Supplemental Material: Learning Inductive Biases with Simple Neural Networks

Reuben Feinman (reuben.feinman@nyu.edu)
Center for Neural Science
New York University

Brenden M. Lake (brenden@nyu.edu)
Department of Psychology and Center for Data Science
New York University

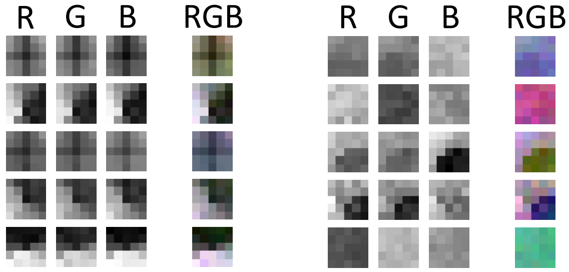
SM 1: Selection of network architectures and training parameters

Architecture. The MLP architecture for Experiment 1 was chosen ad-hoc before running any experiments. The network receives a 60-dimensional input, and thus, we chose a hidden layer size of 30 units to reduce this dimensionality by a factor of 2. L2 regularization was critical to the performance of the network when small training sets were provided. For Experiment 2, we chose the minimal CNN architecture that could effectively learn the image classification task that it was assigned. We generated a large dataset of 30 categories and 20+ examples per category. Then, we started with a large CNN and iteratively reduced the number of parameters until the minimal architecture was found. L2 regularization was again critical to model performance.

Training parameters. For both the MLP and the CNN, we train the network to minimize negative log-likelihood loss, using stochastic gradient descent (SGD) with the RMSprop update rule and a typical batch size of 32. There are a few exceptions to this batch size: when the training set is very small, we adjust the batch size to ensure there are at least 5 training batches. Thus, for a training set with N categories and K examples per category (a total of $N*K$ training points), we use a batch size of min(32, $\frac{N*K}{5}$). The number of training epochs was chosen such that the network loss reaches an asymptote for each the MLP and CNN. Training loss is monitored and used to save the best model.

SM 2: Experiment 2 perceptual sensitivity tests

As in Experiment 1, for Experiment 2 we parametrically manipulate the stimuli to analyze the network's sensitivity to changes along different stimulus dimensions, using a CNN trained with N=30 & K=10. Distance in shape space is quantified as the Modified Hausdorff Distance (Dubuisson & Jain, 1994) between the shape pair. In color space, physical distance is quantified using the cosine similarity of the RGB vector pair. Beginning with an exemplar object stimuli, we sample 50 secondary shapes and order them by their distance from the exemplar. We then modify the shape of the exemplar parametrically by stepping along this list, recording network similarities between the original and modified versions in each case. A mirroring experiment is then performed with color; in each case, only 1 attribute is altered at a time. Results are shown in Fig. 4. As with the MLP, our CNN's selection preferences show a clear parametric dependency on shape, and a much weaker dependency on color



(a) Shape

(b) Color

Figure 1: Visualizing RGB channels of learned first-layer convolution filters. (a) shows the filters of our CNN trained with explicit shape bias training (N=50 & K=18). Each row corresponds to 1 of the 5 filters. The first 3 channels are shown in the 'R', 'G' and 'B' columns, respectively. These 3 channels are shown together in a 4th column, labeled 'RGB'. (b) shows mirroring filters for our CNN trained to label objects with category names based on color. In both (a) and (b), only channels 1-3 of the 4 are shown.

SM 3: Experiment 2 color bias training

For the sake of comparison, in Experiment 2 we also trained our CNN to label objects with names organized by color. Our goal was to compare the required sample complexity for color bias training with that of shape bias training, and to evaluate whether color bias development follows a similar 2-step process. All dataset parameters mirrored those of shape training, except that the object labels were aligned with the color attribute of each training image. Performance on the generalization tests was measured as the fraction of trials for which the network selects the color match. Results for CNN color bias training are shown in Supp. Fig. 2. Notably, the color-trained CNN requires a smaller sample complexity to achieve 0.7 accuracy on the 2^{nd}-order test, reaching a score of 0.73 with N=2 & K=3. Furthermore, this network does not appear to follow the 2-step process of bias development; results for 1^{st}- and 2^{nd}-order generalizations look near-identical to one another. In order to identify a stimulus as a member of a particular color category, the network needs only to find a single pixel of that color, a task that is much simpler than representing and identifying shape. Representing color requires a simple 3D space. By learning to isolate and preserve this space in the hidden layers, the network can easily generalize to novel colors, hence the early 2^{nd}-order results. We inspected the learned representations of both a shape-trained and a color-trained CNN, trained with N=50 & K=18, by visualizing the first-layer convolution filters of each network (Supp. Fig. 1).

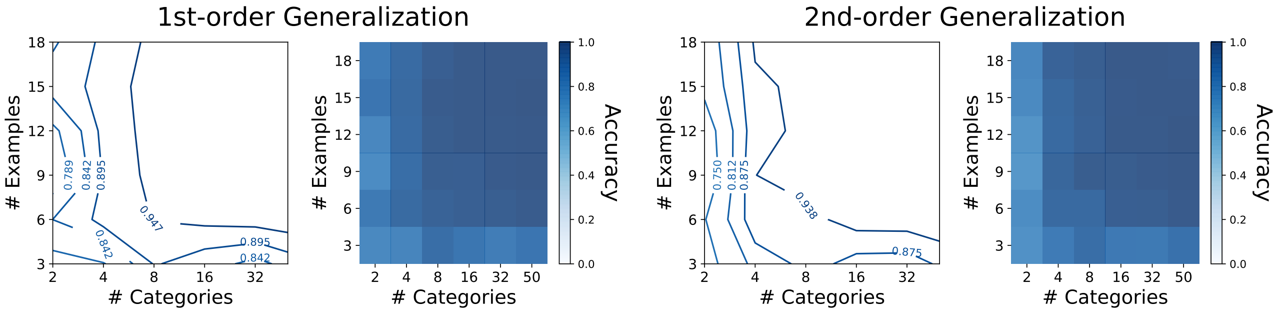


Figure 2: CNN generalization results for color bias training. The network is trained to label objects with category names based on color. In this case, the generalization tests evaluate the fraction of times that the color match is selected. Results in each grid show the average of 10 training runs.

As we would expect, filters of the shape-trained CNN look identical across R, G and B channels, as this network needs no sensitivity to color. In contrast, filters of the color-trained CNN vary across channels, indicating that the network has learned a selectivity for color.

SM 4: Experiment 3 network details

In Experiment 3, we use a slightly modified version of the CNN from Experiment 2 (see Supp. Fig. 3). We train our CNN to simultaneously label the object's name, which correlates with shape, as well as its color and texture names. The CNN thus has 3 softmax layers, each of which extends from the same fully-connected layer, and each of which has its own negative log-likelihood loss function. The training loss is computed as a weighted average of the 3 losses, with weights of 0.6, 0.2 and 0.2 assigned to shape, color and texture, respectively:

$$Loss = 0.6 * shape\_loss + 0.2 * color\_loss + 0.2 * texture\_loss$$

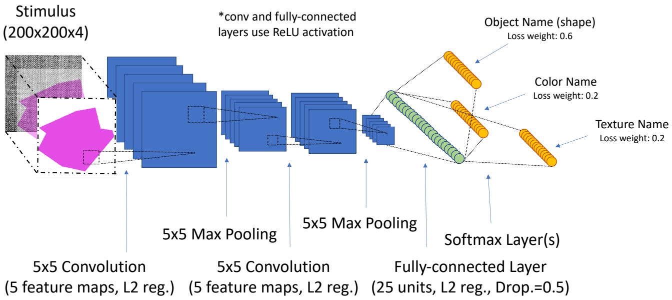


Figure 3: CNN architecture for Experiment 3. The architecture mimics the original CNN of Experiment 2, with the exception of the softmax layer. Here, there are 3 softmax layers (1 for each shape, color and texture), each of which extends from the fully-connected layer.

References

Dubuisson, M., & Jain, A. K. (1994). A modified hausdorff distance for object matching. In *Proceedings of the international conference on pattern recognition* (pp. 566–568).