Similarity of Neural Network Representations Revisited

Simon Kornblith [1] **Mohammad Norouzi** [1] **Honglak Lee** [1] **Geoffrey Hinton** [1]

Abstract

Recent work has sought to understand the behavior of neural networks by comparing representations between layers and between different trained models. We examine methods for comparing neural network representations based on canonical correlation analysis (CCA). We show that CCA belongs to a family of statistics for measuring multivariate similarity, but that neither CCA nor any other statistic that is invariant to invertible linear transformation can measure meaningful similarities between representations of higher dimension than the number of data points. We introduce a similarity index that measures the relationship between representational similarity matrices and does not suffer from this limitation. This similarity index is equivalent to centered kernel alignment (CKA) and is also closely connected to CCA. Unlike CCA, CKA can reliably identify correspondences between representations in networks trained from different initializations.

1. Introduction

Across a wide range of machine learning tasks, deep neural networks enable learning powerful feature representations automatically from data. Despite impressive empirical advances of deep neural networks in solving various tasks, the problem of understanding and characterizing the neural network representations learned from data remains relatively under-explored. Previous work (*e.g.* Advani & Saxe (2017); Amari et al. (2018); Saxe et al. (2014)) has made progress in understanding the theoretical dynamics of the neural network training process. These studies are insightful, but fundamentally limited, because they ignore the complex interaction between the training dynamics and structured data. A window into the network's representation can provide more information about the interaction between machine learning algorithms and data than the value of the loss function alone.

[1]Google Brain. Correspondence to: Simon Kornblith <skornblith@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

This paper investigates the problem of measuring similarities between deep neural network representations. An effective method for measuring representational similarity could help answer many interesting questions, including: (1) Do deep neural networks with the same architecture trained from different random initializations learn similar representations? (2) Can we establish correspondences between layers of different network architectures? (3) How similar are the representations learned using the same network architecture from different datasets?

We build upon previous studies investigating similarity between the representations of neural networks (Laakso & Cottrell, 2000; Li et al., 2015; Raghu et al., 2017; Morcos et al., 2018; Wang et al., 2018). We are also inspired by the extensive neuroscience literature that uses representational similarity analysis (Kriegeskorte et al., 2008a; Edelman, 1998) to compare representations across brain areas (Haxby et al., 2001; Freiwald & Tsao, 2010), individuals (Connolly et al., 2012), species (Kriegeskorte et al., 2008b), and behaviors (Elsayed et al., 2016), as well as between brains and neural networks (Yamins et al., 2014; Khaligh-Razavi & Kriegeskorte, 2014; Sussillo et al., 2015).

Our key contributions are summarized as follows:

- We discuss the invariance properties of similarity indexes and their implications for measuring similarity of neural network representations.
- We motivate and introduce *centered kernel alignment (CKA)* as a similarity index and analyze the relationship between CKA, linear regression, canonical correlation analysis (CCA), and related methods (Raghu et al., 2017; Morcos et al., 2018).
- We show that CKA is able to determine the correspondence between the hidden layers of neural networks trained from different random initializations and with different widths, scenarios where previously proposed similarity indexes fail.
- We verify that wider networks learn more similar representations, and show that the similarity of early layers saturates at fewer channels than later layers. We demonstrate that early layers, but not later layers, learn similar representations on different datasets.

Problem Statement

Let $X \in \mathbb{R}^{n \times p_1}$ denote a matrix of activations of p_1 neurons for n examples, and $Y \in \mathbb{R}^{n \times p_2}$ denote a matrix of activations of p_2 neurons for the same n examples. We assume that these matrices have been preprocessed to center the columns. Without loss of generality we assume that $p_1 \leq p_2$. We are concerned with the design and analysis of a scalar *similarity index* $s(X, Y)$ that can be used to compare representations within and across neural networks, in order to help visualize and understand the effect of different factors of variation in deep learning.

2. What Should Similarity Be Invariant To?

This section discusses the invariance properties of similarity indexes and their implications for measuring similarity of neural network representations. We argue that both intuitive notions of similarity and the dynamics of neural network training call for a similarity index that is invariant to orthogonal transformation and isotropic scaling, but not invertible linear transformation.

2.1. Invariance to Invertible Linear Transformation

A similarity index is invariant to invertible linear transformation if $s(X, Y) = s(XA, YB)$ for any full rank A and B. If activations X are followed by a fully-connected layer $f(X) = \sigma(XW + \beta)$, then transforming the activations by a full rank matrix A as $X' = XA$ and transforming the weights by the inverse A^{-1} as $W' = A^{-1}W$ preserves the output of $f(X)$. This transformation does not appear to change how the network operates, so intuitively, one might prefer a similarity index that is invariant to invertible linear transformation, as argued by Raghu et al. (2017).

However, a limitation of invariance to invertible linear transformation is that any invariant similarity index gives the same result for any representation of width greater than or equal to the dataset size, *i.e.* $p_2 \geq n$. We provide a simple proof in Appendix A.

Theorem 1. *Let X and Y be $n \times p$ matrices. Suppose s is invariant to invertible linear transformation in the first argument,* i.e. $s(X, Z) = s(XA, Z)$ *for arbitrary Z and any A with rank$(A) = p$. If rank$(X) =$ rank$(Y) = n$, then $s(X, Z) = s(Y, Z)$.*

There is thus a practical problem with invariance to invertible linear transformation: Some neural networks, especially convolutional networks, have more neurons in some layers than there are examples the training dataset (Springenberg et al., 2015; Lee et al., 2018; Zagoruyko & Komodakis, 2016). It is somewhat unnatural that a similarity index could require more examples than were used for training.

A deeper issue is that neural network *training* is not invariant to arbitrary invertible linear transformation of inputs or activations. Even in the linear case, gradient descent converges first along the eigenvectors corresponding to the largest eigenvalues of the input covariance matrix (LeCun et al., 1991), and in cases of overparameterization or early stopping, the solution reached depends on the scale of the input. Similar results hold for gradient descent training of neural networks in the infinite width limit (Jacot et al., 2018). The sensitivity of neural networks training to linear transformation is further demonstrated by the popularity of batch normalization (Ioffe & Szegedy, 2015).

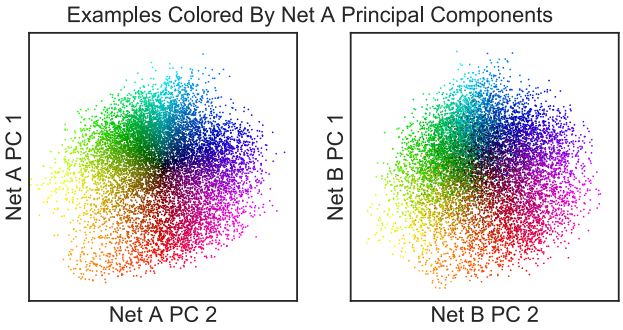

Figure 1. First principal components of representations of networks trained from different random initializations are similar. Each example from the CIFAR-10 test set is shown as a dot colored according to the value of the first two principal components of an intermediate layer of one network (left) and plotted on the first two principal components of the same layer of an architecturally identical network trained from a different initialization (right).

Invariance to invertible linear transformation implies that the scale of directions in activation space is irrelevant. Empirically, however, scale information is both consistent across networks and useful across tasks. Neural networks trained from different random initializations develop representations with similar large principal components, as shown in Figure 1. Consequently, Euclidean distances between examples, which depend primarily upon large principal components, are similar across networks. These distances are *meaningful*, as demonstrated by the success of perceptual loss and style transfer (Gatys et al., 2016; Johnson et al., 2016; Dumoulin et al., 2017). A similarity index that is invariant to invertible linear transformation ignores this aspect of the representation, and assigns the same score to networks that match only in large principal components or networks that match only in small principal components.

2.2. Invariance to Orthogonal Transformation

Rather than requiring invariance to any invertible linear transformation, one could require a weaker condition; invariance to orthogonal transformation, *i.e.* $s(X, Y) = s(XU, YV)$ for full-rank orthonormal matrices U and V such that $U^{\mathrm{T}}U = I$ and $V^{\mathrm{T}}V = I$.

Indexes invariant to orthogonal transformations do not share the limitations of indexes invariant to invertible linear transformation. When $p_2 > n$, indexes invariant to orthogonal transformation remain well-defined. Moreover, orthogonal transformations preserve scalar products and Euclidean distances between examples.

Invariance to orthogonal transformation seems desirable for neural networks trained by gradient descent. Invariance to orthogonal transformation implies invariance to permutation, which is needed to accommodate symmetries of neural networks (Chen et al., 1993; Orhan & Pitkow, 2018). In the linear case, orthogonal transformation of the input does not affect the dynamics of gradient descent training (LeCun et al., 1991), and for neural networks initialized with rotationally symmetric weight distributions, *e.g.* i.i.d. Gaussian weight initialization, training with fixed orthogonal transformations of activations yields the same distribution of training trajectories as untransformed activations, whereas an arbitrary linear transformation would not.

Given a similarity index $s(\cdot, \cdot)$ that is invariant to orthogonal transformation, one can construct a similarity index $s'(\cdot, \cdot)$ that is invariant to any invertible linear transformation by first orthonormalizing the columns of X and Y, and then applying $s(\cdot, \cdot)$. Given thin QR decompositions $X = Q_A R_A$ and $Y = Q_B R_B$ one can construct a similarity index $s'(X, Y) = s(Q_X, Q_Y)$, where $s'(\cdot, \cdot)$ is invariant to invertible linear transformation because orthonormal bases with the same span are related to each other by orthonormal transformation (see Appendix B).

2.3. Invariance to Isotropic Scaling

We expect similarity indexes to be invariant to isotropic scaling, *i.e.* $s(X, Y) = s(\alpha X, \beta Y)$ for any $\alpha, \beta \in \mathbb{R}^+$. That said, a similarity index that is invariant to both orthogonal transformation and non-isotropic scaling, *i.e.* rescaling of individual features, is invariant to any invertible linear transformation. This follows from the existence of the singular value decomposition of the transformation matrix. Generally, we are interested in similarity indexes that are invariant to isotropic but not necessarily non-isotropic scaling.

3. Comparing Similarity Structures

Our key insight is that instead of comparing multivariate features of an example in the two representations (*e.g.* via regression), one can first measure the similarity between every pair of *examples* in each representation separately, and then compare the similarity structures. In neuroscience, such matrices representing the similarities between examples are called representational similarity matrices (Kriegeskorte et al., 2008a). We show below that, if we use an inner product to measure similarity, the similarity between representational similarity matrices reduces to another intuitive notion of pairwise feature similarity.

Dot Product-Based Similarity. A simple formula relates dot products between examples to dot products between features:

$$\langle \mathrm{vec}(XX^\mathrm{T}), \mathrm{vec}(YY^\mathrm{T}) \rangle = \mathrm{tr}(XX^\mathrm{T}YY^\mathrm{T}) = ||Y^\mathrm{T}X||_\mathrm{F}^2. \tag{1}$$

The elements of XX^T and YY^T are dot products between the representations of the i^th and j^th examples, and indicate the similarity between these examples according to the respective networks. The left-hand side of (1) thus measures the similarity between the inter-example similarity structures. The right-hand side yields the same result by measuring the similarity between *features* from X and Y, by summing the squared dot products between every pair.

Hilbert-Schmidt Independence Criterion. Equation 1 implies that, for centered X and Y:

$$\frac{1}{(n-1)^2}\mathrm{tr}(XX^\mathrm{T}YY^\mathrm{T}) = ||\mathrm{cov}(X^\mathrm{T}, Y^\mathrm{T})||_\mathrm{F}^2. \tag{2}$$

The Hilbert-Schmidt Independence Criterion (Gretton et al., 2005) generalizes Equations 1 and 2 to inner products from reproducing kernel Hilbert spaces, where the squared Frobenius norm of the cross-covariance matrix becomes the squared Hilbert-Schmidt norm of the cross-covariance operator. Let $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j)$ where k and l are two kernels. The empirical estimator of HSIC is:

$$\mathrm{HSIC}(K, L) = \frac{1}{(n-1)^2}\mathrm{tr}(KHLH), \tag{3}$$

where H is the centering matrix $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\mathrm{T}$. For linear kernels $k(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\mathrm{T}\mathbf{y}$, HSIC yields (2).

Gretton et al. (2005) originally proposed HSIC as a test statistic for determining whether two sets of variables are independent. They prove that the empirical estimator converges to the population value at a rate of $1/\sqrt{n}$, and Song et al. (2007) provide an unbiased estimator. When k and l are universal kernels, HSIC = 0 implies independence, but HSIC is not an estimator of mutual information. HSIC is equivalent to maximum mean discrepancy between the joint distribution and the product of the marginal distributions, and HSIC with a specific kernel family is equivalent to distance covariance (Sejdinovic et al., 2013).

Centered Kernel Alignment. HSIC is not invariant to isotropic scaling, but it can be made invariant through normalization. This normalized index is known as centered kernel alignment (Cortes et al., 2012; Cristianini et al., 2002):

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\mathrm{HSIC}(L, L)}}. \tag{4}$$

Similarity Index	Formula	Invariant to: Invertible Linear Transform	Invariant to: Orthogonal Transform	Invariant to: Isotropic Scaling
Linear Reg. (R^2_{LR})	$\|\|Q_Y^{\text{T}} X\|\|_{\text{F}}^2 / \|\|X\|\|_{\text{F}}^2$	Y only	✓	✓
CCA (R^2_{CCA})	$\|\|Q_Y^{\text{T}} Q_X\|\|_{\text{F}}^2 / p_1$	✓	✓	✓
CCA ($\bar{\rho}_{\text{CCA}}$)	$\|\|Q_Y^{\text{T}} Q_X\|\|_* / p_1$	✓	✓	✓
SVCCA (R^2_{SVCCA})	$\|\|(U_Y T_Y)^{\text{T}} U_X T_X\|\|_{\text{F}}^2 / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	If same subspace kept	✓	✓
SVCCA ($\bar{\rho}_{\text{SVCCA}}$)	$\|\|(U_Y T_Y)^{\text{T}} U_X T_X\|\|_* / \min(\|\|T_X\|\|_{\text{F}}^2, \|\|T_Y\|\|_{\text{F}}^2)$	If same subspace kept	✓	✓
PWCCA	$\sum_{i=1}^{p_1} \alpha_i \rho_i / \|\|\alpha\|\|_1, \alpha_i = \sum_j \|\langle \mathbf{h}_i, \mathbf{x}_j \rangle\|$	✗	✗	✓
Linear HSIC	$\|\|Y^{\text{T}} X\|\|_{\text{F}}^2 / (n-1)^2$	✗	✓	✗
Linear CKA	$\|\|Y^{\text{T}} X\|\|_{\text{F}}^2 / (\|\|X^{\text{T}} X\|\|_{\text{F}} \|\|Y^{\text{T}} Y\|\|_{\text{F}})$	✗	✓	✓
RBF CKA	$\text{tr}(KHLH) / \sqrt{\text{tr}(KHKH)\text{tr}(LHLH)}$	✗	✓	✓*

Table 1. Summary of similarity methods investigated. Q_X and Q_Y are orthonormal bases for the columns of X and Y. U_X and U_Y are the left-singular vectors of X and Y sorted in descending order according to the corresponding singular vectors. $||\cdot||_*$ denotes the nuclear norm. T_X and T_Y are truncated identity matrices that select left-singular vectors such that the cumulative variance explained reaches some threshold. For RBF CKA, K and L are kernel matrices constructed by evaluating the RBF kernel between the examples as in Section 3, and H is the centering matrix $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\text{T}}$. See Appendix C for more detail about each technique.
*Invariance of RBF CKA to isotropic scaling depends on the procedure used to select the RBF kernel bandwidth parameter. In our experiments, we selected the bandwidth as a fraction of the median distance, which ensures that the similarity index is invariant to isotropic scaling.

For a linear kernel, CKA is equivalent to the RV coefficient (Robert & Escoufier, 1976) and to Tucker's congruence coefficient (Tucker, 1951; Lorenzo-Seva & Ten Berge, 2006).

Kernel Selection. Below, we report results of CKA with a linear kernel and the RBF kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||_2^2/(2\sigma^2))$. For the RBF kernel, there are several possible strategies for selecting the bandwidth σ, which controls the extent to which similarity of small distances is emphasized over large distances. We set σ as a fraction of the median distance between examples. In practice, we find that RBF and linear kernels give similar results across most experiments, so we use linear CKA unless otherwise specified. Our framework extends to any valid kernel, including kernels equivalent to neural networks (Lee et al., 2018; Jacot et al., 2018; Garriga-Alonso et al., 2019; Novak et al., 2019).

4. Related Similarity Indexes

In this section, we briefly review linear regression, canonical correlation, and other related methods in the context of measuring similarity between neural network representations. We let Q_X and Q_Y represent any orthonormal bases for the columns of X and Y, *i.e.* $Q_X = X(X^{\text{T}}X)^{-1/2}$, $Q_Y = Y(Y^{\text{T}}Y)^{-1/2}$ or orthogonal transformations thereof. Table 1 summarizes the formulae and invariance properties of the indexes used in experiments. For a comprehensive general review of linear indexes for measuring multivariate similarity, see Ramsay et al. (1984).

Linear Regression. A simple way to relate neural network representations is via linear regression. One can fit every feature in Y as a linear combination of features from X. A suitable summary statistic is the total fraction of variance explained by the fit:

$$R^2_{\text{LR}} = 1 - \frac{\min_B ||Y - XB||_{\text{F}}^2}{||Y||_{\text{F}}^2} = \frac{||Q_Y^{\text{T}} X||_{\text{F}}^2}{||X||_{\text{F}}^2}. \quad (5)$$

We are unaware of any application of linear regression to measuring similarity of neural network representations, although Romero et al. (2015) used a least squares loss between activations of two networks to encourage thin and deep "student" networks to learn functions similar to wide and shallow "teacher" networks.

Canonical Correlation Analysis (CCA). Canonical correlation finds bases for two matrices such that, when the original matrices are projected onto these bases, the correlation is maximized. For $1 \leq i \leq p_1$, the i^{th} canonical correlation coefficient ρ_i is given by:

$$\begin{aligned} \rho_i = \max_{\mathbf{w}_X^i, \mathbf{w}_Y^i} & \text{corr}(X\mathbf{w}_X^i, Y\mathbf{w}_Y^i) \\ \text{subject to} \quad & \forall_{j<i} \; X\mathbf{w}_X^i \perp X\mathbf{w}_X^j \\ & \forall_{j<i} \; Y\mathbf{w}_Y^i \perp Y\mathbf{w}_Y^j. \end{aligned} \quad (6)$$

The vectors $\mathbf{w}_X^i \in \mathbb{R}^{p_1}$ and $\mathbf{w}_Y^i \in \mathbb{R}^{p_2}$ that maximize ρ_i are the canonical weights, which transform the original data into canonical variables $X\mathbf{w}_X^i$ and $Y\mathbf{w}_Y^i$. The constraints in (6) enforce orthogonality of the canonical variables.

For the purpose of this work, we consider two summary

statistics of the goodness of fit of CCA:

$$R^2_{\text{CCA}} = \frac{\sum_{i=1}^{p_1} \rho_i^2}{p_1} = \frac{||Q_Y^{\text{T}} Q_X||_{\text{F}}^2}{p_1} \tag{7}$$

$$\bar{\rho}_{\text{CCA}} = \frac{\sum_{i=1}^{p_1} \rho_i}{p_1} = \frac{||Q_Y^{\text{T}} Q_X||_*}{p_1}, \tag{8}$$

where $||\cdot||_*$ denotes the nuclear norm. The mean squared CCA correlation R^2_{CCA} is also known as *Yanai's GCD measure* (Ramsay et al., 1984), and several statistical packages report the sum of the squared canonical correlations $p_1 R^2_{\text{CCA}} = \sum_{i=1}^{p_1} \rho_i^2$ under the name *Pillai's trace* (SAS Institute, 2015; StataCorp, 2015). The mean CCA correlation $\bar{\rho}_{\text{CCA}}$ was previously used to measure similarity between neural network representations in Raghu et al. (2017).

SVCCA. CCA is sensitive to perturbation when the condition number of X or Y is large (Golub & Zha, 1995). To improve robustness, *singular vector CCA* (SVCCA) performs CCA on truncated singular value decompositions of X and Y (Raghu et al., 2017; Mroueh et al., 2015; Kuss & Graepel, 2003). As formulated in Raghu et al. (2017), SVCCA keeps enough principal components of the input matrices to explain a fixed proportion of the variance, and drops remaining components. Thus, it is invariant to invertible linear transformation only if the retained subspace does not change.

Projection-Weighted CCA. Morcos et al. (2018) propose a different strategy to reduce the sensitivity of CCA to perturbation, which they term "projection-weighted canonical correlation" (PWCCA):

$$\rho_{\text{PW}} = \frac{\sum_{i=1}^{c} \alpha_i \rho_i}{\sum_{i=1} \alpha_i} \qquad \alpha_i = \sum_j |\langle \mathbf{h}_i, \mathbf{x}_j \rangle|, \tag{9}$$

where $\mathbf{x}_j$ is the j^{th} column of X, and $\mathbf{h}_i = X\mathbf{w}_X^i$ is the vector of canonical variables formed by projecting X to the i^{th} canonical coordinate frame. As we show in Appendix C.3, PWCCA is closely related to linear regression, since:

$$R^2_{\text{LR}} = \frac{\sum_{i=1}^{c} \alpha_i' \rho_i^2}{\sum_{i=1} \alpha_i'} \qquad \alpha_i' = \sum_j \langle \mathbf{h}_i, \mathbf{x}_j \rangle^2. \tag{10}$$

Neuron Alignment Procedures. Other work has studied alignment between individual neurons, rather than alignment between subspaces. Li et al. (2015) examined correlation between the neurons in different neural networks, and attempt to find a bipartite match or semi-match that maximizes the sum of the correlations between the neurons, and then to measure the average correlations. Wang et al. (2018) proposed to search for subsets of neurons $\tilde{X} \subset X$ and $\tilde{Y} \subset Y$ such that, to within some tolerance, every neuron in $\tilde{X}$ can be represented by a linear combination of neurons from $\tilde{Y}$ and vice versa. They found that the maximum matching subsets are very small for intermediate layers.

Mutual Information. Among non-linear measures, one candidate is mutual information, which is invariant not only to invertible linear transformation, but to any invertible transformation. Li et al. (2015) previously used mutual information to measure neuronal alignment. In the context of comparing representations, we believe mutual information is not useful. Given any pair of representations produced by deterministic functions of the same input, mutual information between either and the input must be at least as large as mutual information between the representations. Moreover, in fully invertible neural networks (Dinh et al., 2017; Jacobsen et al., 2018), the mutual information between any two layers is equal to the entropy of the input.

5. Linear CKA versus CCA and Regression

Linear CKA is closely related to CCA and linear regression. If X and Y are centered, then Q_X and Q_Y are also centered, so:

$$R^2_{\text{CCA}} = \text{CKA}(Q_X Q_X^{\text{T}}, Q_Y Q_Y^{\text{T}}) \sqrt{\frac{p_2}{p_1}}. \tag{11}$$

When performing the linear regression fit of X with design matrix Y, $R^2_{\text{LR}} = ||Q_Y^{\text{T}} X||_F^2 / ||X||_F^2$, so:

$$R^2_{\text{LR}} = \text{CKA}(XX^{\text{T}}, Q_Y Q_Y^{\text{T}}) \frac{\sqrt{p_1}||X^{\text{T}} X||_{\text{F}}}{||X||_{\text{F}}^2}. \tag{12}$$

When might we prefer linear CKA over CCA? One way to show the difference is to rewrite X and Y in terms of their singular value decompositions $X = U_X \Sigma_X V_X^{\text{T}}$, $Y = U_Y \Sigma_Y V_Y^{\text{T}}$. Let the i^{th} eigenvector of XX^{T} (left-singular vector of X) be indexed as $\mathbf{u}_X^{\text{i}}$. Then R^2_{CCA} is:

$$R^2_{\text{CCA}} = ||U_Y^{\text{T}} U_X||_{\text{F}}^2 / p_1 = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \langle \mathbf{u}_X^i, \mathbf{u}_Y^j \rangle^2 / p_1. \tag{13}$$

Let the i^{th} eigenvalue of XX^{T} (squared singular value of X) be indexed as λ_X^i. Linear CKA can be written as:

$$\begin{aligned} \text{CKA}(XX^{\text{T}}, YY^{\text{T}}) &= \frac{||Y^{\text{T}} X||_{\text{F}}^2}{||X^{\text{T}} X||_{\text{F}} ||Y^{\text{T}} Y||_{\text{F}}} \\ &= \frac{\sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \lambda_X^i \lambda_Y^j \langle \mathbf{u}_X^i, \mathbf{u}_Y^j \rangle^2}{\sqrt{\sum_{i=1}^{p_1} (\lambda_X^i)^2} \sqrt{\sum_{j=1}^{p_2} (\lambda_Y^j)^2}}. \end{aligned} \tag{14}$$

Linear CKA thus resembles CCA weighted by the eigenvalues of the corresponding eigenvectors, *i.e.* the amount of variance in X or Y that each explains. SVCCA (Raghu et al., 2017) and projection-weighted CCA (Morcos et al., 2018) were also motivated by the idea that eigenvectors that correspond to small eigenvalues are less important, but

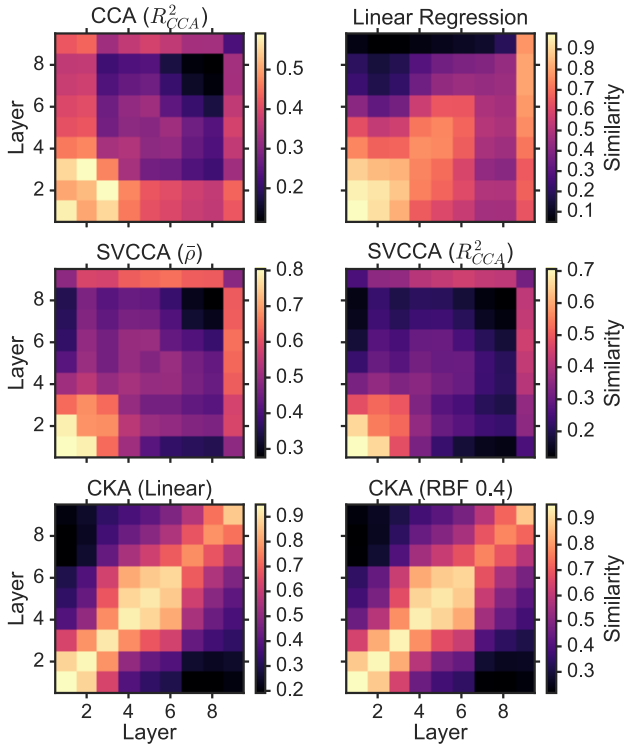

Figure 2. CKA reveals consistent relationships between layers of CNNs trained with different random initializations, whereas CCA, linear regression, and SVCCA do not. For linear regression, which is asymmetric, we plot R^2 for the fit of the layer on the x-axis with the layer on the y-axis. Results are averaged over 10 networks. See Table 2 for a numerical summary.

linear CKA incorporates this weighting symmetrically and can be computed without a matrix decomposition.

Comparison of (13) and (14) immediately suggests the possibility of alternative weightings of scalar products between eigenvectors. Indeed, as we show in Appendix D.1, the similarity index induced by "canonical ridge" regularized CCA (Vinod, 1976), when appropriately normalized, interpolates between R^2_{CCA}, linear regression, and linear CKA.

6. Results

6.1. A Sanity Check for Similarity Indexes

We propose a simple sanity check for similarity indexes: Given a pair of architecturally identical networks trained from different random initializations, for each layer in the first network, the most similar layer in the second network should be the architecturally corresponding layer. We train 10 networks and, for each layer of each network, we compute the accuracy with which we can find the corresponding layer in each of the other networks by maximum similarity. We then average the resulting accuracies. We compare CKA with CCA, SVCCA, PWCCA, and linear regression.

Index	Accuracy
CCA ($\bar{\rho}$)	1.4
CCA (R^2_{CCA})	10.6
SVCCA ($\bar{\rho}$)	9.9
SVCCA (R^2_{CCA})	15.1
PWCCA	11.1
Linear Reg.	45.4
Linear HSIC	22.2
CKA (Linear)	**99.3**
CKA (RBF 0.2)	80.6
CKA (RBF 0.4)	**99.1**
CKA (RBF 0.8)	**99.3**

Table 2. Accuracy of identifying corresponding layers based on maximum similarity for 10 architecturally identical 10-layer CNNs trained from different initializations, with logits layers excluded. For SVCCA, we used a truncation threshold of 0.99 as recommended in Raghu et al. (2017). For asymmetric indexes (PWCCA and linear regression) we symmetrized the similarity as $S + S^{\text{T}}$. CKA RBF kernel parameters reflect the fraction of the median Euclidean distance used as σ. Results not significantly different from the best result are bold-faced ($p < 0.05$, jackknife z-test).

We first investigate a simple VGG-like convolutional network based on All-CNN-C (Springenberg et al., 2015) (see Appendix E for architecture details). Figure 2 and Table 2 show that CKA passes our sanity check, but other methods perform substantially worse. For SVCCA, we experimented with a range of truncation thresholds, but no threshold revealed the layer structure (Appendix F.2); our results are consistent with those in Appendix E of Raghu et al. (2017).

We also investigate Transformer networks, where all layers are of equal width. In Appendix F.1, we show similarity between the 12 sublayers of the encoders of Transformer models (Vaswani et al., 2017) trained from different random initializations. All similarity indexes achieve non-trivial accuracy and thus pass the sanity check, although RBF CKA and R^2_{CCA} performed slightly better than other methods. However, we found that there are differences in feature scale between representations of feed-forward network and self-attention sublayers that CCA does not capture because it is invariant to non-isotropic scaling.

6.2. Using CKA to Understand Network Architectures

CKA can reveal pathology in neural networks representations. In Figure 3, we show CKA between layers of individual CNNs with different depths, where layers are repeated 2, 4, or 8 times. Doubling depth improved accuracy, but greater multipliers hurt accuracy. At 8x depth, CKA indicates that representations of more than half of the network are very similar to the last layer. We validated that these later layers do not refine the representation by training an ℓ^2-regularized logistic regression classifier on each layer of the network. Classification accuracy in shallower architectures progressively improves with depth, but for the 8x deeper

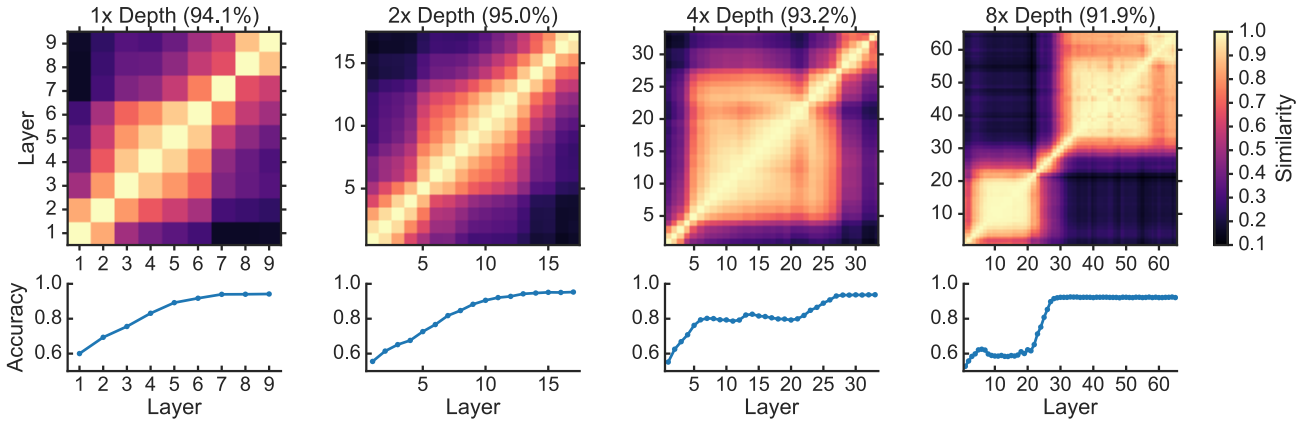

Figure 3. CKA reveals when depth becomes pathological. **Top**: Linear CKA between layers of individual networks of different depths on CIFAR-10. Titles show accuracy of each network. Later layers of the 8x depth network are similar to the last layer. **Bottom**: Accuracy of a logistic regression classifier trained on layers of the same networks is consistent with CKA.

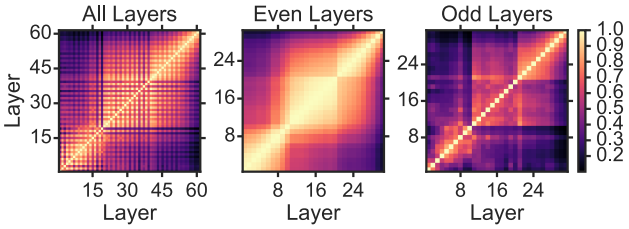

Figure 4. Linear CKA between layers of a ResNet-62 model. The grid pattern for ResNets in the left panel arises from the architecture. Right panels show similarity separately for even layer (post-residual) and odd layer (block interior) activations. Layers in the same block group (*i.e.* at the same feature map scale) are more similar than layers in different block groups.

network, accuracy plateaus less than halfway through the network. When applied to ResNets (He et al., 2016), CKA reveals no pathology (Figure 4). We instead observe a grid pattern that originates from the architecture: Post-residual activations are similar to other post-residual activations, but activations within blocks are not.

CKA is equally effective at revealing relationships between layers of different architectures. Figure 5 shows the relationship between different layers of networks with and without residual connections. CKA indicates that, as networks are made deeper, the new layers are effectively inserted in between the old layers. Other similarity indexes fail to reveal meaningful relationships between different architectures, as we show in Appendix F.5.

In Figure 6, we show CKA between networks with different layer widths. Like Morcos et al. (2018), we find that increasing layer width leads to more similar representations between networks. As width increases, CKA approaches 1; CKA of earlier layers saturates faster than later layers. Networks are generally more similar to other networks of the same width than they are to the widest network we trained.

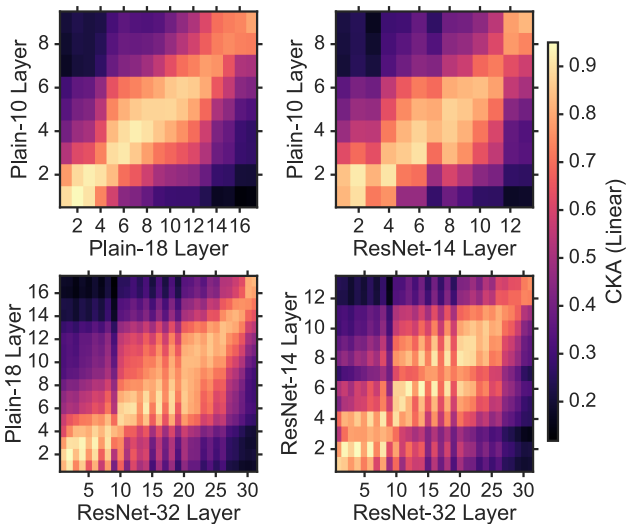

Figure 5. Linear CKA between layers of networks with different architectures.

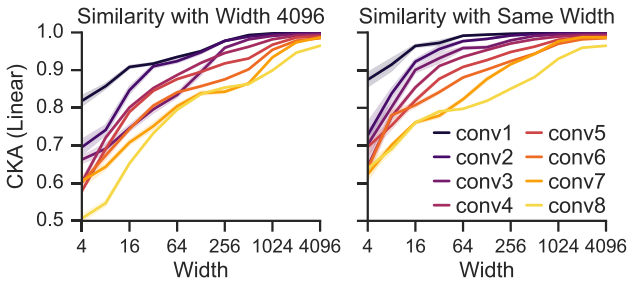

Figure 6. Layers become more similar to each other and to wide networks as width increases, but similarity of earlier layers saturates first. **Left**: Similarity of networks with the widest network we trained. **Middle**: Similarity of networks with other networks of the same width trained from random initialization. All CKA values are computed between 10 networks; shaded regions reflect jackknife standard error.

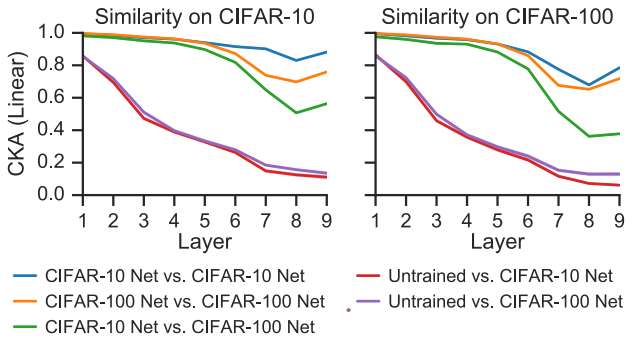

Figure 7. CKA shows that models trained on different datasets (CIFAR-10 and CIFAR-100) develop similar representations, and these representations differ from untrained models. The left panel shows similarity between the same layer of different models on the CIFAR-10 test set, while the right panel shows similarity computed on CIFAR-100 test set. CKA is averaged over 10 models of each type (45 pairs).

6.3. Similar Representations Across Datasets

CKA can also be used to compare networks trained on different datasets. In Figure 7, we show that models trained on CIFAR-10 and CIFAR-100 develop similar representations in their early layers. These representations require training; similarity with untrained networks is much lower. We further explore similarity between layers of untrained networks in Appendix F.3.

6.4. Analysis of the Shared Subspace

Equation 14 suggests a way to further elucidating what CKA is measuring, based on the action of one representational similarity matrix (RSM) YY^{T} applied to the eigenvectors $\mathbf{u}_X^i$ of the other RSM XX^{T}. By definition, $XX^{\mathrm{T}}\mathbf{u}_X^i$ points in the same direction as $\mathbf{u}_X^i$, and its norm $||XX^{\mathrm{T}}\mathbf{u}_X^i||_2$ is the corresponding eigenvalue. The degree of scaling and rotation by YY^{T} thus indicates how similar the action of YY^{T} is to XX^{T}, for each eigenvector of XX^{T}. For visualization purposes, this approach is somewhat less useful than the CKA summary statistic, since it does not collapse the similarity to a single number, but it provides a more complete picture of what CKA measures. Figure 8 shows that, for large eigenvectors, XX^{T} and YY^{T} have similar actions, but the rank of the subspace where this holds is substantially lower than the dimensionality of the activations. In the penultimate (global average pooling) layer, the dimensionality of the shared subspace is approximately 10, which is the number of classes in the CIFAR-10 dataset.

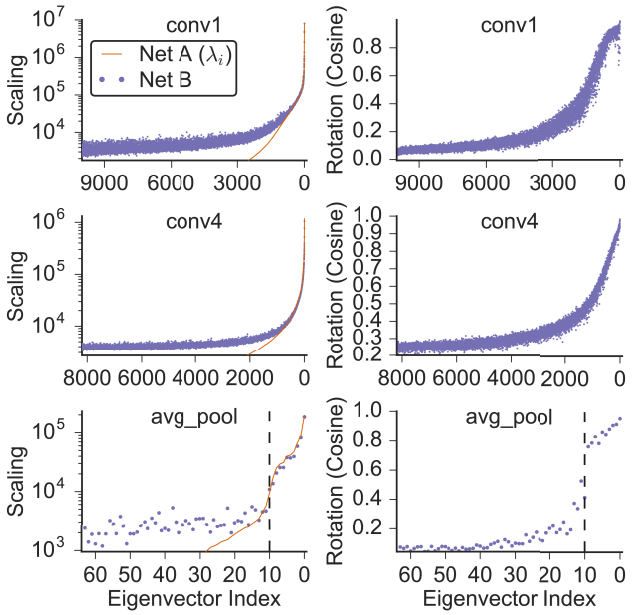

Figure 8. The shared subspace of two Tiny-10 networks trained from random initialization is spanned primarily by the eigenvectors corresponding to the largest eigenvalues. Each row represents a different network layer. Note that the average pooling layer has only 64 units. **Left**: Scaling of the eigenvectors $\mathbf{u}_X^i$ of the RSM XX^{T} from network A by RSMs of networks A and B. Orange lines show $||XX^{\mathrm{T}}\mathbf{u}_X^i||_2$, *i.e.* the eigenvalues. Purple dots show $||YY^{\mathrm{T}}\mathbf{u}_X^i||_2$, the scaling of the eigenvectors of the RSM of network A by the RSM of network B. **Right**: Cosine of the rotation by the RSM of network B, $(\mathbf{u}_X^i)^{\mathrm{T}}YY^{\mathrm{T}}\mathbf{u}_X^i/||YY^{\mathrm{T}}\mathbf{u}_X^i||_2$.

7. Conclusion and Future Work

Measuring similarity between the representations learned by neural networks is an ill-defined problem, since it is not entirely clear what aspects of the representation a similarity index should focus on. Previous work has suggested that there is little similarity between intermediate layers of neural networks trained from different random initializations (Raghu et al., 2017; Wang et al., 2018). We propose CKA as a method for comparing representations of neural networks, and show that it consistently identifies correspondences between layers, not only in the same network trained from different initializations, but across entirely different architectures, whereas other methods do not. We also provide a unified framework for understanding the space of similarity indexes, as well as an empirical framework for evaluation.

We show that CKA captures intuitive notions of similarity, *i.e.* that neural networks trained from different initializations should be similar to each other. However, it remains an open question whether there exist kernels beyond the linear and RBF kernels that would be better for analyzing neural network representations. Moreover, there are other potential choices of weighting in Equation 14 that may be more appropriate in certain settings. We leave these questions as future work. Nevertheless, CKA seems to be much better than previous methods at finding correspondences between the learned representations in hidden layers of neural networks.

Acknowledgements

We thank Gamaleldin Elsayed, Jaehoon Lee, Paul-Henri Mignot, Maithra Raghu, Samuel L. Smith, and Alex Williams for comments on the manuscript, Rishabh Agarwal for ideas, and Aliza Elkin for support.

References

Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Amari, S.-i., Ozeki, T., Karakida, R., Yoshida, Y., and Okada, M. Dynamics of learning in MLP: Natural gradient and singularity revisited. *Neural Computation*, 30(1): 1–33, 2018.

Björck, Å. and Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

Chen, A. M., Lu, H.-m., and Hecht-Nielsen, R. On the geometry of feedforward neural network error surfaces. *Neural Computation*, 5(6):910–927, 1993.

Connolly, A. C., Guntupalli, J. S., Gors, J., Hanke, M., Halchenko, Y. O., Wu, Y.-C., Abdi, H., and Haxby, J. V. The representation of biological classes in the human brain. *Journal of Neuroscience*, 32(8):2608–2618, 2012.

Cortes, C., Mohri, M., and Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(Mar):795–828, 2012.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, 2002.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. In *International Conference on Learning Representations*, 2017.

Edelman, S. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(4):449–467, 1998.

Elsayed, G. F., Lara, A. H., Kaufman, M. T., Churchland, M. M., and Cunningham, J. P. Reorganization between preparatory and movement population responses in motor cortex. *Nature Communications*, 7:13239, 2016.

Freiwald, W. A. and Tsao, D. Y. Functional compartmentalization and viewpoint generalization within the macaque face-processing system. *Science*, 330(6005):845–851, 2010.

Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. Deep convolutional networks as shallow Gaussian processes. In *International Conference on Learning Representations*, 2019.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Golub, G. H. and Zha, H. The canonical correlations of matrix pairs and their numerical computation. In *Linear Algebra for Signal Processing*, pp. 27–49. Springer, 1995.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory*, 2005.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Jacobsen, J.-H., Smeulders, A. W., and Oyallon, E. i-RevNet: Deep invertible networks. In *International Conference on Learning Representations*, 2018.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

Khaligh-Razavi, S.-M. and Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Computational Biology*, 10(11): e1003915, 2014.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008a.

Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., and Bandettini, P. A. Matching categorical object representations in inferior temporal

cortex of man and monkey. *Neuron*, 60(6):1126–1141, 2008b.

Kuss, M. and Graepel, T. The geometry of kernel canonical correlation analysis. Technical report, Max Planck Institute for Biological Cybernetics, 2003.

Laakso, A. and Cottrell, G. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical Psychology*, 13(1):47–76, 2000.

LeCun, Y., Kanter, I., and Solla, S. A. Second order properties of error surfaces: Learning time and generalization. In *Advances in Neural Information Processing Systems*, 1991.

Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. Convergent learning: Do different neural networks learn the same representations? In *NIPS 2015 Workshop on Feature Extraction: Modern Questions and Challenges*, 2015.

Lorenzo-Seva, U. and Ten Berge, J. M. Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology*, 2(2):57–64, 2006.

Morcos, A., Raghu, M., and Bengio, S. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, 2018.

Mroueh, Y., Marcheret, E., and Goel, V. Asymmetrically weighted CCA and hierarchical kernel sentence embedding for multimodal retrieval. *arXiv preprint arXiv:1511.06267*, 2015.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations*, 2019.

Orhan, E. and Pitkow, X. Skip connections eliminate singularities. In *International Conference on Learning Representations*, 2018.

Press, W. H. Canonical correlation clarified by singular value decomposition, 2011. URL http://numerical.recipes/whp/notes/CanonCorrBySVD.pdf.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, 2017.

Ramsay, J., ten Berge, J., and Styan, G. Matrix correlation. *Psychometrika*, 49(3):403–423, 1984.

Robert, P. and Escoufier, Y. A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Applied Statistics*, 25(3):257–265, 1976.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

SAS Institute. *Introduction to Regression Procedures*. 2015. URL https://support.sas.com/documentation/onlinedoc/stat/141/introreg.pdf.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, pp. 2263–2291, 2013.

Smith, S. L., Turban, D. H., Hamblin, S., and Hammerla, N. Y. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*, 2017.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bedo, J. Supervised feature selection via dependence estimation. In *International Conference on Machine learning*, 2007.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations Workshop*, 2015.

StataCorp. *Stata Multivariate Statistics Reference Manual*. 2015. URL https://www.stata.com/manuals14/mv.pdf.

Sussillo, D., Churchland, M. M., Kaufman, M. T., and Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7):1025, 2015.

Tucker, L. R. A method for synthesis of factor analysis studies. Technical report, Educational Testing Service, Princeton, NJ, 1951.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Vinod, H. D. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147–166, 1976.

Wang, L., Hu, L., Gu, J., Wu, Y., Hu, Z., He, K., and Hopcroft, J. E. Towards understanding learning representations: To what extent do different neural networks learn the same representation. In *Advances in Neural Information Processing Systems*, 2018.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.