A ROTATION-EQUIVARIANT CONVOLUTIONAL NEURAL NETWORK MODEL OF PRIMARY VISUAL CORTEX

Anonymous authors
Paper under double-blind review

ABSTRACT

Classical models describe primary visual cortex (V1) as a filter bank of orientation-selective linear-nonlinear (LN) or energy models, but these models fail to predict neural responses to natural stimuli accurately. Recent work shows that models based on convolutional neural networks (CNNs) lead to much more accurate predictions, but it remains unclear which features are extracted by V1 neurons beyond orientation selectivity and phase invariance. Here we work towards systematically studying V1 computations by categorizing neurons into groups that perform similar computations. We present a framework to identify common features independent of individual neurons' orientation selectivity by using a rotation-equivariant convolutional neural network, which automatically extracts every feature at multiple different orientations. We fit this model to responses of a population of 6000 neurons to natural images recorded in mouse primary visual cortex using two-photon imaging. We show that our rotation-equivariant network not only outperforms a regular CNN with the same number of feature maps, but also reveals a number of common features shared by many V1 neurons, which deviate from the typical textbook idea of V1 as a bank of Gabor filters. Our findings are a first step towards a powerful new tool to study the nonlinear computations in V1.

1 INTRODUCTION

The mammalian retina processes image information using a number of distinct parallel channels consisting of functionally, anatomically, and transcriptomically defined distinct cell types. In the mouse, there are 14 types of bipolar cells (Euler et al., 2014), which provide input to 30–50 types of ganglion cells (Baden et al., 2016; Sanes & Masland, 2015). In visual cortex, in contrast, it is currently unknown whether excitatory neurons are similarly organized into functionally distinct cell types. A functional classification of V1 neurons would greatly facilitate understanding its computations just like it has for the retina, because we could focus our efforts on identifying the function of a small number of cell types instead of characterizing thousands of anonymous neurons.

Recent work proposed a framework for learning functional cell types from data in an unsupervised fashion while optimizing predictive performance of a model that employs a common feature space shared among many neurons (Klindt et al., 2017). The key insight in this work is that all neurons that perform the same computation but have their receptive fields at different locations, can be represented by a feature map in a convolutional network. Unfortunately, this approach cannot be applied directly to neocortical areas. Neurons in area V1 extract local oriented features such as edges at different orientations, and most image features can appear at arbitrary orientations – just like they can appear at arbitrary locations. Thus, to define functional cell types in V1, we should treat orientation as a nuisance parameter (like receptive field location) and learn features independent of orientation.

To make progress on this front and characterize V1 computation independent of orientation, we extended the work of Klindt and colleagues (Klindt et al., 2017) to use a rotation-equivariant convolutional neural network, which models the responses of 6000 mouse V1 neurons with a shared feature space, whose features are independent of orientation. We show that this model outperforms state-of-the-art CNNs for system identification and allows predicting V1 responses of thousands of neurons with only 16 learned features. Moreover, we show that many of these features do not resemble simple Gabor filters, suggesting that the classical model of V1 as a Gabor filter bank may not capture the true selectivity of neurons under natural stimulation conditions.

2 RELATED WORK

Functional classification of cell types. Characterizing neurons according to some identified, potentially nonlinear response properties has a long history. For instance, researchers have identified simple and complex cells (Hubel & Wiesel, 1968), pattern- and component-selective cells (Gizzi et al., 1990), end-stopping cells (Schiller et al., 1976), gain control mechanisms (Carandini et al., 1997) and many more. However, these properties have been identified using simple stimuli and the models that have been built for them do not apply to natural stimuli in a straightforward way. In addition, most of these studies were done using single-cell electrophysiology, which does not sample cells in an unbiased fashion. Thus, we currently do not know how important certain known features of V1 computation are under natural stimulation conditions and what fraction of neurons express them. Recent work using two-photon imaging in the monkey (Tang et al., 2018) started closing this gap by systematically investigating response properties of V1 neurons to an array of complex stimuli and found, for instance, that about half of the neurons in V1 are selective to complex patterns such as curvature, corners and junctions rather than just oriented line segments.

Recent work in the retina showed that different types of Ganglion cells (Baden et al., 2016) – and to some extend also bipolar cells (Franke et al., 2017) – can be identified based on functional properties. Thus, in the retina there is a relatively clear correspondence of anatomical and genetic cell types and their functional output, and we are getting closer to understanding each retinal cell type's function.

Learning shared feature spaces for neural populations. Antolík and colleagues pioneered the idea of learning a shared nonlinear feature representation for a large population of neurons (Antolík et al., 2016), which others have also used in both retina and V1 (Batty et al., 2016; Kindel et al., 2017; Cadena et al., 2017). Klindt et al. (2017) proposed a framework to learn functional cell types in an unsupervised fashion as a by-product of performing system identification. They propose a structurally constrained readout layer, which provides a compact characterization of each neuron's computation. By enforcing this representation to be sparse, they suggest that the learned features may correspond to distinct functional cell types. Our present work builds upon this idea and extends it.

Rotation equivariance in CNNs. There is a large body of work on equivariant representations (Nordberg & Granlund, 1996). Here we review only the most closely related approches. Cohen & Welling (2016) introduce group-equivariant CNNs, which include rotation equivariance as a special case and form the basis of our approach. Weiler et al. (2017) use a steerable filter basis to learn rotation-equivariant CNNs. We essentially use the same approach, but with a different set of steerable filters (2d Hermite functions instead of circular harmonics). RotEqNet (Marcos et al., 2016) is related to our work in the sense that it also applies multiple rotated versions of each filter to the input. However, instead of maintaining all resulting feature maps as inputs for the next layer, they apply an orientation pooling operation, which reduces each set of feature maps to a two-dimensional vector field. Harmonic networks (Worrall et al., 2016) are an alternative approach that achieves full 360° rotation equivariance by limiting the structure of the convolutional filters. Finally, Dieleman et al. (2016) achieve rotation equivariance by feeding multiple rotated versions of the input image into parallel CNN streams whose weights are tied together.

3 ROTATION-EQUIVARIANT CNN

Network architecture. Our architecture follows that of Klindt et al. (2017). The image is first processed by a multi-layer convolutional core shared by all neurons (Fig. 1A, blue), which we describe in detail below. The resulting feature representation is then turned into a response prediction for each neuron by applying a sparse linear combination across space, followed by a sparse linear combination across features and a pointwise, soft-thresholding nonlinearity (Fig. 1A, red).

This model has a number of desirable properties in terms of data efficiency and interpretability. The separation into convolutional core and readout, together with the strong structural constraints on the readout pushes all the 'heavy lifting' into the core while the readout weights (spatial mask and feature weights) provide a relatively low-dimensional characterization of each neuron's function. Because the core is shared among all neurons (here: thousands), many of which implement similar functions, we can learn complex non-linear functions very accurately.

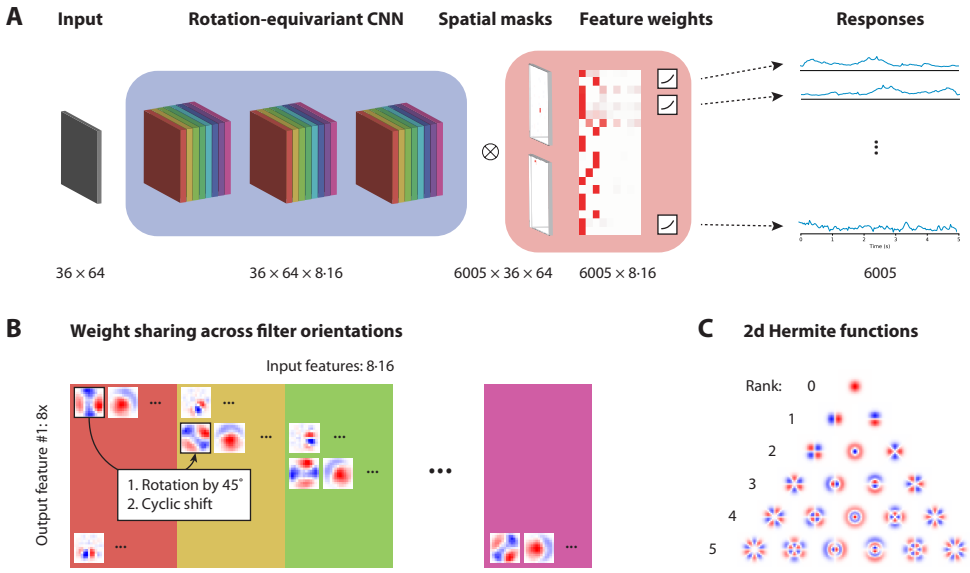

Figure 1: Rotation-equivariant CNN architecture. **A.** The network consists of a rotation-equivariant convolutional core common for all neurons (blue box) and a neuron-specific readout [red box (Klindt et al., 2017)]. Inputs are static images from ImageNet. Prediction targets are responses of 6005 V1 neurons to these images. Rotation equivariance is achieved by using weight sharing across filter orientations. Therefore, eight rotated versions of each filter exist, resulting in eight groups of feature maps (depicted by rainbow colored boxes). **B.** Illustration of weight sharing across orientations for the second and third layer. The previous layer's output consists of eight groups of feature maps (one for each rotation angle). The output is generated by applying a learned filter to each of the input feature maps (here 8×16 kernels shown in the first row). Rotated versions (2nd and following rows) are generated by rotating each kernel and cyclically permuting the feature maps. **C.** Filters are represented in a steerable basis (2d Hermite functions). Functions up to rank 5 are shown here.

Equivariance. A function $f : \mathcal{X} \to \mathcal{Y}$ is called equivariant with respect to a group of transformations Π if such transformations of the input lead to predictable transformations of the output. Formally, for every $\pi \in \Pi$ there is a transformation $\psi \in \Psi$ such that for every $x \in \mathcal{X}$

$$\psi[f(x)] = f[\pi(x)]. \tag{1}$$

CNNs are shift-equivariant by construction, meaning that every translation of the image leads to a matching translation in the feature maps (i. e. π and ψ are both translation by a number of pixels horizontally and vertically). Shift equivariance is a useful property for neural system identification, because it allows us to represent many neurons that perform similar computations but in different locations in space by a single convolutional feature map instead of learning each neurons' nonlinear input-output function 'from scratch.'

Rotation-equivariant CNN. Neurons in V1 do not only perform similar functions at different locations, but also extract similar features with different orientations. Thus, for modeling populations of V1 neurons, learning a representation that is equivariant to rotation in addition to translation would be desirable. To achieve rotation equivariance, we use group convolutions (Cohen & Welling, 2016). Here we use the group of all rotations by multiples of $45°$. That is, for each convolutional filter in the first layer, we have eight rotated copies, each of which produces one feature map (Fig. 1A, blue). Thus, if we learn 16 different filters in the first layer, it will have a total of $8 \times 16 = 128$ feature maps. Formally, if π is a rotation of the image by, say, $45°$, then ψ is a rotation of the feature maps by also $45°$ combined with a cyclic permutation of the feature maps by one rotation step.

For the second (and all subsequent) layers, the procedure becomes a bit more involved. Sticking with the numbers from above, for every feature in the second layer we now learn 128 filters – one for each input feature map (Fig. 1B, first row). To preserve rotation equivariance, we need to create all rotated

copies of these filters and cyclically permute the feature maps such that each rotated filter receives the rotated version of the input (depicted by the colored boxed in Fig. 1B, second and following rows).

To implement weight sharing across filter orientation without aliasing artifacts, we represent the filters in a steerable basis (Weiler et al., 2017). We use the two-dimensional Hermite functions in polar coordinates, which form a steerable, orthonormal basis (Fig. 1C; see also Victor et al., 2006; Hu & Victor, 2016). For filters of size k, we use all 2d Hermite functions up to rank k, which means we have $k(k+1)/2$ basis functions. We sample each filter at twice the resolution and then downsample by 2×2 average pooling to reduce aliasing.

4 EXPERIMENTS

Neural data. We recorded the responses of 6005 excitatory neurons in primary visual cortex (layers 2/3 and 4) from one mouse by taking two consecutive scans with a large-field-of-view two-photon mesoscope (Sofroniew et al., 2016). We selected cells based on a classifier for somata on the segmented cell masks and deconvolved their fluorescence traces (Pnevmatikakis et al., 2016). We did not filter cells according to visual responsiveness. The aquisition frame rate was 4.8 Hz in both scans. We monitored pupil position, dilation, and absolute running speed of the animal. However, because eye movements were rare and we are primarily interested in the average response of neurons given the visual stimulus, we did not further take into account eye position and running speed.

Visual stimuli. Stimuli consisted of 5500 images taken from ImageNet (Russakovsky et al., 2015), cropped to fit a 16:9 monitor, and converted to gray-scale. The screen was 55×31 cm at a distance of 15 cm, covering roughly $120^\circ \times 90^\circ$. In each scan, we showed 5400 of these images once (training and validation set) and the remaining 100 images 20 times each (test set). Each image was presented for 500ms followed by a blank screen lasting between 500ms and 1s. For each neuron, we extract the accumulated activity between 50ms and 550ms after stimulus onset using a Hamming window.

Tools. We performed our model fitting and analyses using DataJoint (Yatsenko et al., 2015), Numpy/Scipy (Walt et al., 2011), Matplotlib (Hunter, 2007), Seaborn (Waskom et al., 2017), Jupyter (Kluyver et al., 2016), Tensorflow (Abadi et al., 2015), and Docker (Merkel, 2014).

Preprocessing. We rescaled the images to 64×36 pixels and standardized them by subtracting the mean over all images in the training set and all pixels from each image and dividing by the standard deviation (also taken over all images in the training set and all pixels). We divided the responses of each neuron by its standard deviation over time. We did not center the neural responses, because they are non-negative after deconvolution and zero has a clear meaning.

Model fitting and evaluation. We initialized all weights randomly from a truncated normal distribution with mean zero and standard deviation 0.01. The biases of the batch normalization layers were initially set to zero. In contrast, we set the biases in each neuron's readout to a non-zero initial value, since the neural responses are not centered on zero. We initialized these biases such that the initial model prediction was on average half the average response of each neuron. To fit the models, we used the Adam Optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.002, a single learning rate decay and early stopping. We monitored the validation loss every 50 iterations and decreased the learning rate once by a factor of 10 when the validation loss had not decreased for five validation steps in a row. We then further optimized the model until the same criterion is reached again.

To evaluate the models, we use the following procedure. For each neuron we compute the Pearson correlation coefficient between the model prediction and the average response over the 20 repetitions of the 100 test images. We then average the correlations over all neurons. This approach tells us how well the model predicts the average response of neurons to a given stimulus, ignoring trial-to-trial variability (which is interesting in itself, but not the focus of the present work).

Architecture details. Our architecture consists of three convolutional layers with filter sizes of 13, 5 and 5 pixels, respectively. Thus, the receptive fields of the CNN's last layer's units were 21 px, corresponding to $\sim 60^\circ$ and covering both classical and extra-classical receptive field. We use zero padding such that the feature maps maintain the same size across layers. We use 16 filter sets (i. e. 128

feature maps) in the first two layers and 8 to 48 filter sets in the third layer (number cross-validated, see below). After each layer, we apply batch normalization followed by a learned scale and bias term for each feature map.[1] After the first and second layer, but not after the third, we apply a soft-thresholding nonlinearity $f(x) = \log(1 + \exp(x))$. The feature maps of the third layer then provide the input for each neuron's readout, which consists of a linear combination first over space and then over features, followed by an added bias term and a final soft-thresholding nonlinearity. Thus, each neuron implements a cascade of three LN operations.

Regularization. We use the same three regularizers as (Klindt et al., 2017). For smoothness of convolution kernels we set the relative weight of the first layer to twice as strong as in the second and third layer to account for the larger kernels. We apply group sparsity to the convolution kernels of the second and third layer. We regularize the spatial masks and feature weights in the readout to be sparse by applying the L1 penalty on the 3d tensor that results from taking their outer tensor product. The weights of all three regularizers are cross-validated as described in the next paragraph.

Model selection. We cross-validated over the number of filter sets in the third layer, ranging from 8 to 48 (i. e. 64 to 384 feature maps[2]) and the strength of the regularizers. For each architecture, we fit 32 models with different initial conditions and randomly drawn hyperparameters (smoothness of filters: 0.001–0.03, group sparsity of filters: 0.001–0.1, sparsity of readout: 0.005–0.03) and chose the best one according to its loss on the validation set (i. e. not using the test set). For all baseline and control models (see below), we also cross-validated over 32 randomly sampled sets of hyperparameters drawn from the same range of values.

Baseline and control experiments. As a baseline, we fit a number of regular CNNs without rotation equivariance. These models are completely identical in terms of number of layers, filter sizes, readout layer and fitting procedure, except for the weight sharing constraint across orientation. We fit models with the same number of feature maps as their rotation-equivariant counterparts as well as smaller models with fewer feature maps (see Table 1).

Previous work has enforced the feature weights in the readout to be positive (Klindt et al., 2017). Because we do not enforce such constraint in the present work, we ran a control experiment, in which we enforce the readout weights (both masks and feature weights) to be positive.

Sparsity of the spatial masks is well justified, because we know that receptive fields are localized. However, it is unclear whether sparsity on the feature weights is desirable, since the function each neuron implements may not be well aligned with the coordinate axes spanned by the small number of features we learn. Thus, we also fit a model without the sparsity regularizer for the feature weights.

5 Results

Architecture search and model selection. We start by evaluating the predictive performance of our rotation-equivariant CNN on a dataset of 6005 neurons from mouse primary visual cortex. First, we performed an initial architecture search, drawing inspiration from earlier work (Klindt et al., 2017) and exploring different numbers of layers and feature maps and different filter sizes. We settled on an architecture with three convolutional layers with filter sizes of 13, 5 and 5 pixels, respectively, and used 16 filter sets (i. e. 128 feature maps) in the first two layers. For the third layer, which provides the nonlinear feature space from which all neurons pool linearly, we cross-validated over the number of features, ranging from 8 to 48 (i. e. 64 to 384 feature maps).

Performance was comparable to earlier studies modeling V1 responses with similar stimuli (Klindt et al., 2017; Antolík et al., 2016). We achieved a correlation on the test set of 0.47. The performance improved with the number of features in the third layer, but the improvement was small beyond 16 features (Fig. 2). For the following analyses we therefore focus on this model with 16 features as a compromise between model complexity and performance.

[1] In our experiments the network did not implement exact rotation equivariance, because batch normalization was applied to each feature map individually instead of jointly to all rotated versions. We therefore re-ran a subset of experiments where we corrected this issue and verified that model performance was indistinguishable.

[2] 384 feature maps was the maximum we could fit into 16 GB of available GPU memory.

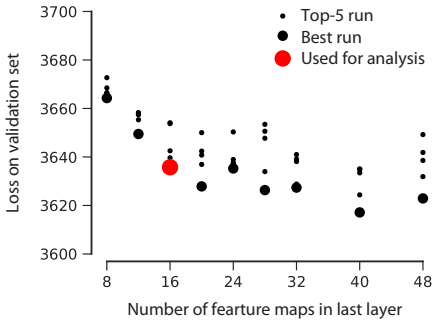

Figure 2: Model comparison via loss on the validation set. Large dots: model with lowest loss on validation set; small dots: top five model fits according to loss on validation set. Red dot: model used for subsequent analyses. 16 feature maps provides a good trade-off between model complexity and predictive performance.

Model	Corr.	S.D.
Rotation-equivariant CNN		
— 16x8–16x8–16x8	**0.47**	**0.005**
— positive feature weights	0.43	0.006
— no feature sparsity	0.27	0.011
CNN (Klindt et al., 2017)		
— 32–32–32	0.39	0.009
— 64–64–64	0.39	0.006
— 128–128–128	0.42	0.008
— 128–128–256	0.41	0.009

Table 1: Average correlation on test set of our rotation-equivariant CNN, two controls, and several regular CNNs as baselines. The three numbers for the CNNs are the number of feature maps in each layer; other parameters are identical to the rotation-equivariant CNN. Standard deviations are across the top-five models of each type.

It is encouraging that a model with only 16 features is sufficient to accurately model a population as large as 6000 neurons. Earlier work modeling V1 responses used a shared feature space of 10–20 (Antolík et al., 2016) or 48 features (Klindt et al., 2017) for populations of 50–100 neurons, which reduced the dimensionality of the feature space by a factor of 2–5 compared to the number of neurons. Here, we reduce the dimensionality by a factor of 375 while achieving a similar level of performance.

Rotation-equivariant CNN outperforms regular CNN. To compare our new model to existing approaches, we also fit a regular CNN with identical architecture except for the weight sharing constraint across orientations. In addition, we fit a number of smaller CNNs with fewer feature maps in each layer, which are more similar in terms of number of parameters, but potentially have less expressive power. Our rotation-equivariant CNN outperforms all baselines (Table 1) and generally requires less data (Fig. 3), showing that enforcing weight sharing across orientations is not only potentially useful for interpreting the model (as we show below), but also serves as a good regularizer to fit a larger, more expressive model.

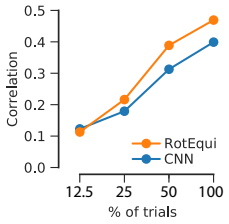

Figure 3: Performance vs. training data.

Feature space generalizes to unseen neurons. To show that our network learns common features of V1 neurons, we performed cross-validation on a held-out set of neurons. Specifically, we excluded 20% of the neurons when fitting the network. We then fixed the rotation-equivariant convolutional core and trained only the readout (spatial mask and feature weights) for those remaining 20% of the neurons. The resulting test correlation for these neurons was indistinguishable (0.46 vs. 0.47) from the full model, showing that the learned features transfer to neurons not used to train these features.

Feature weights are sparse. The intuition behind the sparse, factorized readout layer is that the spatial mask encodes the receptive field location of each neuron while the feature weights parameterize the neuron's nonlinear computation. We now ask how the neurons are organized within this 16-dimensional function space. On the one extreme of the spectrum, each neuron could pick a random direction, in which case sparsity would not be the right prior and the feature weights should be dense. On the other end of the spectrum, there could be a discrete number of functional cell types. In this case, each cell type would be represented by a single feature and the feature weights should be maximally sparse, i. e. one-hot encodings of the cell type.

To analyze the feature weights, we first marginalize them over orientation. To do so, we take the sum of squares over all 8 orientations for each of the 16 features and then normalize them such that the energy of the 16 features sums to one. We find that the feature weights are indeed quite sparse (Fig. 4A): most of the neurons use only 1–4 features (Fig. 4B) and the strongest feature captures more than 50% of the energy for 63% of the neurons.

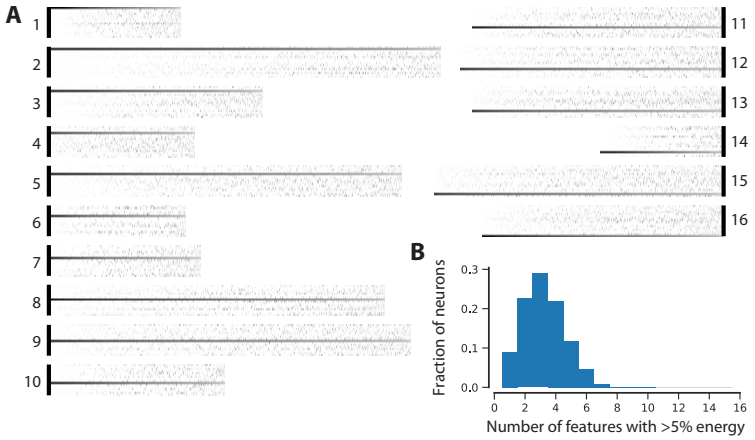

Figure 4: Feature weights are sparse. **A.** We group the neurons into 16 groups by their strongest feature weight. Wihtin each group, we sort the neurons by the sparsity of their weights (in descending order). Within each block labeled 1–16, each row is one feature and each column is one neuron. **B.** Distribution of 'active' features over neurons.

To ensure that the sparsity of the feature weights is a property of the data and not a trivial result of our L1 penalty, we performed an ablation study, where we fit a model that applies sparsity only on the spatial masks and leaves the feature weights unregularized. This model performed substantially worse than the original model (Table 1), suggesting that sparsity is indeed a justified assumption.

Feature weights have consistent sign. There is a second line of evidence that the features learned by our model are meaningful beyond just providing an arbitrary basis of a 16-dimensional feature space in which neurons are placed at random: the weights that different neurons assign to any given feature have remarkably consistent signs (Fig. 5). For 11 of 16 features, more than 90% of the non-zero weights have the same sign.[3] Thus, the negation of one feature that drives a large group of neurons is not a feature that drives many neurons.

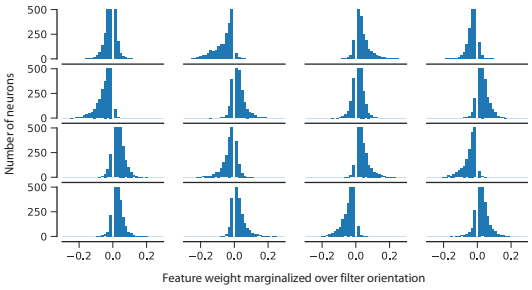

Figure 5: Feature weights have consistent sign across neurons. Note that the majority of weights is zero; the central bin has been excluded for clarity.

Visualization of the different 'cell classes.' Having presented evidence that the features identified by our model do indeed represent a simple and compact, yet accurate description of a large population of neurons, we now ask what these features are. As a first step, we simply assume that each of the 16 features corresponds to one functional cell type and classify all neurons into one of these types based on their strongest feature weight. This approach is obviously to some extent arbitrary. By no means to we want to argue that there are exactly 16 types of excitatory neurons in V1 or that excitatory neurons in V1 can even be classified into distinct functional types. Nevertheless, we believe that the 16 groups defined in this way are useful in a practical sense, because they represent features that cover more than 50% of the variance of a large fraction of neurons and therefore yield a very compact description of the most important features of V1 computation across the entire excitatory neuronal population.

To visualize what each of these features compute, we pick the 16 most representative examples[4] and compute their preferred stimulus by activity maximization (Erhan et al., 2009). That is, starting from

[3]Note that the sign itself does not carry any meaning. Because there is no nonlinearity after the last convolution, we can flip the sign of all filters generating this feature and at the same time flip all neurons' readout weights for this feature, which would leave the model prediction unchanged.

[4]The top-16 neurons with the highest energy on the given feature for which the model achieves at least a correlation of 0.2 on the validation set.

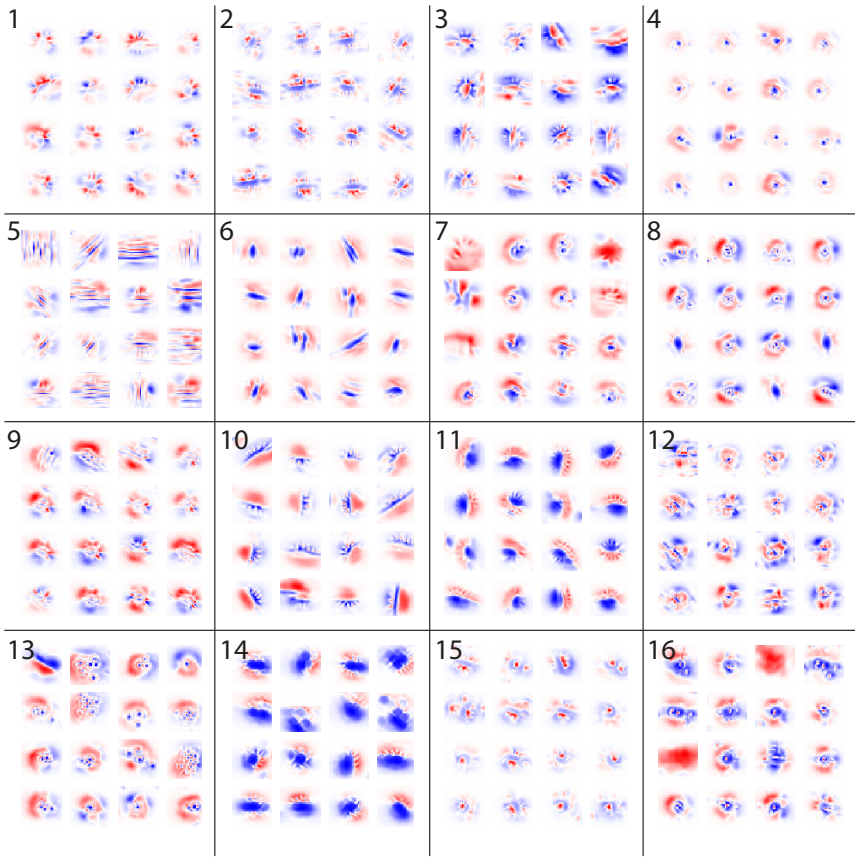

Figure 6: Preferred stimuli for all 16 classes, obtained by activity maximization. For each class, we show the 16 neurons with the sparsest feature weights that have an average correlation of the model prediction with the neural response of at least 0.2 on the validation set.

a white noise image, we iteratively optimize the image via gradient descent to maximize the activity of the neuron while keeping the norm of the image constant. To stabilize the optimization procedure, we use gradient preconditioning (Olah et al., 2017) assuming a $1/\sqrt{f}$ spectrum and regularize the images to be smooth using an L2 penalty on the Laplace-filtered version of the image.

The resulting preferred stimuli of each functional type are shown in Fig. 6. The first and most important point of this plot is that it shows that the premise of rotation equivariance holds: cells are clustered according to similar spatial patterns, but independent of their preferred orientation. The second interesting aspect is that many of the resulting preferred stimuli do not look typical standard textbook V1 neurons which are Gabor filters. While there are some typical Gabor-shaped images (e. g. types 2, 6, 11 and 14), many types feature a prominent center-surround shape, often with an asymmetric, crescent-shaped surround (e. g. types 4, 7, 8, 13, 15, 16).

6 Discussion

We developed a rotation-equivariant convolutional neural network model of V1 that allows us to characterize and study V1 computation independent of orientation preference. Enforcing weight sharing not only in the spatial domain, but also across orientations, allows us to fit larger, more expressive models. While our work lays out the technical foundation, we only scratched the surface of the many biological questions that can now be addressed. Future work will have to investigate to what extent learned features generalize across recording sessions and animals, whether they are consistent across changes in the architecture and – most importantly – whether the functional organization inferred with the methods presented here find any resemblance in anatomical or genetic properties (Tasic et al., 2017) of the neurons recorded.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. URL `http://download.tensorflow.org/paper/whitepaper2015.pdf`.

J. Antolík, S. B. Hofer, J. A. Bednar, and T. D. Mrsic-flogel. Model Constrained by Visual Hierarchy Improves Prediction of Neural Responses to Natural Scenes. *PLoS Comput Biol*, pp. 1–22, 2016. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004927.

Tom Baden, Philipp Berens, Katrin Franke, Miroslav Román Rosón, Matthias Bethge, and Thomas Euler. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586):345, 2016.

E. Batty, J. Merel, N. Brackbill, A. Heitman, A. Sher, A. Litke, E. J. Chichilnisky, and L. Paninski. Multilayer network models of primate retinal ganglion cells. Number Nips, 2016.

S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *bioRxiv*, 2017. doi: 10.1101/201764. URL `https://www.biorxiv.org/content/early/2017/10/11/201764`.

Matteo Carandini, David J Heeger, and J Anthony Movshon. Linearity and normalization in simple cells of the macaque primary visual cortex. *Journal of Neuroscience*, 17(21):8621–8644, 1997.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Bernoulli*, (1341):1–13, 2009. URL `http://igva2012.wikispaces.asu.edu/file/view/Erhan+2009+Visualizing+higher+layer+features+of+a+deep+network.pdf`.

Thomas Euler, Silke Haverkamp, Timm Schubert, and Tom Baden. Retinal bipolar cells: elementary building blocks of vision. *Nature Reviews Neuroscience*, 15(8):507, 2014.

Katrin Franke, Philipp Berens, Timm Schubert, Matthias Bethge, Thomas Euler, and Tom Baden. Inhibition decorrelates visual feature representations in the inner retina. *Nature*, 542(7642): 439–444, 2017.

Martin S Gizzi, Ephraim Katz, Robert A Schumer, and J Anthony Movshon. Selectivity for orientation and direction of motion of single neurons in cat striate and extrastriate visual cortex. *Journal of neurophysiology*, 63(6):1529–1543, 1990.

Qin Hu and Jonathan D Victor. Two-dimensional hermite filters simplify the description of high-order statistics of natural images. *Symmetry*, 8(9):98, 2016.

David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3): 90–95, 2007.

W. F. Kindel, E. D. Christensen, and J. Zylberberg. Using deep learning to reveal the neural code for images in primary visual cortex. Technical report, 2017. URL `http://arxiv.org/abs/1706.06208`.

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

D. Klindt, A. S. Ecker, T. Euler, and M. Bethge. Neural system identification for large populations separating "what" and "where". In *Advances in Neural Information Processing Systems 30*, Sep 2017. in press.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90. IOS Press, 2016.

Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. *arXiv preprint arXiv:1612.09346*, 2016.

Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014. ISSN 1075-3583. URL `http://dl.acm.org/citation.cfm?id=2600239.2600241`.

Klas Nordberg and Gösta Granlund. Equivariance and invariance-an approach based on Lie groups. In *Image Processing, 1996. Proceedings., International Conference On*, volume 3, pp. 181–184. IEEE, 1996.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

E. A. Pnevmatikakis, Y. Gao, D. Soudry, D. Pfau, C. Lacefield, K. Pskanzer, R. Bruno, R. Yuste, and L. Paninski. Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data Eftychios. *Neuron*, 89:285–299, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Joshua R Sanes and Richard H Masland. The types of retinal ganglion cells: current status and implications for neuronal classification. *Annual review of neuroscience*, 38:221–246, 2015.

Peter H Schiller, Barbara L Finlay, and Susan F Volman. Quantitative studies of single-cell properties in monkey striate cortex. i. spatiotemporal organization of receptive fields. *Journal of neurophysiology*, 39(6):1288–1319, 1976.

N. J. Sofroniew, D. Flickinger, J. King, and K. Svoboda. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife*, 5(JUN2016):1–20, 2016. ISSN 2050084X. doi: 10.7554/eLife.14472.

Shiming Tang, Tai Sing Lee, Ming Li, Yimeng Zhang, Yue Xu, Fang Liu, Benjamin Teo, and Hongfei Jiang. Complex Pattern Selectivity in Macaque Primary Visual Cortex Revealed by Large-Scale Two-Photon Imaging. *Current Biology*, 28(1):38–48.e3, 2018.

Bosiljka Tasic, Zizhen Yao, Kimberly A. Smith, Lucas Graybuck, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy A. Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Seana Parry, Nadiya V. Shapovalova, Daniel Hirchstein, Julie Pendergraft, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Boaz P. Levi, Susan M. Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. Shared and distinct transcriptomic cell types across neocortical areas. *bioRxiv*, pp. 229542, December 2017.

Jonathan D. Victor, Ferenc Mechler, Michael A. Repucci, Keith P. Purpura, and Tatyana Sharpee. Responses of V1 Neurons to Two-Dimensional Hermite Functions. *J Neurophysiol*, 95(1):379–400, 2006.

Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. mwaskom/seaborn: v0.8.1 (september 2017), September 2017. URL https://doi.org/10.5281/zenodo.883859.

Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. *arXiv preprint arXiv:1711.07289*, 2017.

D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic Networks: Deep Translation and Rotation Equivariance. *ArXiv e-prints*, December 2016.

D. Yatsenko, J. Reimer, A. S. Ecker, E. Y. Walker, F. Sinz, P. Berens, A. Hoenselaar, R. J. Cotton, A. S. Siapas, and A. S. Tolias. DataJoint: managing big scientific data using MATLAB or Python. Technical report, 2015. URL http://biorxiv.org/lookup/doi/10.1101/031658.