Google AI

Similarity of Neural Network Representations Revisited

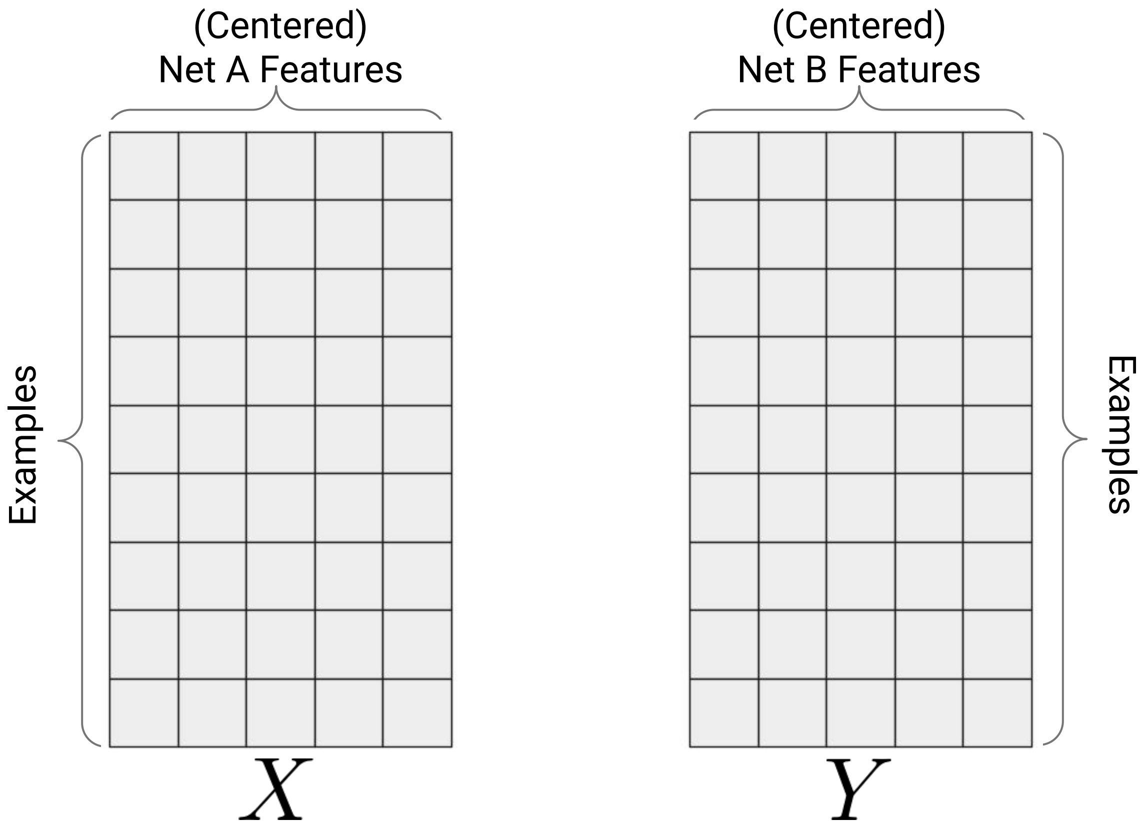
Simon Kornblith, Mohammad Norouzi, Honglak Lee, Geoffrey Hinton




Motivation

- We need tools to understand *trained* neural networks
    - Neural network training involves interactions between an algorithm and structured data
    - We don't know the structure of the data
- One way to understand trained neural networks is by comparing their representations

What is a Representation?



Google AI

Comparing Features = Comparing Examples

$$||X^{\mathrm{T}}Y||_{\mathrm{F}}^2 = \langle \mathrm{vec}(XX^{\mathrm{T}}), \mathrm{vec}(YY^{\mathrm{T}}) \rangle$$

Sum of squared dot products (similarities) between features

Dot product between reshaped inter-example similarity matrices

Comparing Features = Comparing Examples

$$||X^\mathrm{T}Y||_\mathrm{F}^2 = \langle \mathrm{vec}(XX^\mathrm{T}), \mathrm{vec}(YY^\mathrm{T}) \rangle$$

$$\frac{||X^\mathrm{T}Y||_\mathrm{F}^2}{||X^\mathrm{T}X||_\mathrm{F}||Y^\mathrm{T}Y||_\mathrm{F}} = \frac{\langle \mathrm{vec}(XX^\mathrm{T}), \mathrm{vec}(YY^\mathrm{T}) \rangle}{||XX^\mathrm{T}||_\mathrm{F}||YY^\mathrm{T}||_\mathrm{F}}$$

Comparing Features = Comparing Examples

$$||X^\mathrm{T}Y||_\mathrm{F}^2 = \langle \mathrm{vec}(XX^\mathrm{T}), \mathrm{vec}(YY^\mathrm{T}) \rangle$$

$$\frac{||X^\mathrm{T}Y||_\mathrm{F}^2}{||X^\mathrm{T}X||_\mathrm{F}||Y^\mathrm{T}Y||_\mathrm{F}} = \frac{\langle \mathrm{vec}(XX^\mathrm{T}), \mathrm{vec}(YY^\mathrm{T}) \rangle}{||XX^\mathrm{T}||_\mathrm{F}||YY^\mathrm{T}||_\mathrm{F}}$$

Centered kernel alignment (CKA) (Cortes et al., 2012)
RV-coefficient (Robert & Escoufier, 1976)
Tucker's congruence coefficient (Tucker, 1951)

The Kernel Trick

$$\frac{\langle \text{vec}(XX^{\text{T}}), \text{vec}(YY^{\text{T}})\rangle}{||XX^{\text{T}}||_{\text{F}}||YY^{\text{T}}||_{\text{F}}} \rightarrow \frac{\langle \text{vec}(\tilde{K}), \text{vec}(\tilde{L})\rangle}{||\tilde{K}||_{\text{F}}||\tilde{L}||_{\text{F}}}$$

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \quad \tilde{K} = HKH$$

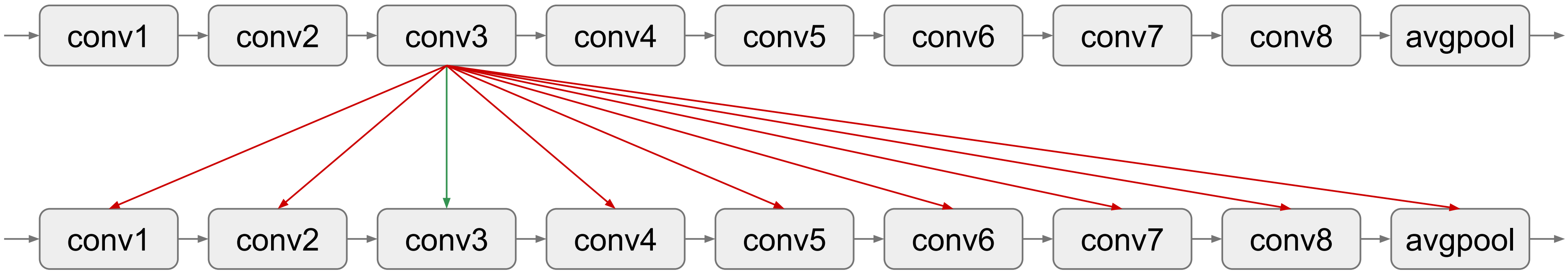
$$L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j) \quad \tilde{L} = HLH$$

H is the centering matrix

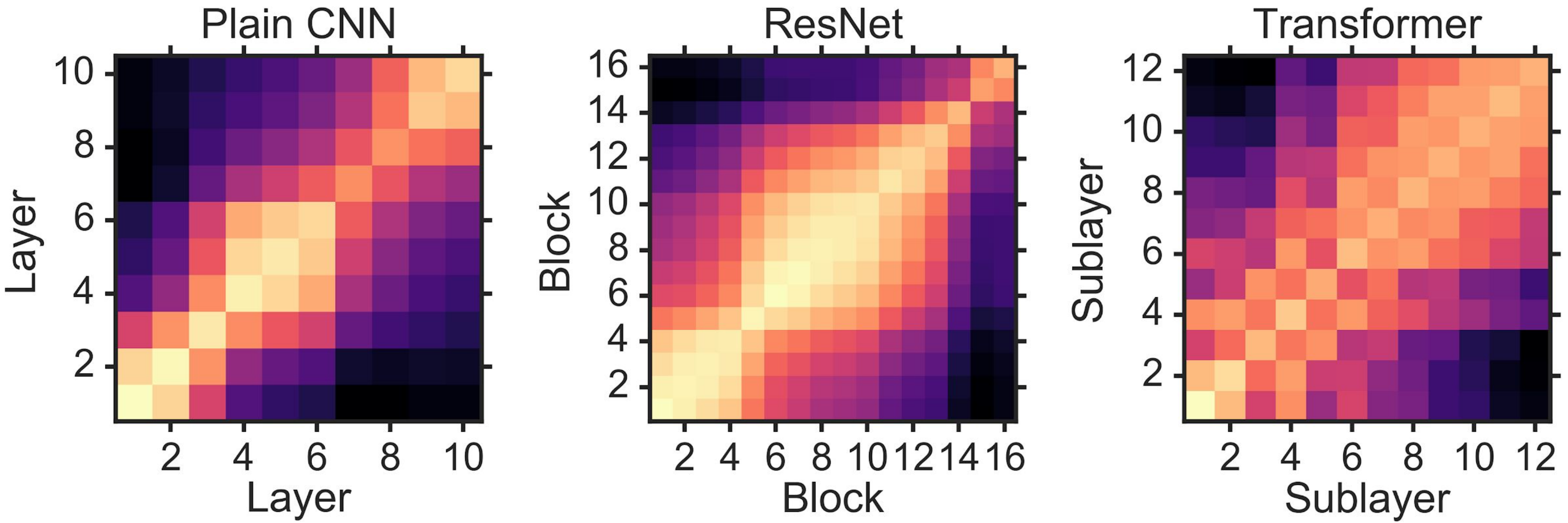
Google AI

A Sanity Check for Similarity

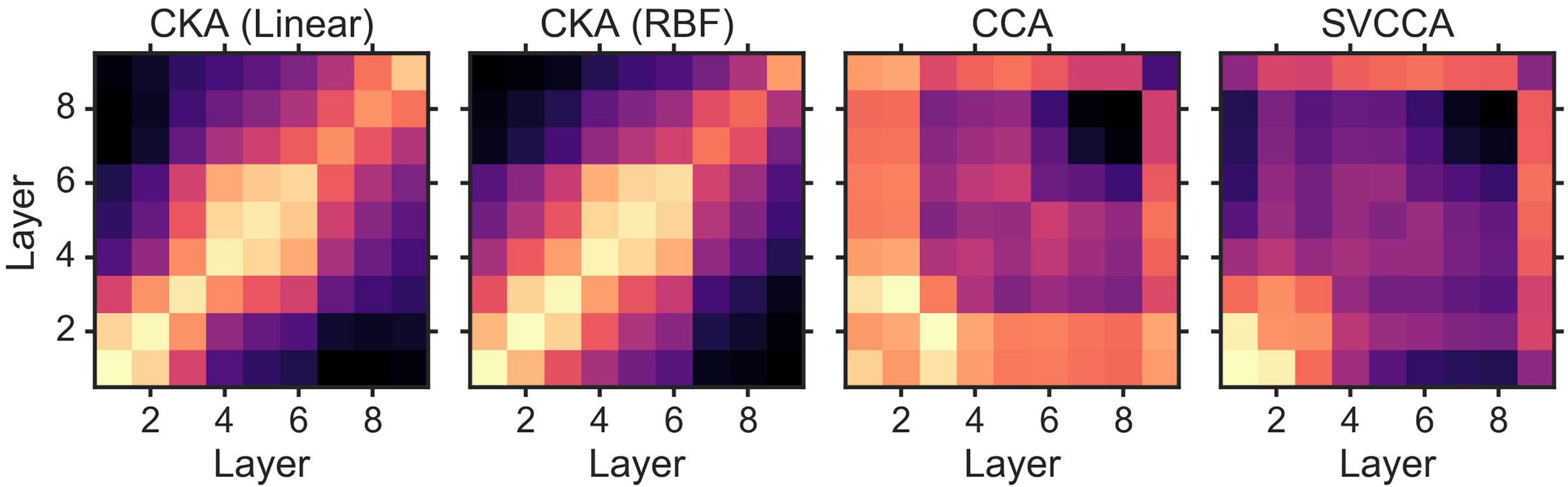
Given two architecturally identical networks A and B trained from different random initializations, a layer from net A should be most similar to the architecturally corresponding layer in net B



A Sanity Check for Similarity

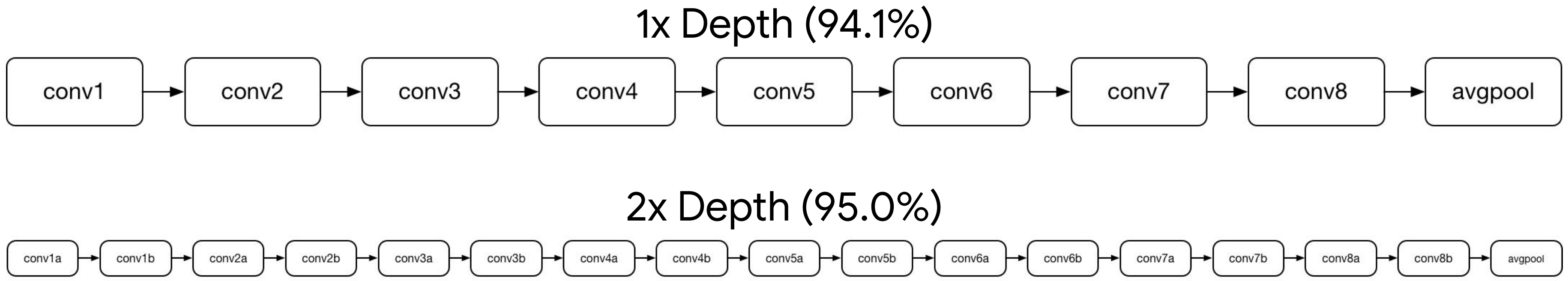


A Sanity Check for Similarity



CKA Reveals Network Pathology

CKA Reveals Network Pathology



CKA Reveals Network Pathology

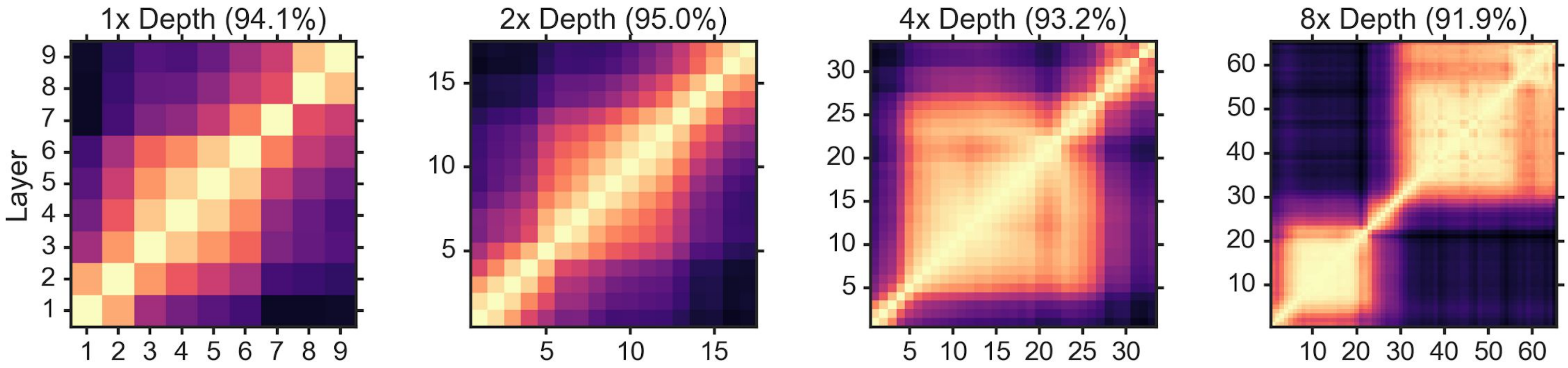
1x Depth (94.1%)

2x Depth (95.0%)

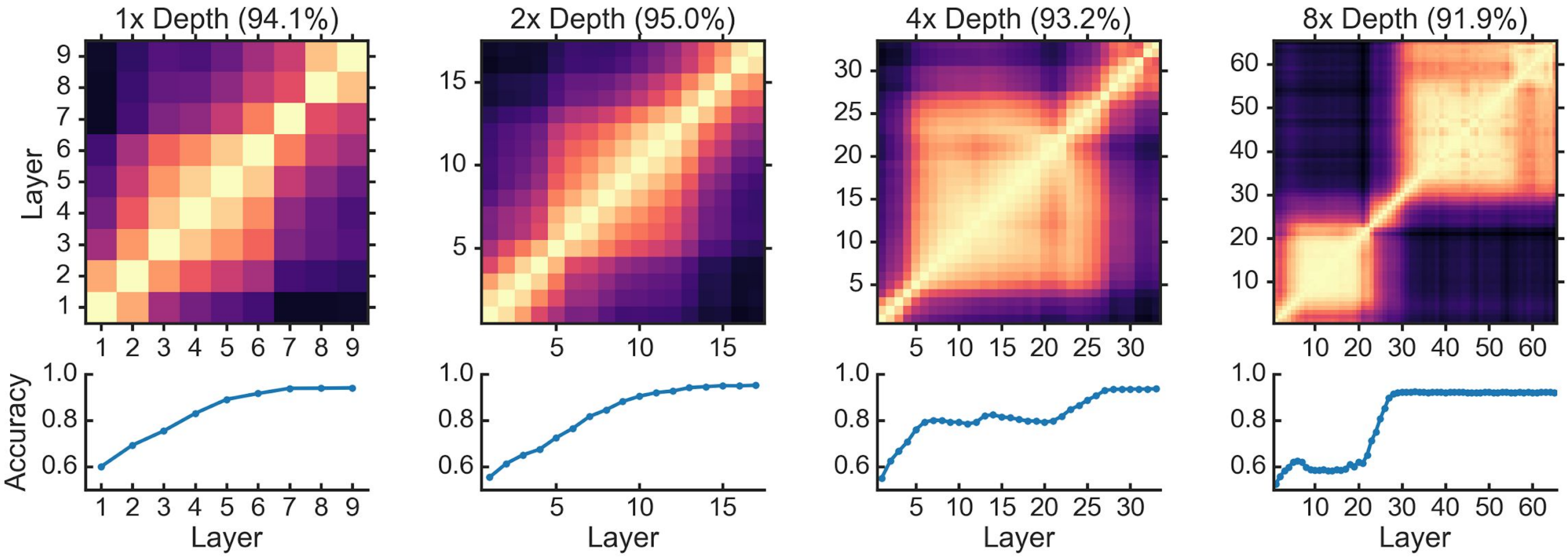
4x Depth (93.2%)

8x Depth (91.9%)

CKA Reveals Network Pathology



CKA Reveals Network Pathology



Similarity of Neural Network Representations Revisited

Thank You!

cka-similarity.github.io