# Applying 3D Convolutional Neural Networks to Psychophysics

Daniel Birman
Stanford University
dbirman@stanford.edu

Dylan Cable
Stanford University
dcable@stanford.edu

Steeve Laquitaine
Stanford University
steeve@stanford.edu

## Introduction

Despite the radical simplicity of convolutional neural networks some researchers have found direct correlates between network layer properties and actual neuron responses. In model systems such as the macaque [1] there are clear analogies between the early visual cortex layers (V1-V4) and the properties of a trained convolutional network. We plan to explore this interesting dynamic in the context of motion. Our goal is to modernize an older model of the visual stream dedicated to motion [2]. The original model explicitly coded the features selected for at each layer, based on the known anatomical properties of macaque V1 and MT. In contrast, we plan to build a generic convolutional neural net architecture, which will be trained to discriminate examples of motion. Because our architecture is more generic it will fail to precisely model the known anatomy of V1. This leaves open the possibility that during training the network will 'learn' similar features, such as simple and complex cell receptive fields. Our goal in building this model is to develop a model system within which we can test other interesting questions, such as whether a convolutional architecture will develop similar behavioral asymmetries to the actual human (and monkey) visual systems.

## Problem Statement

### 1.1 Data

The motion network (MotionNet) that we are designing has an advantage over other convolutional neural networks in that we can generate arbitrary amounts of training data. Psychophysics tasks usually involve random dot displays, consisting of white dots on a uniform black background, or white and black dots on a grey background. We have put together basic functions to generate an unlimited number of random dot displays involving translational (flat), optic flow, expansion/contraction, and rotational motion stimuli. All of these stimuli can have arbitrary contrast (e.g. Michelson contrast [3], difference between luminance intensities in the dots and background, compared to the total luminance), speed, number of dots, etc. This parameterization allows us to generate an unlimited set of training examples on the fly—as well as allowing us to easily manipulate our datasets, for example to better match a psychophysics dataset from a specific experiment. One advantage of having tools to generate online data is that we can build batches without relying on a heavy memory load, allowing us to train our dataset on an arbitrary number of examples with no overhead for loading and saving data. This also allows us to generate far more variability in our input dataset, similar to jittering of image data.

### 1.3 Expected Results & Evaluation

Our first level of evaluation is simply performance on the training, validation, and withheld test sets of data. These summary statistics of our trained model give us an estimate of how well the model is trained to do visual motion discrimination, within the context of psychophysics stimuli. These statistics also give us a sense of whether our convolutional net architecture will be appropriate for the scope of the problem we are putting it up against. As we iterate our design we plan to use the test-set performance as a measure of our architecture success and as a validation of our choice of hyper parameters. But our goal in building this model system is to learn about how convolutional networks might be informative for our understanding of the human and monkey brain. To that end we plan to have two additional layers of evaluation: psychophysics based evaluations (e.g. ability to discriminate untrained features), and neural based evaluations (e.g. similarity to monkey physiology responses). These non-network based measures are a way to evaluate whether or model is effectively meeting its goal of representing the visual system itself. Note again that none of these secondary goals will be built into the model directly; rather we expect that as an emergent property of the layers we will see analogous similarity to the human and monkey visual systems.

Our trained model will be designed to discriminate direction of motion. To do this it will need to be invariant to contrast, speed, and motion type, all of which we expect to be represented within the network to some extent. To test for these representations we plan to pull out the activations at each layer and use a decoding algorithm (SVM) to identify the specific layers and neurons within those layers that represent each of these untrained features. This second evaluation will allow us to identify whether MotionNet is an accurate *computational* representation of the visual system, insofar as it can represent a similar space of features and perform similar computations. Because our network will be relatively easy to train, due to

its fairly small architecture, we plan to test out different training regimes and compare whether a network trained to detect motion performs well at contrast discrimination, and vice versa for each of the feature pairs.

Our model will also be evaluated on whether it has similar response properties to the human and monkey visual systems. We plan to do this by looking for 'response functions' within the architecture. For example, in the human visual system there are monotonic increasing responses to increasing magnitudes of contrast. This is despite the contrast normalization that occurs in retina, so it isn't a measurement of overall luminance in an image. If we see an overall shift in the magnitude of activations in our first few layers due to increasing contrast then we can use this as evidence that our network has an analogous computational representation of contrast (to a minimal extent) to the human early visual cortex. This effect holds true for contrast, speed, and motion coherence (percentage of dots moving together) in the human and monkey visual systems, and we expect to see similar response functions in our architecture.

In sum, we plan to evaluate our model on technical criteria such as the cross-validated performance on the training set, as well as "neural" criteria that are derived from the models ability to capture aspects of human and monkey physiology.

Technical Approach

We plan to model our architecture on a rough outline of how the visual system is organized. To that end we have a layer, or group of layers, corresponding to each cortical and sub-cortical region spanning the retina to MT [Table 1]. Although this is far from a perfect analogy to cortical computations we plan to introduce three constraints that will allow us some anatomical similarity to cortex: (1) we will enforce sparsity between cortical layers with high-dropout rates, (2) we will use spatial batch normalization as a form of 'divisive normalization', (3) we will allow the convolutional layers to be both spatial and temporal convolutions. We plan to use sparsity to force our neurons to depend only on a subset of the neurons in prior layers. Divisive normalization is a property of local sets of neurons in cortex whereby they inhibit each other, part of the brain's constraints that work to limit explosive response rates. Together these three constraints create a bounded space within which the model has flexibility to learn representations. Our hope is that although these constraints are not explicitly anatomically correct, they will nevertheless lead to the MotionNet learning a set of interesting analogous features to the human/monkey visual system.

| Brain Area | Receptive Field Type | Implementation |
|---|---|---|
| Retina | Contrast Normalization | 3x3x1 normalization |
| LGN | Center Surround | 3x3x3 conv + ReLU |
| V1simple | Simple cells: Edge Detectors | Dropout + 3x3x3 conv + ReLU + spatial BN |
| V1complex | Complex cells: edges and shapes | Dropout + 3x3x3 conv + ReLU + spatial BN |
| MT | Motion sensitivity | Dropout + 3x3x3 conv + ReLU |
| LIP + M1 | Readout Layer | Fully-connected affine layer |

Table 1: Overview of MotionNet architecture. Kernel sizes correspond to *row*col*time*.

Preliminary Results

We have built the necessary tools for our experiment, and built and tested a very basic convolutional neural network.

2.1 Data

Our dataset construction is complete, and we have built and evaluated functions for the following four types of motion: translation, rotation, expansion/contraction, and optic flow. The dataset allows for the full range of stimulus types that we anticipate training our model with.

2.2 MotionNet

We are currently building the MotionNet architecture. We are using Keras as a wrapper around Theano, using a forked development version of the Convolution3D library (see https://github.com/MinhazPalasara/keras). Our implementation is in progress as we are still learning how Theano and Keras work, so we do not have a functional network trained on our dataset yet. We have piped our examples through test networks trained using the assignment codes—but due to the importance of taking into account temporal information these networks were not able to learn to categorize motion effectively. Our Keras implementation should be functional within the next few days and then we plan to train several mini-working models and perform hyper parameter search as we work towards training a full model on a much larger training dataset.

References


[1] Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences, 111(23), 8619-8624.

[2] Simoncelli, E. P., & Heeger, D. J. (1998). A model of neuronal responses in visual area MT. Vision research, 38(5), 743-761.

[3] Michelson, A. (1927). *Studies in Optics*. U. of Chicago Press.