Insights on representational similarity in neural networks with canonical correlation

Ari S. Morcos[1*], Maithra Raghu[2*], and Samy Bengio[2]

[*]equal contribution, alphabetical order [1]Work done while at DeepMind, currently at Facebook AI Research (FAIR) [2]Google AI

Introduction

As neural networks have become more powerful, an increasing number of studies have sought to decipher their internal representations (Zeiler et al., 2014, Li et al., 2015, Bau et al., 2017, Arpit et al., 2017, Karpathy et al., 2015, Yosinksi et al., 2015, Morcos et al., 2018a). Most of these have focused on the role of individual units in the computations performed by individual networks. Comparing population representations across networks has proven especially difficult, largely because networks converge to apparently distinct solutions in which it is difficult to find one-to-one mappings of units (Li et al., 2015).

Recently, Raghu et al. (2017) applied Canonical Correlation Analysis (CCA) as a tool to compare representations across networks. Because CCA is invariant to linear transforms, it is capable of finding shared structure across representations which are superficially dissimilar, making CCA an ideal tool for comparing the representations across groups of networks and for comparing representations across time in RNNs.

Using CCA to investigate the representations of neural networks, we make three main contributions:

1. We analyse the technique introduced in Raghu et al., 2017, and identify a key challenge: the method does not effectively distinguish between the signal and the noise in the representation. We address this via a better aggregation technique.

2. Building off of Morcos et al., 2018a, we demonstrate that groups of networks which generalize converge to more similar solutions than those which memorize, that wider networks converge to more similar solutions than narrower networks , and that networks with identical topology but distinct learning rates converge to a small set of diverse solutions.

3. Using CCA to analyze RNN representations over training, we find that, as with CNNs , RNNs exhibit bottom-up convergence. Across sequence timesteps, however, we find that RNN representations vary significantly.

All CCA directions are not created equal

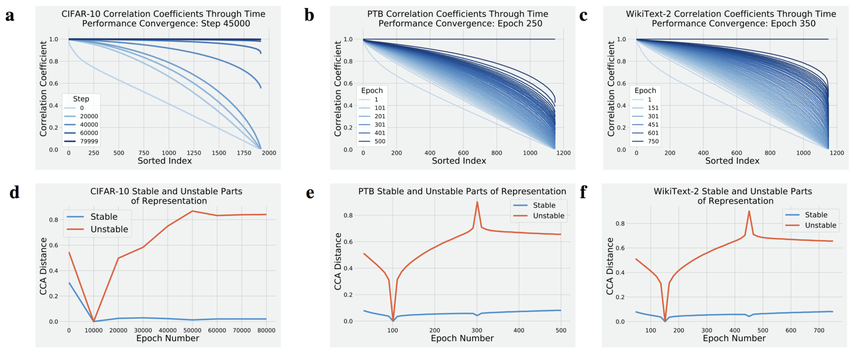

Figure 1: CCA distinguishes between stable and unstable parts of the representation over the course of training. Sorted CCA coefficients ($\rho_t^{(i)}$) comparing representations between layer L at times t through training with its representation at the final timestep T for CNNs trained on CIFAR-10 (**a**), and RNNs trained on PTB (**b**) and WikiText-2 (**c**). For all of these networks, at time $t_o < T$ (indicated in title), the performance converges to match final performance. However, many $\rho_t^{(i)}$ are unconverged, corresponding to unnecessary parts of the representation (noise). To distinguish between the signal and noise portions of the representation, we apply CCA between L at timestep t early in training, and L at timestep $T/2$ to get $\rho_{T/2}$. We take the 100 top converged vectors (according to $\rho_{T/2}$) to form S, and the 100 least converged vectors to form B. We then compute CCA similarity between S and L at time $t > t_{early}$, and similarly for B. S remains stable through training (signal), while B rapidly becomes uncorrelated (**d-f**). Note that the sudden spike at $T/2$ in the unstable representation is because it is chosen to be the least correlated with step $T/2$.

Projection weighted CCA is more stable to noise than mean CCA

$$CCA_{unweighted}(L_1, L_2) = 1 - \frac{1}{c}\sum_{i=1}^{c} \rho^i$$

$$\tilde{\alpha}^i = \sum_j |\langle h_i, z_j \rangle| \qquad \alpha^i = \frac{\tilde{\alpha}^i}{\sum^i \tilde{\alpha}^i}$$

$$CCA_{projection\_weighted}(L_1, L_2) = 1 - \sum_{i=1}^{c} \alpha^i \rho^i$$

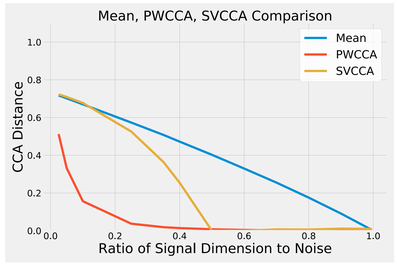

Figure 2: Projection weighted vs. unweighted mean. Unweighted mean (blue) and projection weighted mean (red) were used to compare networks with common (signal) and uncommon (noise) structure, each of fixed dimensionality. As the signal to noise ratio decreases, the unweighted mean underestimates the shared structure, while the projection weighted mean remains largely robust. L_1, L_2 - Activation matrices, ρ^i - CCA coefficients, h - CCA vectors, z - neuron activation vectors

Networks which generalize converge to more similar solutions than memorizing networks

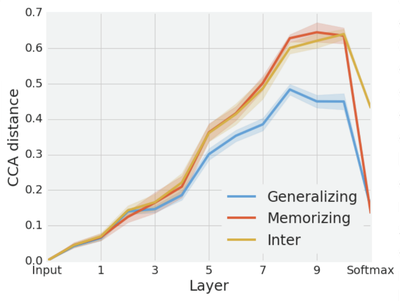

Figure 3: Generalizing networks converge to more similar solutions than memorizing networks. Groups of 5 networks were trained on CIFAR-10 with either true labels (generalizing) or random labels (memorizing). The pairwise CCA distance was then compared within each group and between generalizing and memorizing networks (inter) for each layer, based on the training data. While both categories converged to similar solutions in early layers, likely reflecting convergent edge detectors, etc., generalizing networks converge to significantly more similar solutions in later layers. At the softmax, sets of both generalizing and memorizing networks converged to nearly identical solutions, as all networks achieved near-zero training loss. Error bars represent mean ± std weighted mean CCA distance across pairwise comparisons.

Wider networks converge to more similar solutions than narrower networks

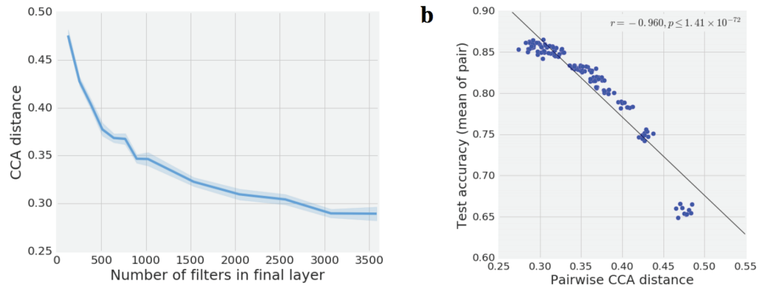

Figure 4: Wider networks converge to more similar solutions. Groups of 5 networks with different random initializations were trained on CIFAR-10. Pairwise CCA distance was computed for members of each group. Groups of larger networks converged to more similar solutions than groups of smaller networks (**a**). Test accuracy was highly correlated with degree of convergent similarity, as measured by CCA distance (**b**).

CCA reveals clusters of converged solutions

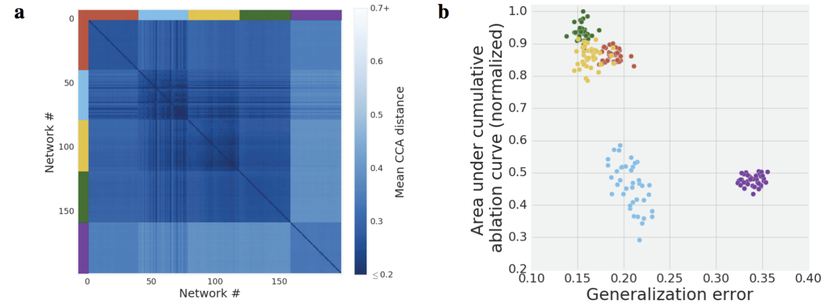

Figure 5: CCA reveals clusters of converged solutions across networks with different random initializations and learning rates. 200 networks with identical topology and varying learning rates were trained on CIFAR-10. CCA distance between the eighth layer of each pair of networks was computed, revealing five distinct subgroups of networks (**a**). These five subgroups align almost perfectly with the subgroups discovered in Morcos et al., 2018a (**b**; colors correspond to bars in **a**), despite the fact that the clusters in Morcos et al., 2018a were generated using robustness to cumulative ablation, an entirely separate metric.

RNNs exhibit bottom-up learning dynamics

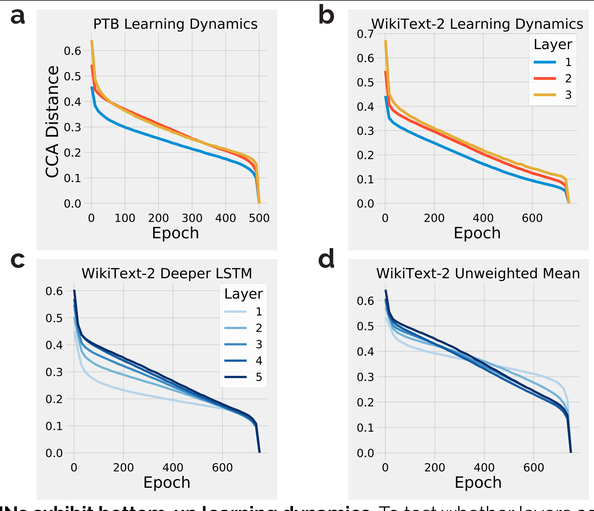

Figure 6: RNNs exhibit bottom-up learning dynamics. To test whether layers converge to their final representation over the course of training with a particular structure, we compared each layer's representation over the course of training to its final representation using CCA. In shallow RNNs trained on PTB (**a**), and WikiText-2 (**b**), we observed a clear bottom-up convergence pattern, in which early layers converge to their final representation before later layers. In deeper RNNs trained on WikiText-2, we observed a similar pattern (**c**). Importantly, the weighted mean reveals this effect much more accurately than the unweighted mean, which is also supported by control experiments (**d**), revealing the importance of appropriate weighting of CCA coefficients.

Summary

- Projection weighted CCA differentiates between signal and noise in CCA coefficients
- Generalizing networks converge to more similar solutions than memorizing networks
- Consistent with the lottery ticket hypothesis, wider networks converge to more similar solutions than narrow networks
- Trained networks with identical topology but different random seeds converge to distinct clusters with diverse representations, but often similar performance
- RNNs exhibit bottom-up learning dynamics
- Code available at: https://github.com/google/svcca