BOTTOM-UP OR TOP-DOWN? DYNAMICS OF DEEP REPRESENTATIONS VIA CANONICAL-CORRELATION ANALYSIS

Maithra Raghu,[1] **Jason Yosinski,**[2] **& Jascha Sohl-dickstein**[1]
[1]Google [2]Uber AI Labs
maithra@google.com, yosinski@uber.com, jaschasd@google.com

ABSTRACT

We present a versatile quantitative framework for comparing representations in deep neural networks, based on Canonical Correlation Analysis, and use it to analyze the dynamics of representation learning during the training process of deep networks. We find that layers converge to their final representation from the bottom-up, but that the representations themselves migrate downwards in the network over the course of learning.

1 INTRODUCTION

Understanding representations learned by neural networks is an open area of research. Past approaches have sought insight by comparing representations between multiple networks by learning stitching layers between networks (Lenc & Vedaldi, 2015), by computing per-neuron correlation and mutual information (Li et al., 2016), and by learning linear projections between features and labels for different layers and over the training process (Alain & Bengio, 2016).

Inspired by these methods, we propose a simple approach to compare representations across different layers, across different networks, and between different points in the training process by using Canonical Correlation Analysis (CCA). CCA is a measure of subspace similarity which determines how correlated different subspaces are, modulo affine transformations. Like stitching networks from Lenc & Vedaldi (2015), we look at the set of neuron activations, but in a deterministic manner allowing for faster comparison of more pairs of representations. In contrast to work by Li et al. (2016), we consider the entire layers representation rather than each neuron individually (i.e. we do not require representations be neuron-aligned to be captured).

In this study we propose analysis via the following pipeline:

1. Train a Reference network $\mathsf{Net_R}$ with $\mathsf{L_R}$ layers to completion and compute the representations at each layer, for each example in a dataset.
2. Train a Specimen network $\mathsf{Net_S}$ with $\mathsf{L_S}$ layers to completion and compute representations at each layer, for each example, at N subsampled iterations during training. $\mathsf{Net_S}$ and $\mathsf{Net_R}$ may be identical.
3. Compute the CCA coefficients between each pair of layers and each timestep. This produces a 4-D CCA tensor C with shape $L_R \times L_S \times N \times$ [layer size].
4. Analyze the similarities of representations across layers and timesteps captured by this tensor. Here we plot several derived quantities, and present associated conclusions about the dynamics of learning.

2 MEASURING SIMILARITY BETWEEN LAYER REPRESENTATIONS

Neurons are functions over the dataset. The response h^l_{is} of unit i in layer l to sample index s from a dataset χ fully describes the function computed by that unit on the dataset. If we restrict the domain of the function to be the dataset, i.e. restrict network input to $\mathbf{x} \in \chi$, then the vector $\mathbf{h}^l_i$ corresponds to the function computed by unit i.

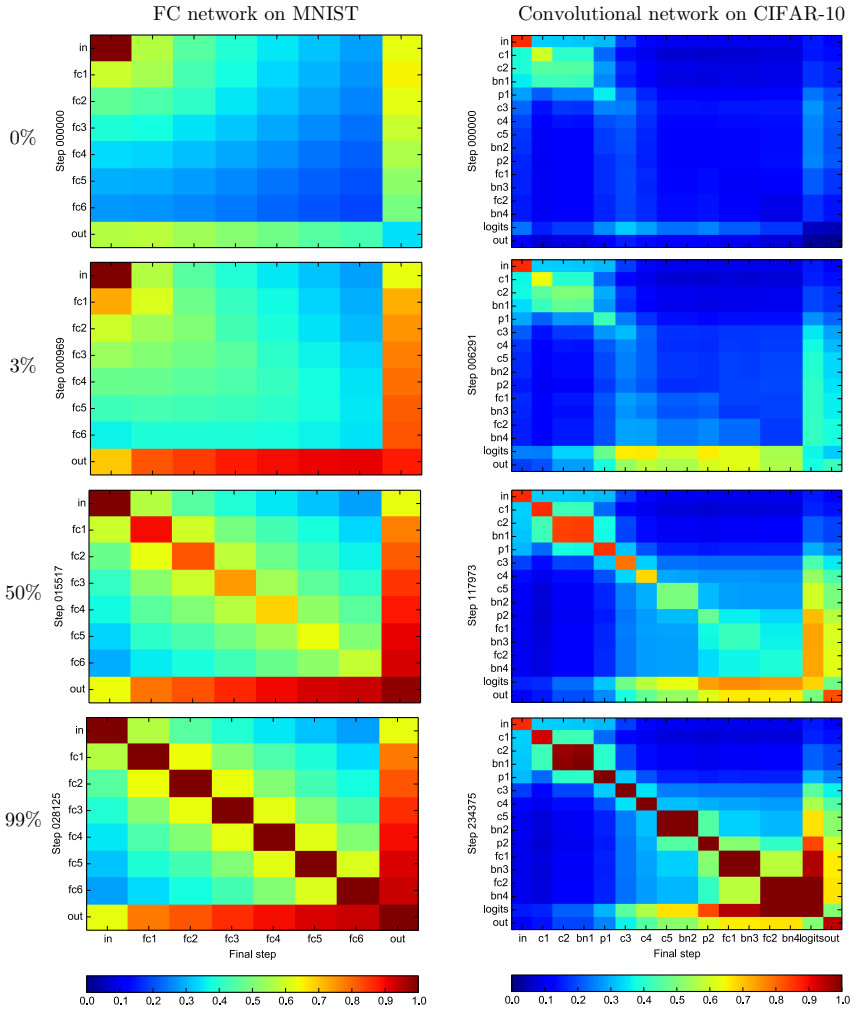

Figure 1: Slices of the 4-D C CCA tensor representing CCA similarities ρ between pairs of layers at different training steps.**Left:** fully connected network trained on MNIST. **Right:** convolutional network trained on CIFAR-10. The four rows of panes correspond (approximately) to 0%, 3%, 50%, and 99% through training. The x-axis in each pane indexes over to the final, converged, representations at the given layer. The y-axis in each pane indexes over the layers during training. For example, in the top-left most plot, the teal square four from the top and one from the left side corresponds to the average cannonical correlation between the representation of the fc3 layer as randomly initialized (0% trained) and the final trained fc1 layer. The data input layer (upper left in each pane) always has similarity 1 with itself because the representation in data-space is fixed, and each layer is always perfectly correlated with its final version (diagonal at 100% training, not shown). **Bottom-up convergence:** The correlation of intermediate layer representations with their final representation grows along the diagonal as training progresses, showing representations congeal to their final representation in a bottom-up fashion over the course of learning. **Top-down representation crawl:** The 1% rows reveal a tantalizing and subtle phenomenon. Early in training, higher layers of the network contain representations which are similar to the final representations in the bottom layers after convergence. This suggests that in the beginning of training, the network on all layers starts to learn the final lower-layer representations, and these are then squeezed from the top down to fit into the lower layers through the course of training (making way for final higher layer representations).

Layers are subspaces in function space. Since the activations of each neuron in a layer correspond to a vector in function space, the activations $\mathbf{h}^l$ of all the units in layer l to all inputs $\mathbf{x} \in \chi$ describes a subspace in function space. The subspace of functions computed by a layer describes the repre-

sentations in that layer, up to a trivial affine transformation. Note that affine transformations can be absorbed into the weights and biases of a layer readout.

Canonical Correlation Analysis (CCA) is a measurement of subspace similarity. Correlation provides a measure of similarity between vectors. CCA generalizes this, and provides a measure of similarity between subspaces. Letting c_j be the jth CCA coefficient, we define CCA similarity as $\rho = \sqrt{\frac{\sum_{j=1}^{n} \mathbb{E}_j[c_j]^2}{n}}$. If $\rho = 0$ between two layers, their representations are linearly independent. If $\rho = 1$, then the representation in the smaller layer corresponds to an affine transformation of the larger layer (if two layers are the same size, both can be predicted from the other).

3 Dynamics of Representation Learning

We apply this method to the setting where Net_S and Net_R are the same, with Net_R being the fully converged network at the final trainstep and Net_S the network in earlier trainsteps. We find two general conclusions:

- The network converges bottom up: layers become monotonically closer to their final representations, with the lowest layers converging most quickly, and the higher layers most slowly (see Figures 1 and 2).
- Representations 'crawl' down the network: higher layers in Net_S *begin* by showing greater similarity to lower layers of Net_R, before converging to their higher layer representations (see Figures 1 and 3).

3.1 Experiments

We trained a fully connected network on MNIST and a convolutional network on CIFAR-10.

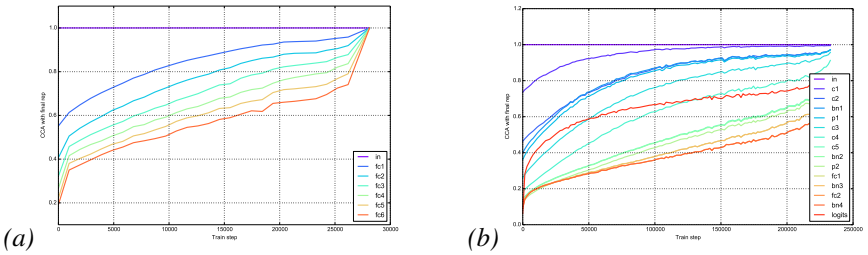

Figure 2: The representations in deep networks converge from the bottom up over the course of training. This figure shows the CCA similarity between each layer's representation at timestep t with its final representation at the end of training for *(a)* a fully connected network on MNIST, and *(b)* a CNN trained on CIFAR-10. Aside from the logit layer, which behaves differently, we see that the network demonstrates bottom up convergence: layers converge sequentially to their final representation with the lowest layers converging first, and the highest layers converging last.

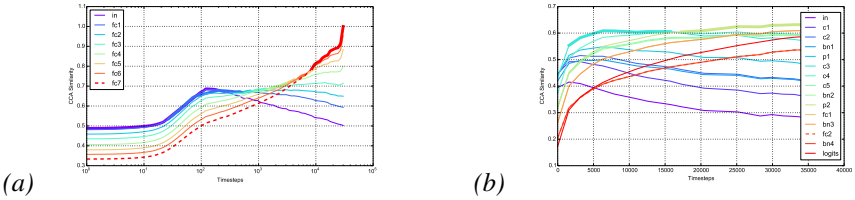

Figure 3: While convergence happens bottom up, representations may also crawl from the top down. The above figure takes one of the top layers for *(a)* a fully connected network on MNIST and *(b)* a convolutional network on CIFAR-10 and measures its CCA similarity to the final representations of all other layers in the network over the course of training. We see that initially, the final representations of the lower layers of the network are most similar to the evolving higher layer, suggesting that the initial representations learnt by the higher layer crawl down the network.

REFERENCES

G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *ArXiv e-prints*, October 2016.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft. Convergent Learning: Do different neural networks learn the same representations? In *International Conference on Learning Representations (ICLR)*, May 2016.