Learning Inductive Biases with 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

Abstract

People use rich prior knowledge about the world in order to efficiently learn new concepts. These priors–commonly referred to as "inductive biases"–pertain to the space of internal models considered by a learner, and they help maximize the amount of information that is extracted from limited data. Recently, it was shown that performance-optimized deep neural networks (DNNs) develop inductive biases similar to those possessed by human children. However, these models use unrealistic training data, and it remains unclear whether they develop their biases in the same way as humans. We investigate the development of inductive biases in DNNs and perform novel layer-wise and parametric analyses of these biases. TODO... finish.

Keywords: learning-to-learn; neural networks; inductive biases

Introduction

TODO... write introduction

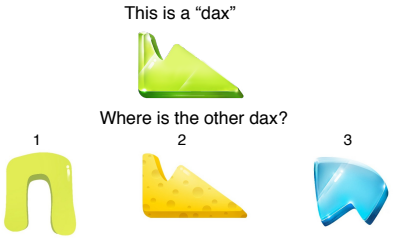


Figure 1: The shape bias. Children learn that objects with the same name tend to have the same shape, and thus option 2 above is likely the right answer. This inductive bias helps with future word learning.

Efficient Word Learning

We first set out to model the infant learning tasks described in (Smith, Jones, Landau, Gershkoff-Stowe, & Samuelson, 2002) using simple neural networks. In order to do so, we use artificial toy data that is designed to mimic the training data described in the paper. Each object sample is assigned a shape, texture and color value. There are two types of model evaluations performed, both drawn from (Smith et al., 2002).

1. First-order generalization test: For the first-order generalization test, infants are asked to evaluate novel instances of familiar objects. To simulate this test, we train our neural network models to classify objects, ensuring that objects of the same category are assigned the same shape. Then, we build a test set by creating one novel exemplar of each category that appeared in the training set. The novel exemplar has the same shape as the training exemplars of that category, but a new color and texture combination. Accuracy is defined as the fraction of test images that are correctly classified by the model. This test is repeated for different training set sizes, i.e. different combinations of $\{\#\ categories, \#\ exemplars\}$. It is important to note that as *# categories* increases, the first-order task becomes more difficult.

2. Second-order generalization test: For the second-order generalization test, infants are presented with an exemplar of a novel object category as a baseline. Then, they are shown 3 comparison objects: one which has the same shape as the baseline, one with the same color, and one with the same texture. In each case, the other 2 features are different from the baseline. The infants are asked to select which of the 3 comparison objects are of the same category as the baseline object. We simulate this test by creating an evaluation set containing groupings of 4 samples: the baseline, the shape constant, the color constant, and the texture constant. Each grouping serves as one test example. We find which of the 3 samples the NN thinks to be most similar by evaluating the cosine similarity using the hidden layer features of the model. Accuracy is defined as the fraction of groupings for which the model chose the correct (shape-similar) object. This test was repeated for different training set sizes, i.e. different combinations of $\{\#\ categories, \#\ exemplars\}$.

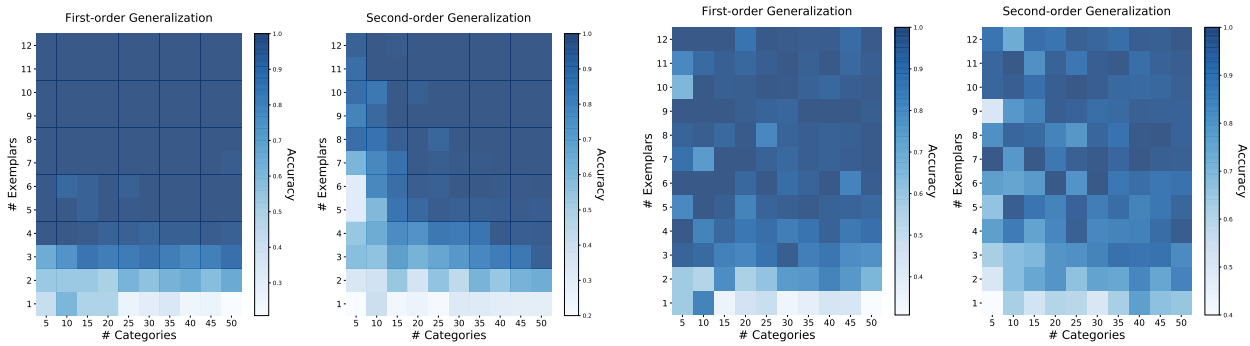
Simple Multilayer Perceptron

To begin with, we use a simple multilayer perceptron (MLP) that operates on categorical data. Since shape, color, & texture have categorical feature values, we encode the values using unique bit vectors that are randomly assigned at the beginning of the experiment. We use a simple feed-forward NN with one hidden layer of 30 units, and the ReLU activation function. The number of units in the softmax layer depends on the *# categories* parameter for the particular dataset. Results are shown in Fig. 2a.

Simple Convolutional Neural Network

As a second type of model, we used a simple convolutional neural network (CNN) architecture consisting of... TODO. To

The source code repository for this paper can be found at `http://github.com/rfeinman/learning-to-learn`



(a) MLP

(b) CNN

Figure 2: First- and second-order generalization results for the simple MLP and CNN models. For each {*# categories*, *# exemplars*} pair, the average result from 5 trials is shown.

train this CNN, we generated images of artificial 2-D objects (Fig. 3) with a variety of different shape, color and texture values. TODO... finish. Results are shown in Fig. 2b.

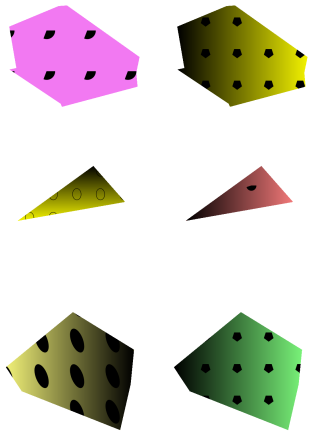


Figure 3: Computer-generated images of 2D objects with different shape, color and texture features.

Figure 4: Artist-designed images of 3D objects with different shape, color and texture features.

Figure 5: CogPsyc images with different shape, color and texture features (TODO).

Extensive Bias Analysis

We next set out to analyze the biases of image classification models in detail. The model that we choose to investigate is the popular VGG-16 network (), which has been pre-trained for the ImageNet classification task. Our analysis consists of two parts: 1) a detailed layer-wise investingation of the shape, color and texture biases, and 2) a parametric analysis of the shape and color biases in these models.

Layer-wise Biases

The first step of our analysis is to evaluate the shape, color and texture biases of VGG-16 at each of its layers, in order to get a picture of how these biases develops along the higherarchy of the model's internal representation. In order to probe the model, we make use of two unique image datasets with stimuli that mimic (Smith et al., 2002).

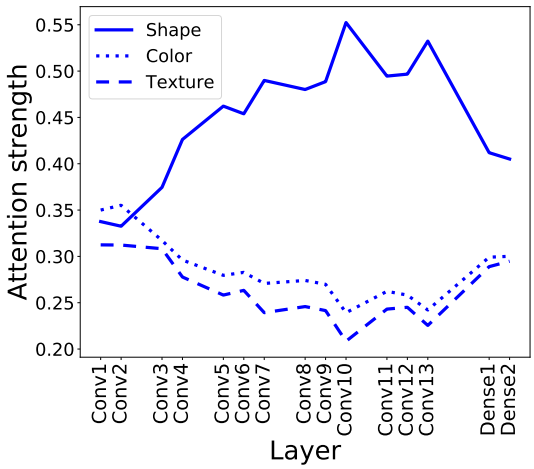
1. Artist-generated object dataset: These images were generated by an artist in Adobe Photoshop. See Fig. 4.

2. CogPsyc object dataset: These images were provided by cognitive psychologist Linda Smith, and they were used in the experiments of (Ritter, Barrett, Santoro, & Botvinick, 2017). See Fig. 5.

The layer-wise bias results are shown in Fig. 6a.

Parametric Biases

The second step of our analysis is to examine how the shape and color biases depend on the intensity of their respective feature similarities. For these tests, we make use of our

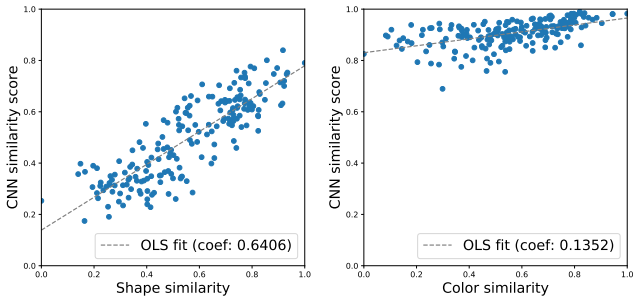


(a) Artist-generated images

(b) CogPsyc images (TODO)

Figure 6: VGG-16 layer-wise biases on two image datasets. Attention strength refers to the network's similarity score between the target objectand objects that match in either shape, color or texture.

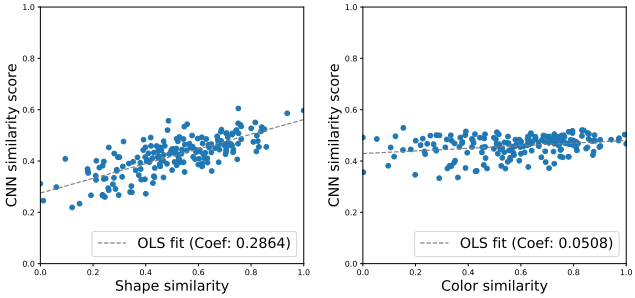
computer-generated images so that we can quantitatively manipulate and evaluate the shape and color features of our objects. The parametric bias results are shown in Fig. 7 and 8.



(a) Shape

(b) Color

Figure 7: VGG-16 parametric shape and color biases w/ other features constant.



(a) Shape

(b) Color

Figure 8: VGG-16 parametric shape and color biases w/ other features varying.

Acknowledgements

This research was supported by a Henry M. McCracken fellowship and NYU start-up faculty funding? We thank Subhankar Ghosh for useful code and preliminary experiment ideas.

References

Ritter, S., Barrett, D. G. T., Santoro, A., & Botvinick, M. M. (2017). Cognitive psychology for deep neural networks: a shape bias case study. In *Proceedings of the 34th international conference on machine learning* (pp. 2940–2949).

Smith, L. B., Jones, S. S., Landau, B., Gershkoff-Stowe, L., & Samuelson, L. (2002). Object name learning provides on-the-job training for attention. *Psychological Science*, *13*(1), 13–19.