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?

(Centered)
Net A Features

Examples

\(X\)

(Centered)
Net B Features

Examples

\(Y\)
Comparing Features = Comparing Examples

$$\|X^T Y\|_F^2 = \langle \text{vec}(XX^T), \text{vec}(YY^T) \rangle$$

- Sum of squared dot products (similarities) between features
- Dot product between reshaped inter-example similarity matrices
Comparing Features = Comparing Examples

$$\frac{\|X^T Y\|_F^2}{\|X^T X\|_F \|Y^T Y\|_F} = \frac{\langle \text{vec}(X X^T), \text{vec}(Y Y^T) \rangle}{\|X X^T\|_F \|Y Y^T\|_F}$$
Comparing Features = Comparing Examples

\[ \left\| X^T Y \right\|_F^2 = \langle \text{vec}(X X^T), \text{vec}(Y Y^T) \rangle \]

\[ \frac{\left\| X^T Y \right\|_F^2}{\left\| X^T X \right\|_F \left\| Y^T Y \right\|_F} = \frac{\langle \text{vec}(X X^T), \text{vec}(Y Y^T) \rangle}{\left\| X X^T \right\|_F \left\| Y Y^T \right\|_F} \]

Centered kernel alignment (CKA) (Cortes et al., 2012)

RV-coefficient (Robert & Escoufier, 1976)

Tucker’s congruence coefficient (Tucker, 1951)
The Kernel Trick

\[
\frac{\langle \text{vec}(XX^T), \text{vec}(YY^T) \rangle}{\|XX^T\|_F \|YY^T\|_F} \quad \rightarrow \quad \frac{\langle \text{vec}(\tilde{K}), \text{vec}(\tilde{L}) \rangle}{\|\tilde{K}\|_F \|\tilde{L}\|_F}
\]

\[
K_{ij} = k(x_i, x_j) \quad \tilde{K} = HHK
\]

\[
L_{ij} = l(y_i, y_j) \quad \tilde{L} = HLH
\]

\(H\) is the centering matrix
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

Similarity of Neural Network Representations Revisited
A Sanity Check for Similarity
CKA Reveals Network Pathology

1x Depth (94.1% on CIFAR-10)

conv1 → conv2 → conv3 → conv4 → conv5 → conv6 → conv7 → conv8 → avgpool
CKA Reveals Network Pathology

1x Depth (94.1%)

2x Depth (95.0%)
CKA Reveals Network Pathology

1x Depth (94.1%)

conv1 → conv2 → conv3 → conv4 → conv5 → conv6 → conv7 → conv8 → avgpool

2x Depth (95.0%)

conv1a → conv1b → conv2a → conv2b → conv3a → conv3b → conv4a → conv4b → conv5a → conv5b → conv6a → conv6b → conv7a → conv7b → conv8a → conv8b → avgpool

4x Depth (93.2%)

... (Diagram of 4x Depth)

8x Depth (91.9%)

... (Diagram of 8x Depth)
CKA Reveals Network Pathology

Similarity of Neural Network Representations Revisited
CKA Reveals Network Pathology

Similarity of Neural Network Representations Revisited
Thank You!

cka-similarity.github.io