TASK-OPTIMIZED RETINA-INSPIRED CNN CONVERGES TO BIOLOGICALLY PLAUSIBLE FUNCTIONALITY

Keith T. Murray
Massachusetts Institute of Technology
ktmurray@mit.edu

Mien Brabeeba Wang
Massachusetts Institute of Technology
brabeeba@mit.edu

Nancy Lynch
Massachusetts Institute of Technology
lynch@csail.mit.edu

ABSTRACT

Convolutional neural networks (CNN) are an emerging technique in modeling neural circuits and have been shown to converge to biologically plausible functionality in cortical circuits via task-optimization. This functionality has not been observed in CNN models of retinal circuits via task-optimization. We sought to observe this convergence in retinal circuits by designing a biologically inspired CNN model of a motion-detection retinal circuit and optimizing it to solve a motion-classification task. The learned weights and parameters indicated that the CNN converged to direction-sensitive ganglion and amacrine cells, cell types that have been observed in biology, and provided evidence that task-optimization is a fair method of building retinal models. The analysis used to understand the functionality of our CNN also indicates that biologically constrained deep learning models are easier to reason about their underlying mechanisms than traditional deep learning models.

Keywords Convolutional Neural Network · Biologically Plausible · Task-Optimization

1 Introduction

The retina serves as the first step in visual processing for the brain in nearly all animal species (Baden et al. [2020]). While it may serve only as a first step, the retina processes visual information using complex neural circuits that have yet to be fully understood or modeled (Gollisch and Meister [2010]). As an understanding of what these complex retinal circuits do has evolved, so too have the models used to explain these circuits. Linear-nonlinear (LN) models that linearly filter and nonlinearly transform visual data (Ölveczky et al. [2003], Hosoya et al. [2005], Baccus et al. [2008]) have since been replaced by deep convolutional neural networks (CNNs) that are able to more fully predict retinal cell activity through filtering and nonlinearly transforming information through many layers with weights adjusted via backpropagation algorithms (McIntosh et al. [2016], Maheswaranathan et al. [2019], Tanaka et al. [2019]). This switch came as CNNs were also shown to accurately predict cortical activity and converge to cortical representations in a variety of perceptual tasks after task-optimization training (Yamins et al. [2014], Kell et al. [2018]).

CNNs that converge to cortical representations via task-optimization training provide a possible explanation for why neural circuits in the brain are organized in a particular fashion (Yamins and DiCarlo [2016], Saxe et al. [2020]). While such CNN modeling has provided evidence that CNNs can model retinal circuits, this modeling was driven by optimization that sought to fit retinal cell data and thus cannot provide theoretical evidence as to why retinal circuits are organized in their current fashion (McIntosh et al. [2016], Maheswaranathan et al. [2019], Tanaka et al. [2019]). We sought to find this evidence through designing biologically-constrained CNNs that are trained to optimize performance in a motion-classification task. This approach also requires developing methods to understand the trained CNN, methods which have previously been elusive (Barrett et al. [2019]), and could provide new algorithms for solving the motion-classification tasks.

2 Methods

A two-directional motion-classification task was designed that required the CNN to output the direction corresponding to the population of dots that had the greater speed. The CNN processed this task-stimulus via three layers, with each layer having 8 different cells types, and a residual connection that correspond to previous motion-detection literature (Ölveczky et al. [2003], Baccus et al. [2008], Gollisch and Meister [2010]). A linear decoder mapped ganglion cell activity to correspond to left-, and right-direction classes that could then be used to determine the task accuracy of the CNN. The CNN was trained using the Adam optimization algorithm (Kingma and Ba [2014]) and ablated. The functionality of the cell types in the ablated network were classified using deep visualization (Yosinski et al. [2015]) and neurophysiology techniques (Ölveczky et al. [2003], Hosoya et al. [2005], Ozuysal and Baccus [2012]).

2.1 Task Design

The task design was inspired by neuroscience literature investigating the function of the middle temporal visual area (MT) of the visual cortex (Newsome and Pare [1988]). In the literature, MT-related tasks involved motion discrimination between a population of randomly moving dots and dots moving synchronously in a particular direction. We drew inspiration from this aspect of population discrimination among moving dots but used speed as the discriminating factor instead of motion synchronicity among the dots.

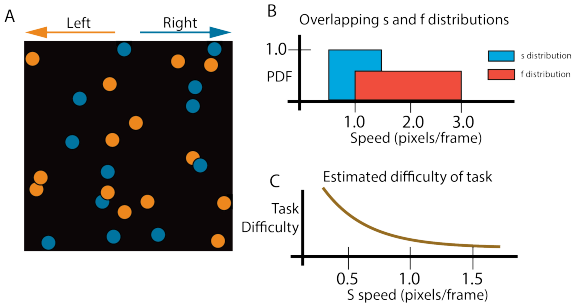

Figure 1: Illustrations of task design. (A) An example stimulus frame. Note, there are no colors in the actual stimulus. (B) The s and f distributions and the overlap between them. This overlap adds another layer of difficulty. (C) The estimated difficulty of the task decreases as the S speed increase and creates a larger separation from F.

Our task consisted of randomly placed dots on a field moving either left or right (Figure 1A). All dots that moved in a particular direction would be assigned a speed and one direction would have a greater speed. To solve the task, the

model would have to indicate which population of dots, right-directional or left-directional, had the greater speed. We hypothesized that a CNN that could solve this task would have to encode motion. The mechanisms that the CNN used to encode motion could then be further studied to describe what mechanisms a retina could use to encode motion.

The data set consisted of 1000 stimuli with each stimulus being an instance of the task. Each instance of the task was a 255 by 255 dimensional video consisting of 51 frames. For each stimuli, the placement of the dots was randomly initialized with an average of 16.67 dots being in one frame at any given frame. Each dot was assigned a direction, right or left, with uniform probability. For each stimulus, the speed for the right- and left-directional was determined via the following method:

1. The slow direction (S) is chosen uniformly between left and right.
2. S is assigned a speed by $s \sim U(0.5, 1.5)$.
3. The fast direction (F) is assigned a speed by $f = \alpha * s$, where α is the velocity multiplier.

The environment of the data set was chosen to have an α variable of 2 for training and testing the model (Figure 1B). The distribution of s determined the difficulty of the stimulus. An s near the lower end of the distribution is more difficult because the difference between f and s is smaller and less perceptible (Figure 1C).

2.2 Model Architecture and Training

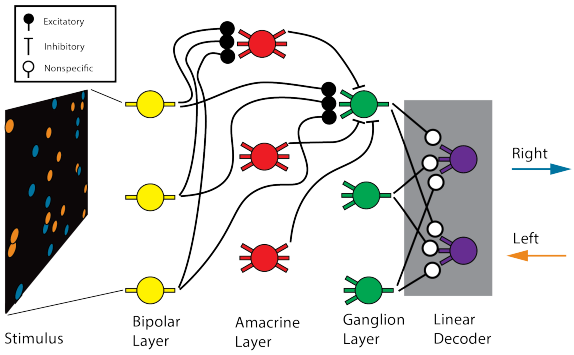

Figure 2: Diagram of the CNN and its connections. Only 3 cell types per layer are depicted here, but 8 per layer were used. Each cell type consists of a spatial convolution denoted by the connections, an internal temporal convolution, and a ReLU activation function.

The CNN consisted of 3 layers where each layer modeled the bipolar, amacrine, or ganglion layer. Each layer was connected to the next layer, and an extra direct connection, a *residual* connection, existed between the bipolar and ganglion layers (Figure 2). While *residual* connections have been shown to increase performance of CNNs in object-recognition tasks (He et al. [2016]), our motivation for inserting a residual connection in our CNN came from an observed direct connection between the bipolar and ganglion layers in the retina (Ölveczky et al. [2003], Baccus et al. [2008]). Previous CNN models of the retina have not accounted for this residual connection (McIntosh et al. [2016], Maheswaranathan et al. [2019], Tanaka et al. [2019]). By including a residual connection, a connection constraint between cell types could be enforced that had previously not existed in other retinal CNN models (McIntosh et al. [2016], Maheswaranathan et al. [2019], Tanaka et al. [2019]) but has been biologically observed (Ölveczky et al. [2003], Baccus et al. [2008], Gollisch and Meister [2010]). Connection enforcement was performed through constraining weights in the amacrine and ganglion layers to be positive (negative weights were zeroed out) and taking the additive inverse from the otherwise positive output of the amacrine layer.

Each cell layer in our CNN included 8 different cell types. For example, in the bipolar layer, there were 8 different bipolar cell types where each type had access to the same information as every other type. In the amacrine and ganglion layers, each cell type had connections to every cell type in the previous layer (e.g. amacrine cell type 0 had connections to bipolar cell types 0-7). While this connection pattern may not be biologically constrained, this connection pattern is a principled pattern for allowing emergent representations through learning.

Each cell type consisted of a spatial convolution, temporal convolution, and a rectified linear unit (ReLU) activation. For the bipolar layer, spatial convolutions are analogous to information received from photoreceptors, the fundamental unit of the retina (Gollisch and Meister [2010]), and for the amacrine and ganglion layers, spatial convolutions represent the connections between the layers. The residual connection took shape in the form of an additional spatial convolution

between the bipolar and ganglion layers. The temporal convolution in the bipolar layer is analogous to the feedback from horizontal cells and analogous to the amacrine and ganglion layers represents the decay of activity. The ReLU activation was chosen because of its use in previous modeling (Ölveczky et al. [2003], Baccus et al. [2008], McIntosh et al. [2016], Maheswaranathan et al. [2019], Tanaka et al. [2019]).

The output from the ganglion cell layer was fed through a linear decoder that gave two outputs, each corresponding to a direction (left or right). Feeding the stimulus through the CNN model created 51 outputs from the linear decoder. The linear decoder output with the greater activation on the last frame of the stimulus would indicate the direction the model predicted of the F direction. The linear decoder was used because the output could be used for training via a cross-entropy loss function (Paszke et al. [2019]),

$$L(y, class) = -y[class] + \log(\textstyle\sum_j \exp(y[j])),$$

where y is the output from the linear decoder and $class$ indicates whether the left or right direction is the F direction.

Our CNN was trained over the course of 500 epochs using the Adam optimization algorithm (Kingma and Ba [2014]) and a dropout rate of 40% via the PyTorch libraries (Paszke et al. [2019]).

2.3 Analysis Methods

The analysis of any deep learning model has been shown to be a difficult undertaking (Barrett et al. [2019]), but the analysis of our model was inspired by the analysis of previous biological retinal circuits and CNN models (Ölveczky et al. [2003], Hosoya et al. [2005], Maheswaranathan et al. [2019], Tanaka et al. [2019]). The trained CNN's performance was first evaluated on a variety of environments where the s and α variables were systematically manipulated to reveal how robust the CNN was and give insights into its functionality.

In line with previous deep learning literature working to understand how deep learning models work (Blalock et al. [2020]), the CNN was ablated to gain an understanding about the fundamental operations performed by each cell layer. Our methods for ablating differ from previous literature in that cell types were ablated instead of randomized parameters or some other algorithm (Blalock et al. [2020]). Individual cell types were ablated and the model's accuracy was tested to establish which cell types were necessary.

After ablation, each cell type's functional role was established via functional observations in performance on test sets and reasoning about their respective deep visualization. The functional observations used in analyzing cell types included measuring the response amplitude of a cell, analyzing the accuracy of the ablated model when further ablated, and reasoning about the cell type's convolutional kernels. A deep visualization has been shown to be particularly useful in classifying the functional relevancy of neural units in neural networks (Yosinski et al. [2015]) but has not previously been utilized in classifying the functional relevancy of neural units in CNN models of neural circuits. By treating the model as a static transformer, the individual pixels in the stimulus can be treated as parameters that are tuned to maximize the response of a particular neuron. Deep visualizations are particularly useful when gaining an intuition about how the spatial convolution and temporal convolution profiles interact.

The final step in the analysis was to merge an understanding of how all cell layers function into an understanding as to how the CNN solves the task as a whole. The objective was to explain the performance across various environments observed in the first step and to draw conclusions about how biologically plausible the functionality of the CNN was.

3 Results

The trained model exhibited robustness over various environments containing a range of α values (Figure 3A). This robustness indicated that the model did encode for motion and that further analysis could be done to understand how the CNN model did this motion encoding. The full model had 24 cell types in total and we sought to understand the model by analyzing an ablated version. By examining all possible combinations of 4 cell types, we found an ablated version of the CNN that only had bipolar cell type 4, amacrine cell type 2, and ganglion cell type 0 and 2 that exhibited a relatively high performance of 70% on the test set. The ablated model displayed a similar pattern of robustness as the full model but to a lesser extent (Figure 3A).

The functionality of the bipolar cell layer was understood by examining the performance of the ablated model with each bipolar cell type substituted in. A data set in a range of speeds and dots moving only in the rightward direction was fed into the model. In Figure 3B, as the speed of the stimuli increases, the ablated models with the various bipolar cell types begins to correctly solve the task at different speeds. This performance demonstrates that the bipolar cell types responded to speeds after they crossed a certain threshold. This threshold effect may serve to compensate for the overlapping distribution of speeds in the training set (Figure 1B).

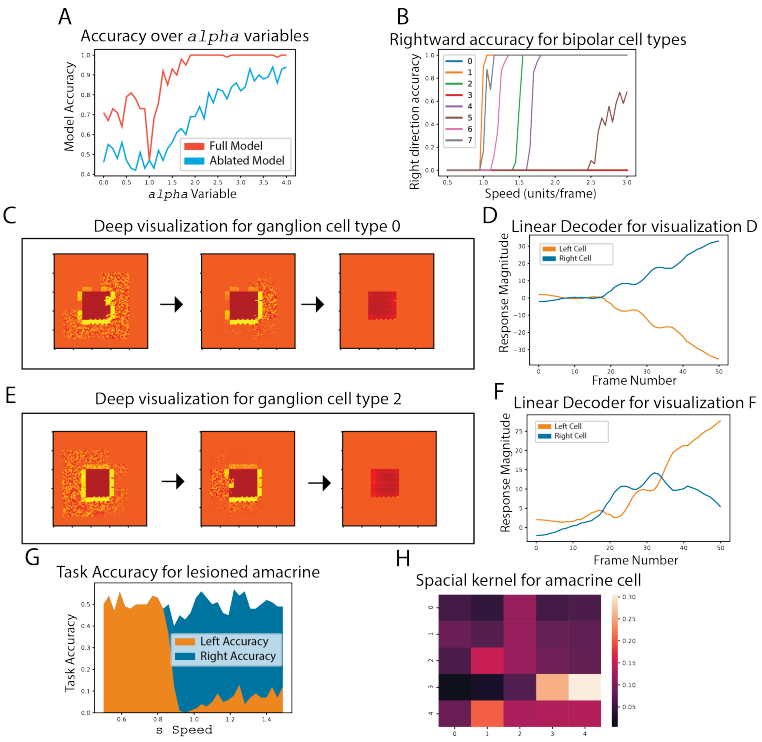

Figure 3: Summary figure for the analysis of the model. (A) Psychometric curve for the α parameter of full and ablated model. (B) Accuracy in the rightward direction of the ablated model with each bipolar cell type. Each bipolar cell shows a clear speed threshold. (C) Deep visualization for ganglion cell type 2. This deep visualization displays activity that fades toward the left direction, indicating direction-selectivity. (D) Activity of the linear decoder when the deep visualization (C) was input. This activity confirms that ganglion cell type 2 is direction-selective for the left direction. (E) Deep visualization for ganglion cell type 0. The deep visualization displays activity that fades toward the right direction, indicating direction-selectivity. (F) Activity of the linear decoder when the deep visualization (E) was input. This activity confirms that ganglion cell type 0 is direction-selective for the right direction. (G) Relative left and right accuracies for the ablated model with the amacrine cell type lesioned. The model biased towards the right direction after bipolar activation indicating that the amacrine cell is right direction-selective. (H) Spacial convolutional kernel for the ablated amacrine cell type. There is an increase in weights toward the right direction confirming the selectivity apparent in (G).

The functionality of the ganglion cell types were understood by using deep visualizations (Yosinski et al. [2015]) to visualize what stimulus would optimally activate the ganglion cell types in the ablated model. The deep visualization for ganglion cell type 0 indicated that dots moving in the rightward direction would optimally activate the cell type because the deep visualization fades to the right (Figure 3C). The deep visualization for ganglion cell type 0 was then fed into the ablated model and the linear decoder outputted that the rightward direction was present in the stimulus (Figure 3D). This same direction-selectivity of ganglion cell type 0 was also found in ganglion cell type 2 (Figure 3E-f) but for the leftward direction instead. This observed direction selectivity of ganglion cells is also observed in biological retinas (Wei [2018]) and indicates that our CNN model of the retina did converge to a biologically plausible model via task-optimization.

The functionality of the amacrine cell layer was also observed to be biologically plausible because the spacial kernel of amacrine cell type 2 in the ablated model displayed direction selectivity (Figure 3H). This direction-selectivity in the amacrine cell type is critical to the functionality of the model (Figure 3G) and is similar to the functionality of starburst amacrine cells (SACs) observed in biological retinas (Wei [2018]).

4 Future Directions

The task-optimized CNN was shown to converge a biologically plausible functionality. Future directions include exploring how the bipolar cells threshold according to speed and understanding the algorithmic basis of the model.

References

Tom Baden, Thomas Euler, and Philipp Berens. Understanding the retinal basis of vision across species. *Nature Reviews Neuroscience*, 21(1):5–20, 2020.

Tim Gollisch and Markus Meister. Eye smarter than scientists believed: neural computations in circuits of the retina. *Neuron*, 65(2):150–164, 2010.

Bence P Ölveczky, Stephen A Baccus, and Markus Meister. Segregation of object and background motion in the retina. *Nature*, 423(6938):401–408, 2003.

Toshihiko Hosoya, Stephen A Baccus, and Markus Meister. Dynamic predictive coding by the retina. *Nature*, 436 (7047):71–77, 2005.

Stephen A Baccus, Bence P Ölveczky, Mihai Manu, and Markus Meister. A retinal circuit that computes object motion. *Journal of Neuroscience*, 28(27):6807–6817, 2008.

Lane T McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen A Baccus. Deep learning models of the retinal response to natural scenes. *Advances in neural information processing systems*, 29:1369, 2016.

Niru Maheswaranathan, Lane T. McIntosh, Hidenori Tanaka, Satchel Grant, David B. Kastner, Josh B. Melander, Aran Nayebi, Luke Brezovec, Julia Wang, Surya Ganguli, and Stephen A. Baccus. The dynamic neural code of the retina for natural scenes. *bioRxiv*, 2019. doi:10.1101/340943. URL `https://www.biorxiv.org/content/early/2019/12/17/340943`.

Hidenori Tanaka, Aran Nayebi, Niru Maheswaranathan, Lane McIntosh, Stephen A Baccus, and Surya Ganguli. From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. *arXiv preprint arXiv:1912.06207*, 2019.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, pages 1–13, 2020.

David GT Barrett, Ari S Morcos, and Jakob H Macke. Analyzing biological and artificial neural networks: challenges with opportunities for synergy? *Current opinion in neurobiology*, 55:55–64, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Yusuf Ozuysal and Stephen A Baccus. Linking the computational structure of variance adaptation to biophysical mechanisms. *Neuron*, 73(5):1002–1015, 2012.

William T Newsome and Edmond B Pare. A selective impairment of motion perception following lesions of the middle temporal visual area (mt). *Journal of Neuroscience*, 8(6):2201–2211, 1988.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Wei Wei. Neural mechanisms of motion processing in the mammalian retina. *Annual review of vision science*, 4: 165–192, 2018.