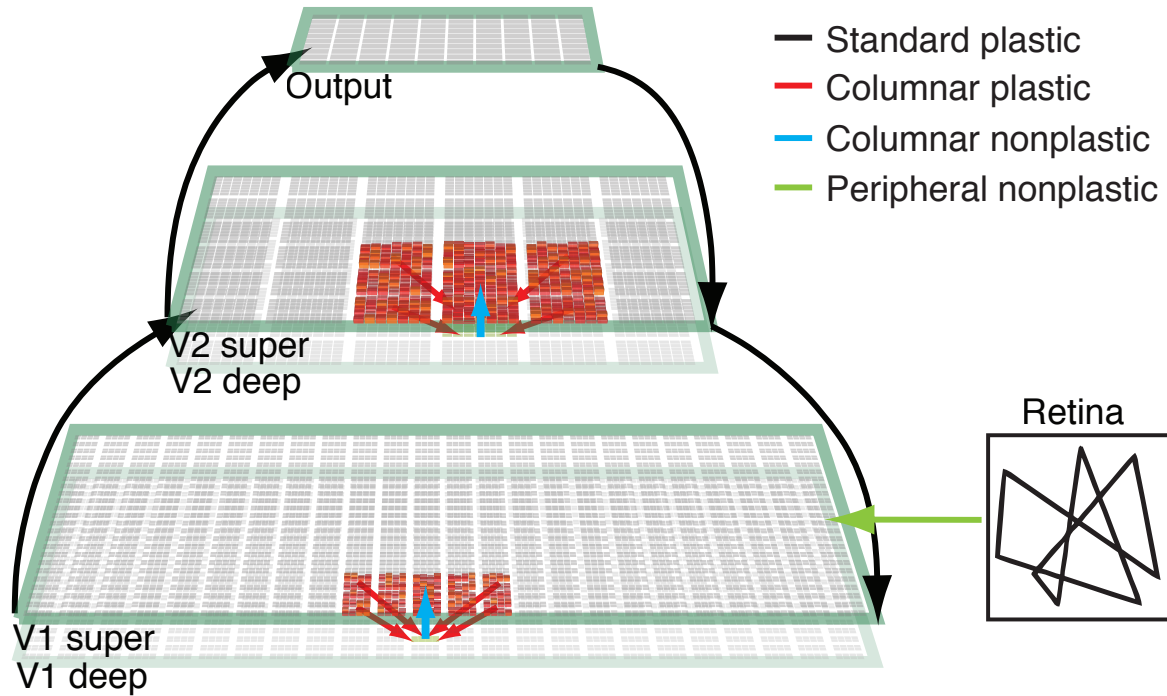
Figure 1.1: Model architecture

consistent with a large class of biological models describing object recognition in cortex (e.g., Riesenhuber & Poggio, 1999; Serre, Oliva, & Poggio, 2007; O'Reilly, Wyatte, Herd, Mingus, & Jilk, 2013) and in the case of the present model, represents visual processing from the level of the retina through V1 simple cells (Hubel & Wiesel, 1962). Grayscale bitmap images were scaled to 24x24 pixels and convolved with Gabor filters at four orientations (0°, 45°, 90°, and 135°) and two polarities (off-on and on-off) producing a 24x24x4x2 set of inputs. Each Gabor filter was implemented as 6x6 pixel kernel, with a wavelength $\lambda = 6$ and Gaussian width terms of $\sigma_x = 1.8$ and $\sigma_y = 1.2$. A static nonlinearity was applied to the output of the filtering step in the form of a modified k-Winners-Take-All (kWTA) inhibitory competition function that reduced activation across the 4x2 filter bank to the equivalent of $k = 1$ fully active units (see O'Reilly et al., 2013, Supporting Information).

Primary visual layers: 24x24 topographic layer arranged into groups of 4x2 units (4608 total units), decomposed into superficial and deep neuron subtypes. Each superficial unit received

the output of the retina/V1 preprocessing step. kWTA inhibition for superficial units was set to 60% of the average of the top k active units compared to the average of all other superficial units with each 4x2 unit group using a value of $k = 2$. Deep units received from 5x5 columns of superficial units (200 inputs per deep unit) integrated into a single value that was used as the additional context input channel for each superficial unit.

Secondary visual layers: 6x6 topographic layer arranged into groups of 7x7 units (1764 total units), also decomposed into superficial and deep neuron subtypes. Each superficial unit received from 8x8 topographical neighborhoods of early visual columns (512 afferents per unit) and sent back reciprocal connections with the same topography. kWTA for superficial units was set to 60% of the average of the top k active units compared to the average of all other superficial units with each and 15% activity within each unit group. Deep units received from 3x3 columns of superficial units (441 inputs per deep unit) integrated into a single value that was used as the additional context input channel for each superficial unit.

Output layer: 10x10 layer (100 total units) without unit group or columnar substructure. Each unit received a full projection from secondary visual columns (1764 afferents per unit) and fully projected back to all columns. A scalar of 10% was used to limit the influence of the output units on secondary visual columns during the training period, preventing runaway representation that can become disconnected from bottom-up inputs. A kWTA value of $k = 1$ was used to enforce a localist representation. The localist representation is a computational simplification that allowed an identity readout of lower-level features without population decoding similar to that provided by inferior temporal (IT) neurons (Hung, Kreiman, Poggio, & DiCarlo, 2005; Li, Cox, Zoccolan, & DiCarlo, 2009).

1.2.2   LeabraTI learning algorithm

LeabraTI was implemented as an extension of the standard Leabra algorithm which is described in detail in O'Reilly and Munakata (2000) and O'Reilly, Munakata, Frank, Hazy, and Contributors (2012). Standard Leabra learning operates across two phases: a *minus* phase that rep-

resents the system's expectation for a given input and a *plus* phase, representing observation of the outcome. The difference between the minus and plus phases, along with additional self-organizing mechanisms, is used in computing the synaptic weight update function at the end of each plus phase.

LeabraTI extends standard Leabra learning by interleaving its minus and plus phases over temporally contiguous input sequences. In standard Leabra, the minus phase depends on clamped inputs from the sensory periphery to drive the expectation while the plus phase uses clamped outputs from other neural systems to drive the outcome. In LeabraTI, the minus phase expectation is not driven by the sensory periphery, but instead by lagged context represented by deep (Layer 6) neurons. During the plus phase, driving potential shifts back to the sensory periphery. Deep neurons' context is also updated after each plus phase.

LeabraTI was only used to update the synaptic weights between superficial and deep neurons. Inter-areal feedforward and feedback projections bifurcate from the local column, directly synapsing disparate populations of superficial neurons and thus weight updates in these cases were handled by standard Leabra equations. In computing the weight update, the standard Leabra delta rule (O'Reilly, 1996) uses the difference in rate between the plus and minus phases of receiving units (y) in proportion to the rate of sending units (x) in the minus phase:

$$\Delta_{leabra} w_{ij} = x^-(y^+ - y^-)$$

In the LeabraTI framework, deep neurons are considered to be the receiving units as they are the terminus of the descending columnar synapses. However, deep units are proposed to only be active during the minus phase when they drive the prediction, and thus cannot be used to compute an error signal. To address this issue, we invert the LeabraTI delta rule:

$$\Delta_{leabrati} w_{ij} = super^-(deep^+ - deep^-)$$

$$\approx deep^-(super^+ - super^-)$$

Additionally, the temporally extended nature of the algorithm requires that the receiving units represent the current state (time t) and sending units the previous moment's state (time t - 1).

While conceptualized as the previous equation, the actual implementation is as follows:

$$\Delta_{leabrati} w_{ij} = super_{t-1}^{+}(super_{t}^{+} - super_{t}^{-})$$

This formulation allows the driving potential of deep neurons to be computed just once using the previous plus phase state of superficial neurons (multiplied by the superficial $\rightarrow$ deep learned weights) and held constant as an input to superficial neurons during the minus phase. This is a gross simplification of the actual biological process of deep neurons, but is vastly more computationally efficient than explicit modeling by computing an additional rate for each deep neuron at each time step. This formulation is also equivalent to the simple recurrent network (SRN) (Elman, 1990; Servan-Schreiber, Cleeremans, & McClelland, 1991), thus providing a potential biological substrate for its computational function.

One limitation of LeabraTI's interleaving of minus and plus phases over time is that the initial minus phase in an input sequence does not have access to the previous moment's context. Even if there was lagged context available, it would represent information from a prior, possibly unrelated input sequence. To address this, weight updates are disabled for the first minus-plus phase pair, and enabled thereafter. In the brain, this process might be facilitated by a neural mechanism that is sensitive to the repetition of inputs over time (e.g., acetylcholine) (Thiel, Henson, Morris, Friston, & Dolan, 2001; Thiel, Henson, & Dolan, 2002).

1.2.3 Training and testing environment

The same four "paper clip" objects used in the experiment described in Chapter **??** were used in training and testing the model. During training, input sequences depicted one of the four objects rotating coherently through all 30 view renderings (adjacent views spaced 12° apart). During the minus phase, the model made a prediction about the upcoming view and during the plus phase, the view was processed by the retina/V1 filter banks and clamped as an input to the model. The output unit corresponding to each object was also clamped during the plus phase bias views belonging to the same object toward similar lower-level feature representations. Training proceeded for 20

epochs of 10 input sequences each. The learning rate on all plastic projections started at 1.0 and was halved every 8 epochs.

Training efficacy was evaluated by computing the average cosine (normalized dot product) between the minus and plus phase for the primary and secondary visual layers:

$$cos\theta = \frac{1}{n}\sum_{k=1}^{n}\frac{layer_k^- \cdot layer_k^+}{||layer_k^-||||layer_k^+||}$$

The cosine varies between 0 and 1 and expresses the degree of similarity of LeabraTI's prediction to the actual outcome in layers with columnar substructure (Figure 1.2). A value of zero indicates the minus phase prediction is completely orthogonal to the plus phase sensation and a value of 1 indicates complete overlap. The lower-bound on the cosine is not likely to be zero as it would require spurious activations in topographic regions that do not contain any features. A better approximation of the lower bound is the case when the system simply reproduces the plus phase from the previous moment's (t - 1) state. This can be thought of as the amount of perceptual overlap between adjacent views of the stimuli, and thus any additional features that contribute to a higher cosine value indicate positive prediction.

The response rate of the output unit corresponding to the each object was also used to evaluate training. This was necessary as stimulus predictability and stimulus-response mappings can be learned independently (Wyart, Nobre, & Summerfield, 2012; Kok, Rahnev, Jehee, Lau, & de Lange, 2012). In neural models, this is characterized by runaway representations in higher-level layers that become disconnected from the bottom-up inputs. The problem is most apparent in deep architectures with many layers and several solutions exist such as staged pairwise training of adjacent layers (Hinton & Salakhutdinov, 2006).

The present model was compact enough that staged training was not necessary, but an additional method was developed to ensure that predictive ability was consistent with stimulus-response mappings. Specifically, each plastic projection in the model contained two sets of synapses with different update intervals that contribute a weighted mixture to the input of each receiving unit. The first "standard" set of synapses were updated after every plus phase, whereas a second "sta-
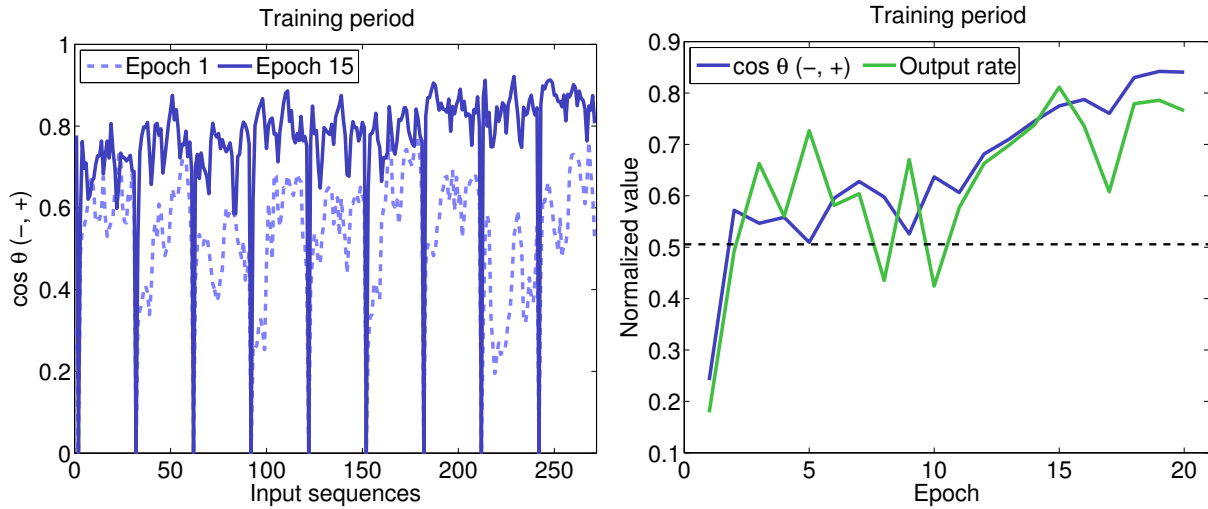
Figure 1.2: Model training

Average cosine between minus and plus phase for layers with columnar substructure and output response rate over the course of training. Sharp drops in the cosine to zero indicate the start of a new input sequence and are unlearnable. The lower bound for the cosine was computed as the reproduction of the plus phase from the previous moment's (t - 1) state and the overall average is indicated by the dotted line. A cosine greater than this level indicates positive prediction.

ble" set of synapses were updated at the end of each epoch. In the present model, a 80% stable to

20% standard synaptic mixture was used. This allowed the network to more slowly integrate learn-

ing across an entire epoch's worth of input sequences without runaway representations caused by

being exposed to the same rotating object's features over multiple time steps while still maintaing

the moment-to-moment spatiotemporal predictive learning central to LeabraTI.

Testing

1.3   Results

1.4   Discussion

References

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507.

Hubel, D., & Wiesel, T. N. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. Journal of Physiology, 160(1), 106–154.

Hung, C. P., Kreiman, G., Poggio, T., & DiCarlo, J. J. (2005). Fast readout of object identity from macaque inferior temporal cortex. Science, 310(5749), 863–866.

Kok, P., Rahnev, D., Jehee, J. F. M., Lau, H. C., & de Lange, F. P. (2012). Attention reverses the effect of prediction in silencing sensory signals. Cerebral Cortex, 22(9), 2197–2206.

Li, N., Cox, D., Zoccolan, D., & DiCarlo, J. (2009). What response properties do individual neurons need to underlie position and clutter "invariant" object recognition? Journal of Neurophysiology, 102(1), 360–376.

O'Reilly, R. C. (1996). Biologically Plausible Error-driven Learning using Local Activation Differences: The Generalized Recirculation Algorithm. Neural Computation, 8(5), 895–938.

O'Reilly, R. C., & Munakata, Y. (2000). Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain. Cambridge, MA: The MIT Press.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., & Contributors (2012). Computational Cognitive Neuroscience. Wiki Book, 1st Edition, URL: http://ccnbook.colorado.edu.

O'Reilly, R. C., Wyatte, D., Herd, S., Mingus, B., & Jilk, D. J. (2013). Recurrent processing during object recognition. Frontiers in Psychology, 4, 124.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. Nature Neuroscience, 2(11), 1019–1025.

Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. Proceedings of the National Academy of Sciences of the United States of America, 104(15), 6424–6429.

Servan-Schreiber, D., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. Machine Learning, 7(2–3), 161–193.

Thiel, C. M., Henson, R. N., Morris, J. S., Friston, K. J., & Dolan, R. J. (2001). Pharmacological modulation of behavioral and neuronal correlates of repetition priming. The Journal of Neuroscience, 21(17), 6846–6852.

Thiel, C. M., Henson, R. N. A., & Dolan, R. J. (2002). Scopolamine but not lorazepam modulates face repetition priming: A psychopharmacological fMRI study. Neuropsychopharmacology, 27(2), 282–292.

Wyart, V., Nobre, A. C., & Summerfield, C. (2012). Dissociable prior influences of signal probability and relevance on visual contrast sensitivity. Proceedings of the National Academy of Sciences of the United States of America, 109(9), 3593–3598.