PNNL-33472

DeepDataProfiler

A Platform and Methodology for the Analysis and Interpretation of Neural Networks

September, 2022

Brenda Praggastis
Davis Brown
Emilie Purvine
Madelyn Shapiro
Bei Wang (U Utah)

Prepared for the U.S. Department of Energy
Under contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33472

DeepDataProfiler

A Platform and Methodology for the Analysis and Interpretation of Neural Networks

September, 2022

Brenda Praggastis
Davis Brown
Emilie Purvine
Madelyn Shapiro
Bei Wang (U Utah)

Prepared for
the U.S. Department of Energy
Under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Abstract

The DeepDataProfiler is a methodology and framework for providing interpretability to trained neural networks. Its approach is to decompose a network into a weighted graph of neurons and synapses and link the components of the graph to human identifiable concepts. By identifying concepts important to the network and tracking the decision process employed by the network, the network becomes more transparent and less like a *black box*. Spurious decisions and poor generalization strategies can be identified and a measure of trustworthiness can be established.

The DeepDataProfiler project was funded by PNNL's Mathematics for Artificial Reasoning in Science (MARS) initiative. Its two year research objective was to apply techniques from topological data analysis (TDA) and graph theory to the problem of interpretability of convolutional neural networks (CNNs) used for the classification of images. This report will highlight the project's milestones, experiments, results, and transition strategy as well as its vision for the future.

Executive Summary

The process of training a deep neural network is often likened to the human development of semantic knowledge, described as the acquisition, category extraction, and hierarchical organization of experiences. Young children quickly identify trucks after their parents point and say the word enough times for them to catch on. Then, after countless evenings reading *Big Trucks Do Work* and looking at pictures of pickups, tractors, tow trucks, fire engines, cement mixers, and tankers they learn to recognize the distinguishing characteristics of each, and name them accordingly.[1] Neuroscience uses deep neural networks (DNN) to model this learning process and to develop explanations of how the synaptic connections, circuits, and neural pathways developed within the brain organize such categorical concepts into semantic knowledge [1]. The DeepDataProfiler (DDP) is a methodology and framework for understanding the fundamental synaptic connections, circuits, and pathways used by trained DNNs for classification tasks.

The DDP approach is to decompose a trained DNN into a weighted graph of representative neurons (activations and/or groups of activations) and link them to human identifiable concepts, providing both interpretability and trustworthiness for the network. Individual inputs are assigned *profile graphs*, a weighted graph linking neurons by their mutual contribution and weighting both neurons and their connections by their importance to the network's classification ability. Profile graphs are studied and compared using TDA and graph theory. Similarities in the structural characteristics of the graphs correlate to similarities in the underlying data.

Activations are projections of the output tensors of the individual linear layers of a DNN. For large activation values to persist through non-linear layers they must be highly correlated with the left singular vectors of a matrization of the weight tensors [2]. Saxe et al [1] demonstrate that these correlations reveal semantic distinctions in the hierarchy of classification and the singular values are indicators of the strength of these distinctions in the data. More meaningful than the *Euclidean projections* used for the profile graphs, the singular vectors can be ranked by their associated singular values. Mahoney and Martin [3, 4] demonstrate empirically that it is only when these singular values converge to a certain distribution that the network has been adequately trained, implying that it is these very singular values which the network depends upon to encode learned concepts.

By optimizing inputs to be highly correlated to specific projections of the weight tensors, DDP exposes the features of the input the network is most interested in. For image classification networks this optimization is called *feature visualization* (FV) [5]. Figure 1 illustrates both a profile graph and feature visualizations for an eagle image classified by VGG-16 [6].

DDP has a working pipeline to generate, analyze, and visualize a variety of profiles based on tensor slicing with multiple weighting schemes. DDP also produces profiles using singular vectors which are based not on inter-layer graphs but by intra-layer hypergraphs. Experimental evidence and the use of singular vectors in [3, 4, 1] indicate that these may offer better interpretability.

MARS/DDP had three software releases [7]. Project code and tutorials are publicly available on GitHub [8]. A lightweight proof-of-concept interactive feature visualization tool is available through Streamlit [9]. A research article has been submitted for publication and is currently on arXiv [10]. DDP's TRL is between 3 and 4.

[1]Anecdotal experience of authors.

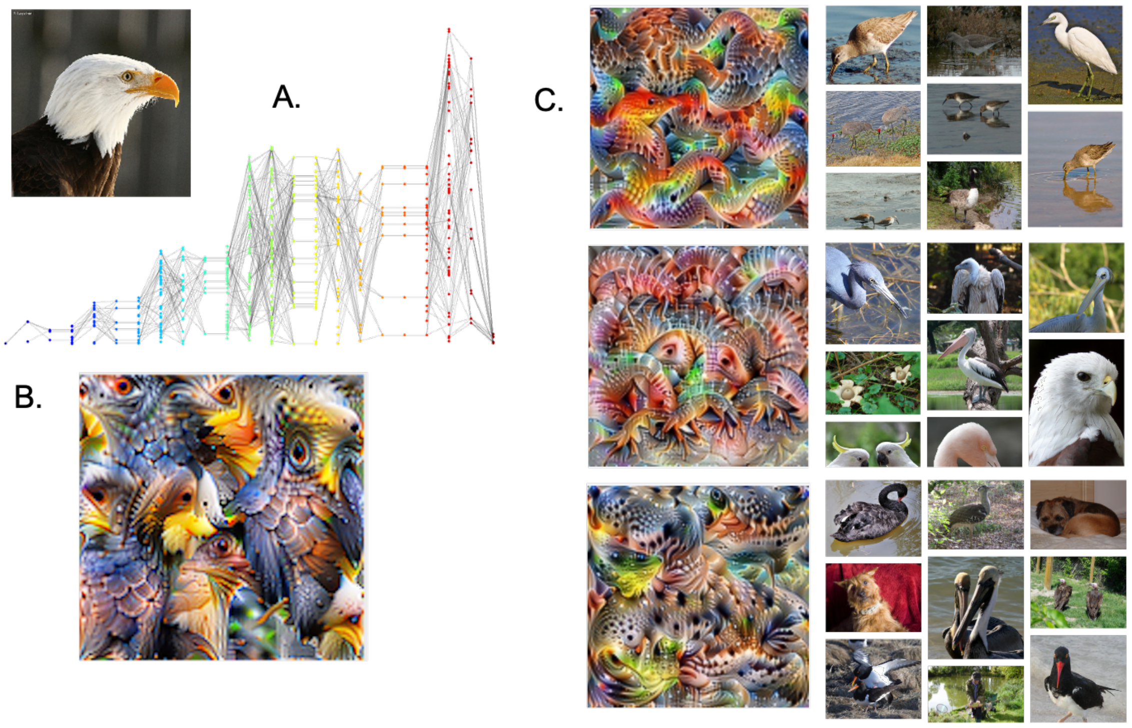


Figure 1. **Profiling the Imagenet dataset on pretrained VGG16 model from Torchvision.** **A**. The profile graph of an eagle image. **B**. Feature visualization of an eagle profile graph. **C**. Feature visualization of three left singular vectors of the weight tensor for the last convolutional layer, which are positively correlated with the eagle image. To the right of each optimized image are other images which also positively correlated with these vectors.

Acronyms and Abbreviations

CNN	convolutional neural network
DDP	DeepDataProfiler
DL	deep learning
DNN	deep neural network
FV	feature visualization
MARS	Mathematics for Artificial Reasoning in Science
PNNL	Pacific Northwest National Laboratory
SVD	singular value decomposition
TDA	topological data analysis

Acknowledgments

This research was supported by the Mathematics for Artificial Reasoning in Science (MARS) initiative via the Laboratory Directed Research and Development (LDRD) investments at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL0-1830.

Contents

Abstract iv
Executive Summary v
Acronyms and Abbreviations vi
Acknowledgments vii

1.0 Background 1

2.0 Project milestones FY21 3
  2.1 Profile Graphs 3
    Element to element profile graphs 3
    Tensor slice to tensor slice profile graphs 4
  2.2 Analysis of Profile Graphs 5
    Jaccard Similarity Metrics 5
    Topological Data Analysis 5
    Feature Visualization 8

3.0 Project milestones FY22 10
  3.1 The SVD 10
  3.2 Hypergraph Profiles 10
  3.3 Application in Nuclear Forensics 11
    Supervised nuclear forensics model 11
    Unsupervised nuclear forensics model 12

4.0 Conclusion 15

5.0 Bibliography 16

Figures

1 **Profiling the Imagenet dataset on pretrained VGG16 model from Torchvision. A**. The profile graph of an eagle image. **B**. Feature visualization of an eagle profile graph. **C**. Feature visualization of three left singular vectors of the weight tensor for the last convolutional layer, which are positively correlated with the eagle image. To the right of each optimized image are other images which also positively correlated with these vectors. vi

2 3-medoids clustering by class profile Jaccard on ImageNet reveals broad groups for objects, mammals, and non-mammals. 6

3 10-medoids clustering by class profile Jaccard on subsets of ImageNet (animal classes vs object classes) 6

4 We define a metric space on the vertices of a profile graph, which we can then analyze using persistent homology. 6

5 Persistent homology for a profile graph (ImageNet on VGG-16). 7

6 Feature visualizations optimizing neurons in profile graphs for a single image at different thresholds. Graph is pictured aligning neurons by their layer, from early layers on the left to later layers on the right. The example comes from VGG16 trained on Imagenet. 8

7 Nodes from profile graph in Figure 2.2 are clustered into 50 clusters. Choosing one cluster, we optimize an image for each neuron in the cluster noting where they occur in the profile and a single image optimizing for all neurons in the profile. . . . 9

8 **Illustration of a hypergraph and its semantic hierarchy.** Data comes from a simple CNN trained on MNIST. Nodes in the hypergraph correspond to one or more singular vectors in the first fully connected layer. Visualizations activating each of these vectors are shown the diagram along with exemplary images, which highly activate each singular vector [10]. 11

9 **Exploration of VGG-16 [6] on CIFAR-10 [11]** – Diagram for part of the semantic hierarchy induced by a hypergraph for the 7th convolutional layer [10] 12

10 **Exploration of VGG-16 [6] on ImageNet [12] – Significant features shared by two images.** The hypergraph models the relationship between singular vectors highly significant for two dog images from the 11th convolutional layer. Exemplary images and feature visualization are for the singular vector v_7. Similarity overlays highlight cosine similarity between the latent representations of the three images [10] 13

11 **MISO Signal, (final convolution on 3rd block)**. A feature visualization optimized for singular vector 16 as well as exemplary images included with their projected activations. Positive signals corresponded to negative space for $SDU \to UO_2$ in later layers. 14

12 **Comparing the unsupervised VQ-VAE with the supervised MISO model.** 14

1.0 Background

There seems to be an inverse relationship between the accuracy of deep neural classification networks (DNNs) and our human ability to understand how they make their decisions and verify their reliability. The more reliable and generalizable a model, the more impenetrable its internal black box seems to be. But acquiring the ability to interpret the reasoning of a DNN, to answer the questions “Why does this work?” and “How did it know?” and to be able to unambiguously explain to someone the causal connections made from input to classification is paramount if DNNs are to be relied upon to inform our decisions To this end there are several methodologies for obtaining interpretability of DNNs and extensive summaries of their inner workings and success [13, 14, 15, 16, 17, 18]. At a very high level these approaches share a common goal: identify the most important input features the model uses to make its classifications and if possible trace their interrelationships and causal connections. Interpretability methods often avoid the black box nature of the model by probing the inputs and activations, the layerwise outputs [19, 18]. The idea is that the black box might be impenetrable but, by closely approximating its behavior, interpretability can be achieved through proxy.

Convolutional neural networks (CNNs) used for image classification provide the most accessible opportunities to derive semantic meaning from the activations because the features of interest are visual objects. Influential activations are identified by probing the network with image tensors and measuring their activations [5]. Strong responses in convolutional layers indicate strong correlation (positive or negative) to the filters in the weight tensor, and in fully connected layers to the rows of the weight matrix. In the remainder of this section, we summarize some existing approaches to CNN interpretability and contrast with our approach.

Class activation maps [20] and Grad-CAM [21] use heat maps to identify the most influential regions of images in image classification networks, the regions with strongest responses. These tools are particularly powerful when the images are well understood and the identified regions contain interpretable features. When the identified region contains multiple objects, textures, shapes, colors, or other attributes, which are not separable or perhaps even human interpretable, saliency maps narrow the scope of where to look for features influencing classification. But saliency map methods have been shown to give misleading explanations, perhaps most notably giving similar saliency outputs for models with random weights [22].

Concept activation vectors (CAVs) [18] are used to determine a model’s tendency to classify to a specific class when that concept is present. Trained to separate output activations at a certain layer by a specific concept, these vectors can validate the model’s sensitivity to the concept by strength of response, allowing domain experts to determine if the features they deem important are important to the model. This can be taken further by learning a basis of class specific CAVs for the activation set [23, 24]. Activation vectors are then expressed as linear combinations of class specific concept vectors.

LIME [16] emulates the behavior of a DNN. By creating a simple interpretable classifier to match a model’s behavior in a small neighborhood of a single image it provides insight as to what the model considers important for the image’s classification in terms of predefined human identifiable concepts.

These interpretability tools are useful for verifying the model’s sensitivity to important domain-centric concepts. Their disadvantage is that they introduce a prior of what concepts to focus on and can miss potential influencers that cannot be humanly identified but are crucial for the model’s accuracy. While they measure sensitivity to humanly understandable concepts, they don’t explain the actual workings of the model.

We assert that interpretability research starting from the premise that latent representations of trustworthy models must correspond to known domain-centric concepts assumes advance knowledge of everything in the domain that is meaningful. This could produce bias and eliminate the possibility that concepts used by the model to describe class distinctions could be very different than what is expected and yet still be domain-centric and legitimate for classification.

As a consequence we took a different approach. Once a model is trained, its parameters are known and it becomes a deterministic function composed of linear and non-linear functions. Moreover, while the number of individual parameters may be huge, they are still seated within affine linear transformations; and linear transformations have been well understood for centuries. We extract and interpret the features identified by a trained model as important and then look for semantic meaning. The subtle difference in discovering the semantics of the network versus the sensitivity of the network to predefined concepts is discussed extensively in Olah et al. [25, 26].

DDP was inspired by the work of Qiu, et al. [27]. They decompose a trained model into functional blocks and extract an *effective path* linking the most influential neuron activations used for an input's classification by thier mutual contributions. Effective paths of inputs belonging to the same class tend to share more neurons and synaptic links than those belonging to different classes. Qiu, et al. use these paths to detect anomalous behaviors and adversarial inputs. DDP repurposes their work to generate *profile graphs* similar to effective paths but designed to make the model's decision process more transparent [28]. DDP profiles identify the indices of the influential neural activations, spatial activations, and channels used for an input's classification, then applies feature optimization techniques [5, 25] to link the corresponding activations to semantically meaningful concepts in the input domain. Influential activations are identified by the strength of their correlation with projections of the weight tensors. But correlation does not imply causation. DDP takes the next step to ask what a strong correlation really means for the linear layers.

The singular value decomposition (SVD) holds a special place in the heart of data science and numerical linear algebra in general [29]. As a robust matrix factorization method it facilitates data compression [30, 31] and network pruning [32]. For DDP the unitary nature of the singular vectors make the SVD an ideal factorization for understanding the dynamics, and hence the correlations measured by the weight tensors.

In the case of a multi-layer perceptron, domain concepts producing activations highly correlated to a singular vector are scaled by the corresponding singular value. We claim these basic correlations drive the success of the model so that interpretability rests largely on learning what input features correlate most closely to the singular vectors with the largest singular values. We give strong mathematical arguments for this assertion and evidence of how this knowledge can be applied.

In summary, the MARS/DDP project used two approaches for the problem of interpretability. For FY21 the goal was to represent the decision strategy for image classification as a profile graph of influential activations and synaptic connections and explain its structural and topological characteristics in terms of invariant structures used for classification. During FY22 the project widened its scope to a study of intra layer hypergraphs linking the singular vectors associated with the weight tensors.

2.0 Project milestones FY21

The first year focused on generating profile graphs for two benchmark datasets: CIFAR-10 [11] and ImageNet [12], and studying their topology. The data was classified using Torchvision [33] models and pretrained weights, when available. DDP used both VGG and ResNet architectures. We also explored data visualization methods to flesh out visual interpretations of the profiles. We give highlights below.

2.1 Profile Graphs

A profile graph describes the dependency relationships between positions in the activations for a specific input. Our construction builds upon ideas put forth by Qiu et al [27]. Every input for the DNN generates a network of activation tensors. Each element and slice of an activation is linked functionally to corresponding pieces of activation tensors coming from a predecessor layer. We reference the pieces of the activations as neurons and their functional relationships as synapses, and construct a profile graph with neurons represented by nodes and synapses represented by edges weighted by contribution. To illustrate, let $\mathcal{L}$ be a single convolutional layer composed of cross-correlation of an input tensor X with a weighted tensor W, followed by the addition of a bias tensor B, followed by a nonlinear operation σ. Let $\mathcal{L}(X)$ represent the tensor of activations generated by layer $\mathcal{L}$ for X so that

$$\mathcal{L}(X) = \sigma(W \star X + B).^{2} \tag{1}$$

We note the input X is the output of a predecessor layer to $\mathcal{L}$.[3] A profile graph will describe the functional relationships between X and $\mathcal{L}(X)$ in terms of indices, not values. This is important because it ties an input to the portion of the network most influential in its classification and permits meaningful comparison. It is important to note that linking by functional relationships depends upon the linking neurons, that are the projection of activations onto the Euclidean basis, and the activation function, which operates independently on the subspaces generated by each basis element. Let $Y = W \star X$.

Element to element profile graphs

An element of a d-tensor is a single number identified by its d-tuple index. Let a neuron correspond to an element of an activation tensor and reference it by the layer of the activation and its index. For example neuron $(10, (5, 25, 13))$ would reference the element in the activation 3-tensor from layer 10 with index $(5, 25, 13)$. We will always assume the nonlinear function σ is an element-wise operation so that once an index has been identified in $\mathcal{L}(X)$, it can be used as a reference index in Y for computing functional dependencies with elements of X.

Synaptic connections are defined by the functional relationships between the receptive fields in X and each element in Y. A single synapse exists between the neuron corresponding to $Y_{h,i,j}$ and each neuron corresponding to each of the elements $X_{r,i+s,j+t}$ found in the summands. The synapse is referenced by the pair of corresponding neuron references and weighted by some function of the proportional contribution the corresponding summand brings to $Y_{h,i,j}$.

[2]We use the $\star$ instead of $*$ here to remind us the operation happening inside a typical *convolution* layer is actually *cross-correlation*.

[3]We treat concatenation of multiple layers as its own layer and will treat it separately.

A profile graph is created from the most *influential* of these neurons and synapses. This, of course, is a subjective decision. By choosing too many, we could end up with unwieldy graphs that can't be distinguished from the original network. Choosing too few produces graphs that can't be distinguished from each other. The *sweet spot* is a graph that retains the smallest number of influential neurons and synapses required to distinguish inputs by class.

The profile graph is constructed in two phases. First, identify the elements of each output activation that pass some threshold value or are in the a top percentage of values; this defines a set of *influential neurons*. Second, for each influential neuron with index (h, i, j), find the functional relationship between $Y_{h,i,j}$ and the elements of X, where X is the output from a predecessor layer. As long as the nonlinear function of the layer is monotonically increasing, it is sufficient to order the summands $\{W_{h,r,s,t} \cdot X_{r,i+s,j+t}\}$ in descending order and identify the indices of the top contributors to $Y_{h,i,j}$ in X either by thresholding over the cumulative sums or taking a top percentage. The profile graph takes as its nodes the chosen influential neurons and their top contributors and takes as its edges the synaptic connections between them.

Here we should pause to remark on the assumption that only valid cross correlations are allowed. If the input tensor X was padded then the index in the neuron reference would need to reflect the index of the element in X before it was padded. For symmetric p-padding this could be as simple as translating the spatial indices by the tuple $(-p, -p)$.

Tensor slice to tensor slice profile graphs

A similar procedure as above is used when neurons are identified with tensor slices. In the case of 1-dimensional spatial activations, a neuron is referenced by layer and spatial index. For example the neuron corresponding to the spatial activation $Y_{:,i,j}$[4] in layer k would be referenced by $(k, (i, j))$. A spatial activation $Y_{:,i,j}$ is the result of cross correlation of W and a single receptive field $\psi_{i,j}^{(m,k)}(X)$, so the functional relationship between spatial activations is described by the sum

$$Y_{:,i,j} = \sum_{s,t} (W)_{:,:,s,t} \cdot \psi_{i,j}^{(m,k)}(X)_{s,t}. \tag{2}$$

The 1-dimensional summands in this equation are the result of matrix multiplication of a $2d$-slice of W times a $1d$ slice (or spatial activation) in $\psi_{i,j}^{(m,k)}(X)$.

It isn't obvious how to efficiently identify the most contributing neurons in this sum. One way is to eliminate small contributors by focusing on the terms with highest correlation to $Y_{:,i,j}$. Define

$$z_{s,t} = (W)_{:,:,s,t} \cdot \psi_{i,j}^{(m,k)}(X)_{s,t}$$

and order $\{z_{s,t}\}$ in descending order by $\|z_{s,t} \cdot Y_{:,i,j}\|$. Compute the partial sums of the $z_{s,t}$ at each index in the ordering of $z_{s,t}$ and choose the first sum whose distance from $Y_{:,i,j}$ is less than some threshold $\epsilon\|Y_{:,i,j}\|$. The terms $z_{s,t}$ in the chosen partial sum correspond to neurons referenced by the indices $i + s, j + t$. Weight the corresponding edge by its relative contribution using

$$w_{i,j,s,t} = \frac{z_{s,t} \cdot Y_{:,i,j}}{\|Y_{:,i,j}\|^2}.$$

If X was padded a shift in the index reference may be required as it was for the element to element profile graphs.

The case of 2-dimensional channel activations is handled similarly. Here each neuron corresponds to a channel activation and the functional relationship is given by summing over

[4]The colon indicates the full range of indices in that dimension.

the Hadamard products

$$Y_{h,:,:} = \sum_r W_{h,r,:,:} \odot X_{r,:,:}. \tag{3}$$

As before, the nodes in the profile graph will correspond to the index of each channel activations and the edges will be defined by influential contributions between channels of adjacent layers.

2.2 Analysis of Profile Graphs

Jaccard Similarity Metrics

Using the Jaccard index between sets of influential neurons and/or synapses of two profiles, we developed metrics to compute similarity between profiles and perform clustering. The Jaccard index between two sets is a measure of their similarity based on the sizes of their intersection and union:

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{4}$$

We defined three Jaccard-based similarity metrics between profiles. These metrics can be taken with respect to either influential neurons or synapses; from here on we will refer to the influential neurons. For the set of influential neurons P in a profile of a model with layers L, let $P_i \subseteq P$, where $i \in L$ is the subset of influential neurons at layer i. The Jaccard index between sets of influential neurons from two profiles X, Y is the *profile Jaccard*:

$$\text{ProfileJaccard}(X, Y) := \text{Jaccard}(X, Y). \tag{5}$$

The *average Jaccard* between profiles is the average over the Jaccard at each layer:

$$\text{AverageJaccard}(X, Y) := \underset{i \in L}{\text{Ave}}(\text{Jaccard}(X_i, Y_i)). \tag{6}$$

The *instance Jaccard* is the fraction of influential neurons in X which were also found to be influential in Y, typically used for comparing a single-input profile to an aggregated class profile (i.e., $|X| << |Y|$):

$$\text{InstanceJaccard}(X, Y) := \frac{|X \cap Y|}{|X|}. \tag{7}$$

We used the profile Jaccard as a similarity metric to perform k-medoids clustering on aggregated class profiles of all 1000 ImageNet1k classes. Figures 2 and 3 show conceptually coherent clusters, providing further support for profile graphs as an interpretability tool.

Topological Data Analysis

Topological Data Analysis (TDA) is a powerful tool for the analysis of large metrizable spaces of data. We explore the use of TDA to analyze the structure of profile graphs and uncover meaning behind the interconnection of the synapses, independent of labels on nodes and synapses.

In a profile graph, edge weights are defined as a function of the influence weight assigned to the corresponding synapse. To facilitate the construction of a meaningful metric space we have explored two such functions, which we refer to as the original and inverted weighting schemes. Under the original weighting scheme the weight of an edge is equal to the influence weight of its corresponding synapse, so nodes connected by synapses with greater influence weights are

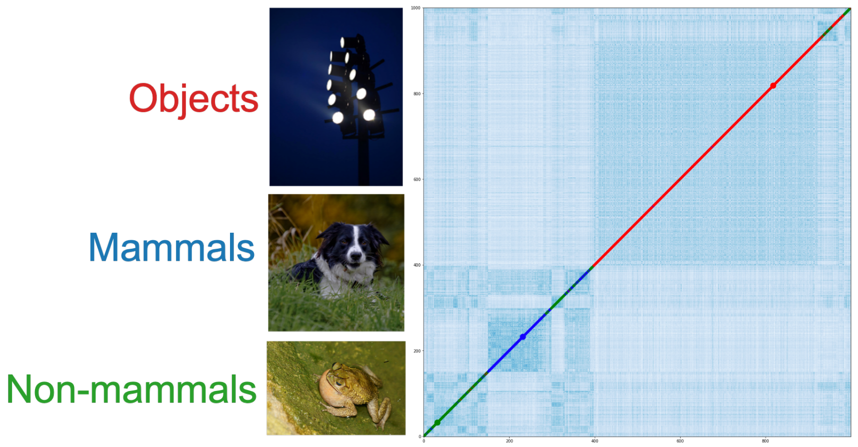
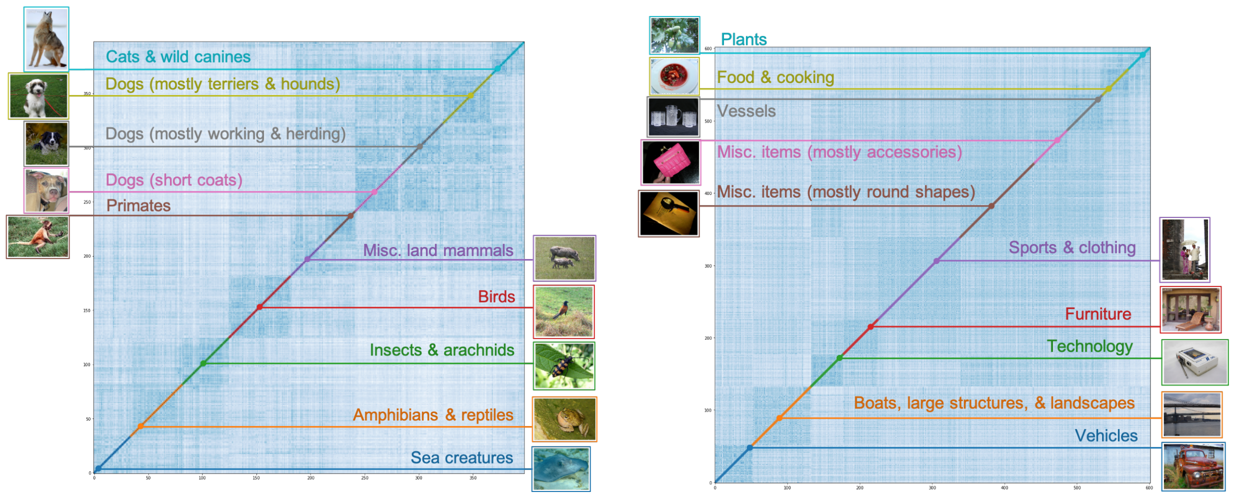


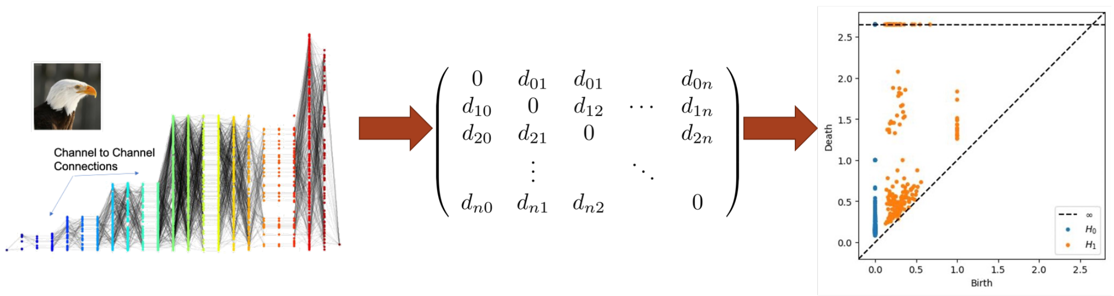
Figure 2. 3-medoids clustering by class profile Jaccard on ImageNet reveals broad groups for objects, mammals, and non-mammals.



(a) 398 animal classes, appear to be clustered roughly by species

(b) 602 object classes, appear to be clustered roughly by features and context

Figure 3. 10-medoids clustering by class profile Jaccard on subsets of ImageNet (animal classes vs object classes)



$$\begin{pmatrix} 0 & d_{01} & d_{01} & \cdots & d_{0n} \\ d_{10} & 0 & d_{12} & & d_{1n} \\ d_{20} & d_{21} & 0 & & d_{2n} \\ & \vdots & & \ddots & \\ d_{n0} & d_{n1} & d_{n2} & & 0 \end{pmatrix}$$

Figure 4. We define a metric space on the vertices of a profile graph, which we can then analyze using persistent homology.

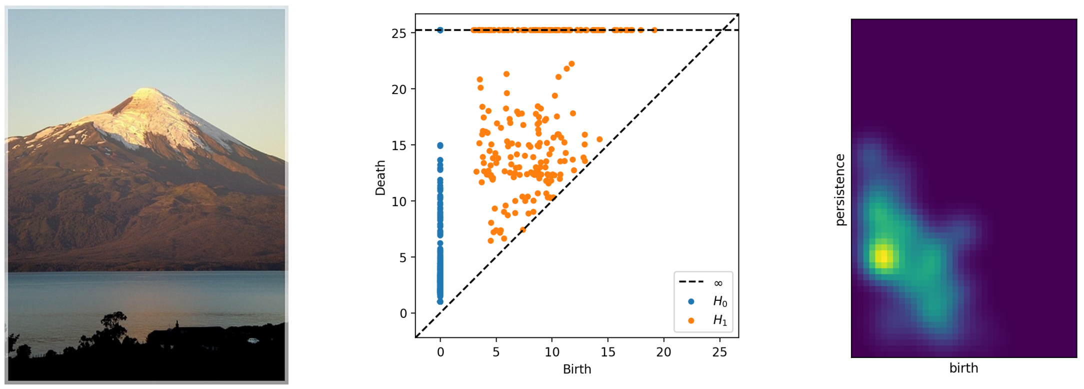
further apart by shortest path distance. We define the inverted weighting scheme to assign a weight of w_i^{-1} to edge i, where w_i is the influence weight of the corresponding synapse i. Profile graphs that use the inverted weighting scheme are appropriate when our method of analyzing the graph places greater importance on points that are close together.

The vertices of the profile graph can be represented in a metric space by constructing the distance matrix using the shortest path distance. Optionally, some kernel function can then be applied to the distances to produce a desired effect on the metric space. One example that we have explored is the Gaussian kernel, given by $g(x) = 1 - e^{-x/2\sigma}$, where σ is the standard deviation of the finite shortest path distances. The Gaussian kernel is an increasing function that spreads out low distances and contracts high distances. When the edge weights of a profile graph are defined according to an inverted weighting scheme, the low distances correspond to the most influential connections. In this case, spreading out the low distances can reveal more nuanced structures that emerge at those distance thresholds.

Persistent homology allows us to summarize the "shape" of profile graph data based on the appearance of topological features at different distance thresholds. We calculate the persistent homology of a metric space, and then study its persistence diagram to identify topological features of the corresponding profile graph. Persistence diagrams allow us to visualize the persistence of features by plotting a point for each topological feature, whose coordinates are (*birth*, *death*). The *birth* of a feature, such as an open loop, represents the distance threshold when the loop was formed, and the *death* represents the distance threshold when the loop was closed or triangulated.

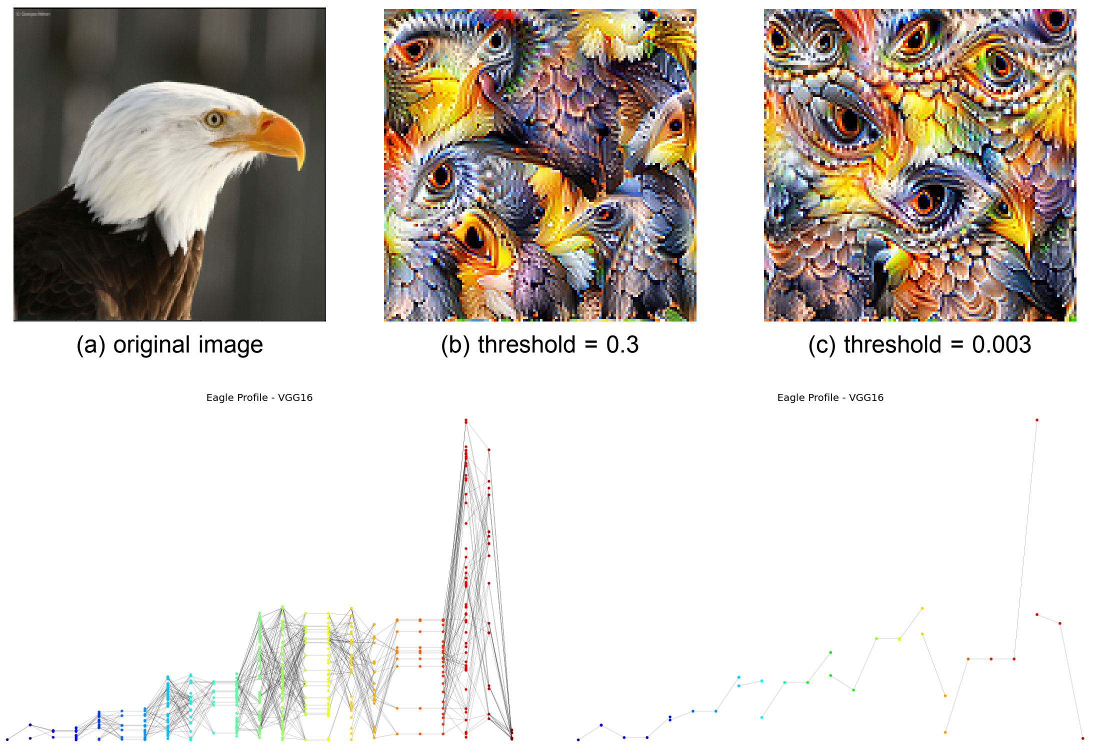
Persistence images are finite-dimensional vector representations of persistence diagrams [34]. We have used persistence images as part of our initial exploration of the topological features of profile graphs, since they provide alternative visualizations that can be compared by Euclidean distances, a metric that is much more computationally efficient than the current standard methods for comparing persistence diagrams.

Our TDA visualization tool allows persistence diagrams and persistence images to be viewed alongside the input image from which their corresponding profile graph was generated by Deep Data Profiler. The tool includes image and persistence data for 50 images from each class of the ImageNet1k dataset, profiled using element-wise and channel-wise neuron definitions, on both VGG-16 and ResNet-18 architectures. All persistence images were generated using the same scale and parameters, so they can be visually compared between different input images and classes.



(a) Class: volcano (b) Persistence diagram (c) Persistence image

Figure 5. Persistent homology for a profile graph (ImageNet on VGG-16).



(a) original image (b) threshold = 0.3 (c) threshold = 0.003

(d) profile for 0.3 threshold (e) profile for 0.003 threshold

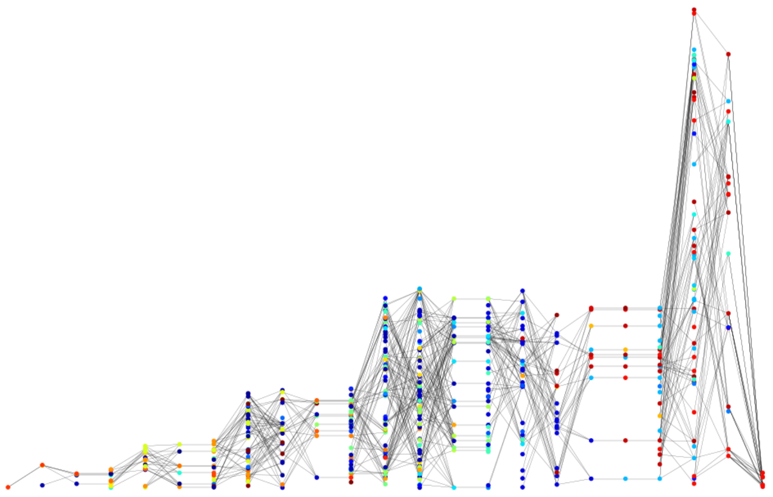
Figure 6. Feature visualizations optimizing neurons in profile graphs for a single image at different thresholds. Graph is pictured aligning neurons by their layer, from early layers on the left to later layers on the right. The example comes from VGG16 trained on Imagenet.

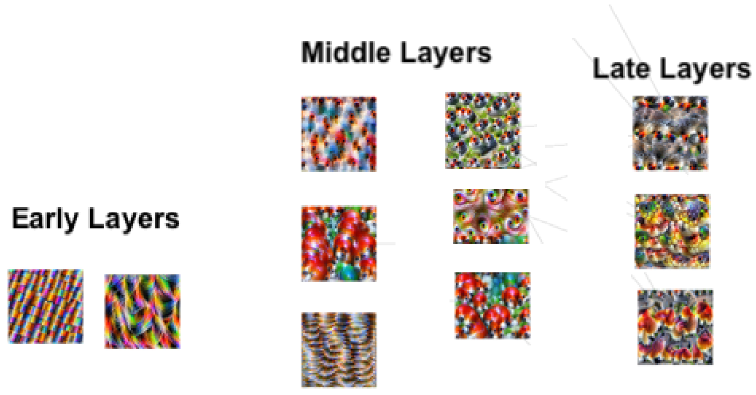
Feature Visualization

Feature Visualization (FV) is an optimization technique for generating images that highly activate a neuron or collection of neurons in a CNN [25, 5]. FV measures a neuron's response to an image, akin to a neurologist measuring a the brain's neural response to stimuli; then it iteratively improves the image to increase the response, thereby creating a superstimulus. FV can be used to establish causal understanding of the neuron(s), because every aspect of a feature visualization is found via the optimization process and is not merely a correlate of the activating characteristic. By coupling FV with the DDP profiles, we look for human interpretable visualizations describing the sequence of classification decisions the network makes. Figure 7 shows feature visualizations optimized for profile graphs. We note along with [5, 26] that interpretation of the images is often difficult in part due to their polysemantic nature of the images; multiple features could stimulate the same set of neurons.

Spectral clustering is a graph theoretic tool for clustering the nodes of a graph [35]. Using the adjacency matrix for a profile graph, we use the corresponding Laplacian matrix and apply k-means clustering on a subset of the Eigen vectors to cluster the nodes across the profile. We apply feature visualization optimizing for each node in a cluster separately and then together as shown in Figure 7.

(a) original image


(b) threshold = 0.5



(c) optimized cluster neurons

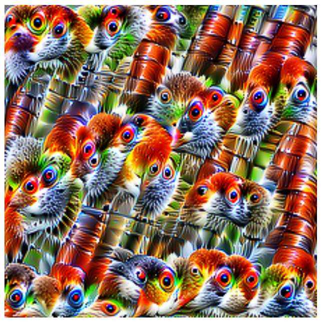

(d) optimized cluster

Figure 7. Nodes from profile graph in Figure 2.2 are clustered into 50 clusters. Choosing one cluster, we optimize an image for each neuron in the cluster noting where they occur in the profile and a single image optimizing for all neurons in the profile.

3.0 Project milestones FY22

At the end of FY21 we shifted our focus away from profiling based on projections of the activations onto the Euclidean basis to projections onto a basis which aligned to the linear dynamics of the weight tensors. This involved developing a meaningful singular value decomposition (SVD) for a matrization of these tensors, profiling tools to identify significant singular vectors, and optimization methods for visualizations. We found the SVD profiles are more meaningful and are clearly linked to the dynamics the model uses to move data around for linear separability. We used hypergraph theory to study the intra-layer relationships between singular vectors and feature visualizations to gain visual interpretations for groups of neurons.

The details of the research for FY22 are described in **The SVD of Convolutional Weights: A CNN Interpretability Framework** [10], but we highlight them here.

3.1 The SVD

When a weight tensor W acts on an input tensor X, it linearly stretches, rotates, and contracts the receptive fields in X giving increased importance to some features and decreased importance to others. The SVD of W exposes the dynamics of this map by identifying directions of stretch, contraction, and rotation. Since the weight 3-tensor acts independently on each receptive field we unfold the weight tensor into a matrix and the input 3-tensor into a matrix with receptive fields in the columns. Convolution is then completely described by matrix multiplication. The output is a matrix of spatial activations, which can then be folded into the 3-tensor normally obtained from the convolution operation. This perspective recognizes that the dynamics of convolution are best studied using the matrix form of a linear transformation from the space of receptive fields to the space of spatial activations. The singular values are indicators of the features most important to the network. These features are discovered by projecting the spatial activations onto the left singular vectors.

The mathematics involved in transforming the weight matrix highlight a distinct difference between our approach and previous work, which focuses on the effect of channels or filters on the full activation space, capturing important elements of the activation tensor by using their norm [25, 36, 37]. Our approach recognizes that while a large activation value means something triggered the model, it doesn't say what, nor how the model responded.

Since convolution acts independently on the receptive fields we look at the linear transformation between the receptive field space of the inputs and the space of spatial activations in the output. By using an SVD to study the transformation we identify the singular vectors most highly correlated to each class. We optimize inputs, which are correlated to each singular vector with the goal of identifying the features in the input domain triggering those vectors.

The result is a hierarchy of the features, which are highly correlated to singular vectors and are ranked by the corresponding singular values. We model this hierarchy using hypergraphs.

3.2 Hypergraph Profiles

Hypergraphs are generalizations of graphs, which model the many-to-many relationships within data. Hypergraphs preserve the important mutual relationships that can be lost in ordinary graphs [38]. A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set of hyperedges $\mathcal{E}$ such that each $e \in \mathcal{E}$ is a subset of $\mathcal{V}$. While graph edges correspond to exactly two nodes,

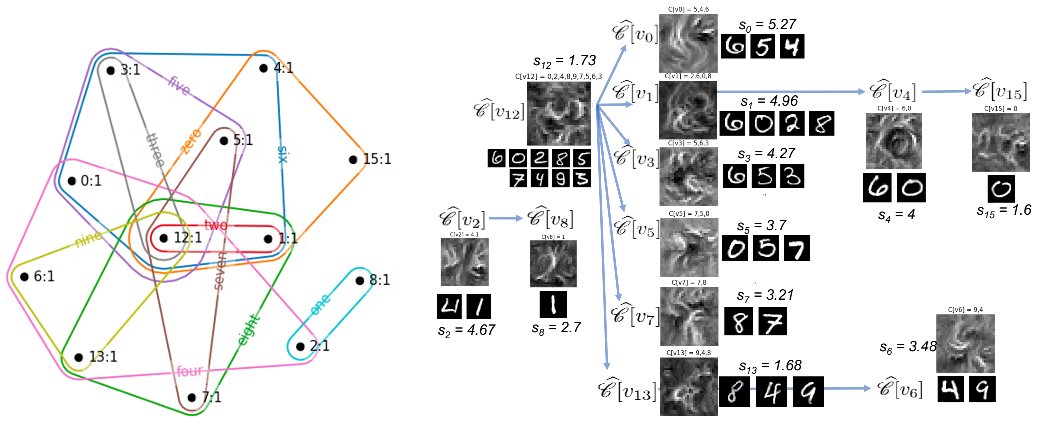


Figure 8. **Illustration of a hypergraph and its semantic hierarchy.** Data comes from a simple CNN trained on MNIST. Nodes in the hypergraph correspond to one or more singular vectors in the first fully connected layer. Visualizations activating each of these vectors are shown the diagram along with exemplary images, which highly activate each singular vector [10].

hyperedges correspond to any number of nodes so that hypergraphs are often thought of as set systems but with more structure [39].

We model the relationships between the target classes and the singular vectors significant for each class using hypergraphs [40]. The size of the hypergraph depends on a threshold used to determine if a particular singular vector has high correlation with the elements of a target class. The diagrams in Figures 8 and 9 illustrate the bifurcation of target classes by the singular vectors most significant for their classifications. The feature visualizations optimize images for each singular vector or a group of singular vectors. Figure 10 illustrates how the singular vectors activated by two dog images differ, possibly due to different perspectives. The feature visualization is for a singular vector common to both images. The overlays highlight receptive fields in the original image whose activations most correlate with this singular vector.

3.3 Application in Nuclear Forensics

We apply our methodology to two models used to classify the attribution of nuclear materials. This use case of machine learning is important, in part, because it helps to determine the provenance of nuclear materials. The synthetic pathway of a nuclear material, for example the precipitating reagants and calcination conditions, can often be accurately inferred by the surface morphology of the material [41]. The scanning electron microscopy (SEM) allows practitioners to acquire images of the surface morphology. DL methods can be accurate and fast, as opposed to labor intensive morphological analysis [42]. We investigate a supervised learning approach [43] and an unsupervised DL approach [44].

Supervised nuclear forensics model

For the supervised model, we use a ResNet-34 network from [43] that accepts SEM images of four different magnifications as input. [43] considers different parameterizations of the weights of the ResNet-34 model with respect to the four inputs. We limit our analysis to the

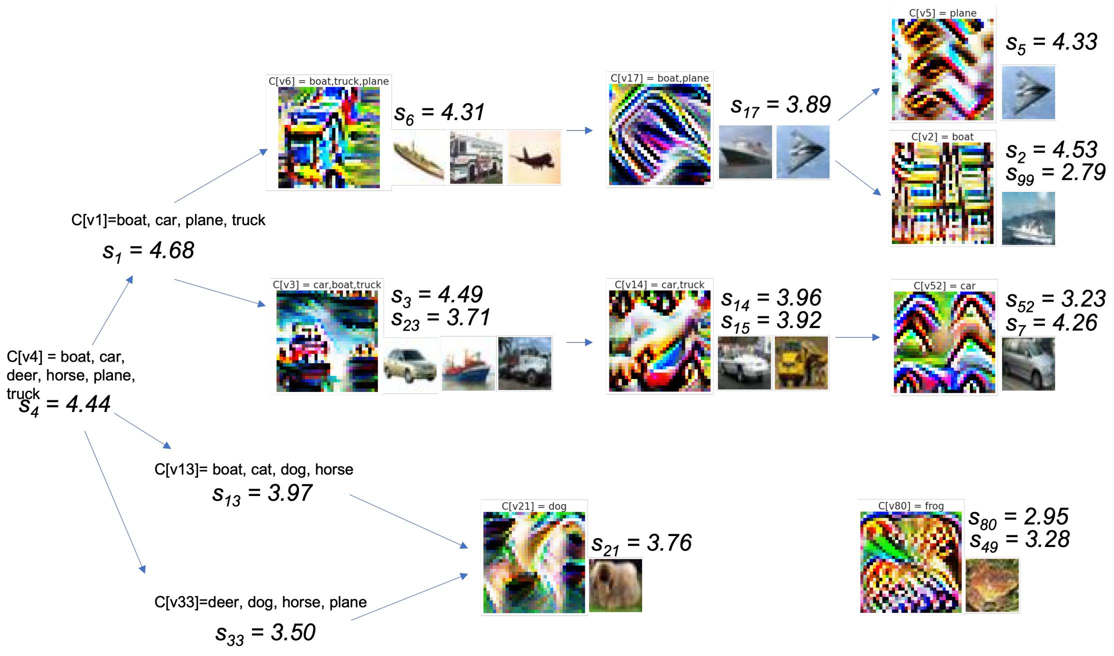


Figure 9. **Exploration of VGG-16 [6] on CIFAR-10 [11]** – Diagram for part of the semantic hierarchy induced by a hypergraph for the 7th convolutional layer [10]

'weight-sharing' parameterization, where the same model is used to process each input magnification in parallel.

We include some exploratory results for the MISO model. In Figure 11, we display a class restricted hypergraph for the final convolutional layer in the ResNet-34. While we calculate the relative singular value importance with the full set of classes, for convenience we restrict our hypergraph analysis to display only 5 classes consisting of the synthetic route to UO$_2$. We focus on signal 16, which discriminates the class $SDU \rightarrow UO_2$ for the layer. Via the hypergraph profiles, we note that positive signals corresponded to negative space for $SDU \rightarrow UO_2$ (as well as $AUC \rightarrow UO_2$) in later layers. This feature was previously suspected in forensics literature [42]. However, images of samples taken in the real world will likely be more dispersed. In these cases, this will fail to be a useful signal.

Unsupervised nuclear forensics model

For the unsupervised model, we study a Vector Quantized Variational Autoencoder (VQ-VAE) [45, 46] from [44]. The VQ-VAE most prominently differs from a standard variational autoencoder in its use of a discrete, rather than continuous, latent bottleneck layer, or 'codebook,' between the encoder and decoder. The models were trained on a dataset of SEM images of materials with different precipitating reagents [47]. The initial unsupervised training for the VQ-VAE used a standard image reconstruction task with this dataset along with VQ-VAE specific loss terms to help with optimization and condition the discrete latent space to be well-behaved.

After the unsupervised training Girard et al. discard the decoder and use the VQ-VAE codebook as a feature vector for a simple supervised classification model. The VQ-VAE encoder is not updated during the further supervsied training. The authors test three models for downstream supervised tasks: a random forest, a support vector machine, and a shallow

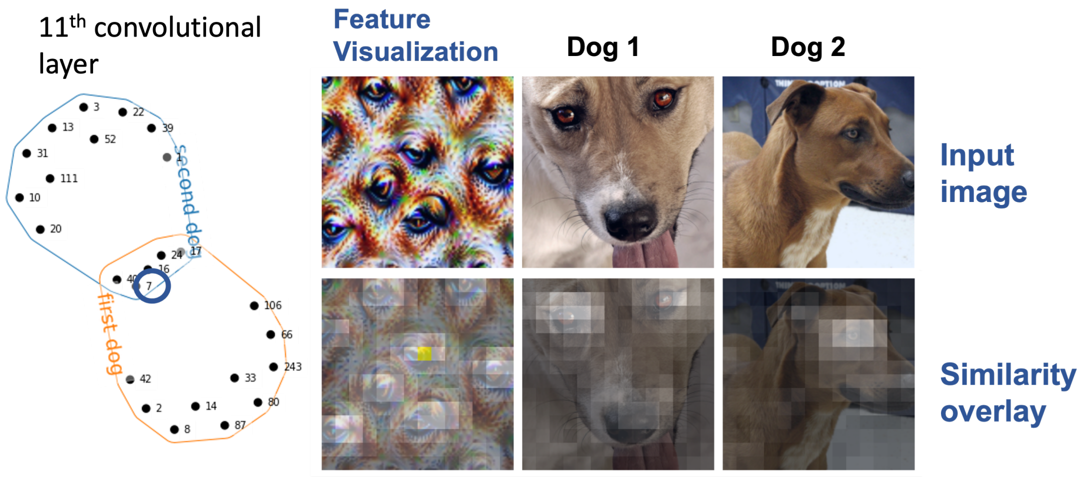


Figure 10. **Exploration of VGG-16 [6] on ImageNet [12] – Significant features shared by two images.** The hypergraph models the relationship between singular vectors highly significant for two dog images from the 11th convolutional layer. Exemplary images and feature visualization are for the singular vector v_7. Similarity overlays highlight cosine similarity between the latent representations of the three images [10]

multi-layer perceptron. All models achieve about the same performance. We focus on the VQ-VAE encoder layers for our hypergraph and feature visualization workflow. We give a result comparing the VQ-VAE to the supervised model MISO model in Figure 12. The full model feature visualizations are created using the methodology shown in Figure 7, now using the singular vector. We found a sharp distinction in the way these respective models classified materials; namely, the VQ-VAE model relied heavily on texture features, whereas the supervised MISO model used large-scale shape structures.

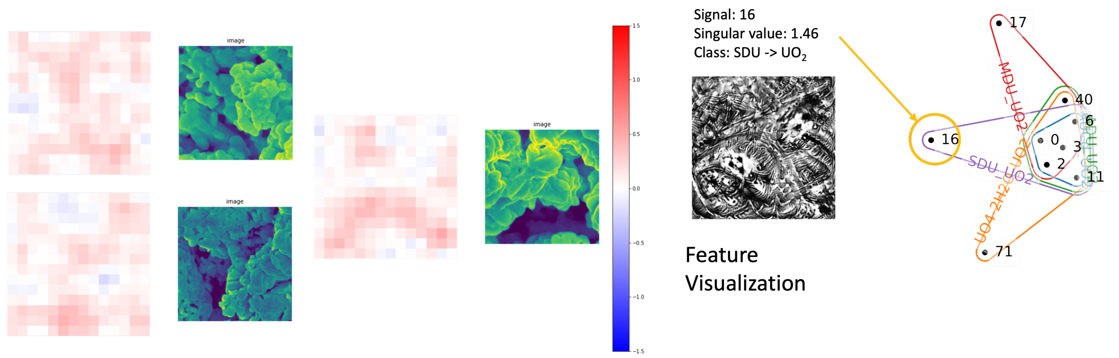


Figure 11. **MISO Signal, (final convolution on 3rd block)**. A feature visualization optimized for singular vector 16 as well as exemplary images included with their projected activations. Positive signals corresponded to negative space for $SDU \rightarrow UO_2$ in later layers.

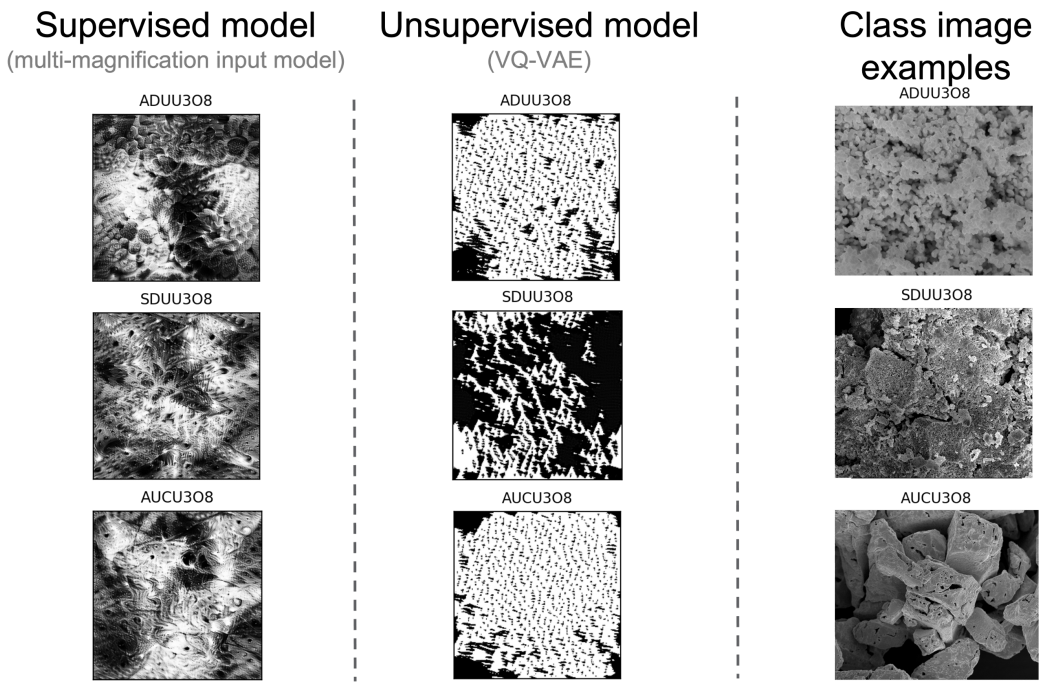


Figure 12. **Comparing the unsupervised VQ-VAE with the supervised MISO model.**

4.0 Conclusion

The DDP project's initial goal was to tackle the problem of CNN interpretability using topological and graph theoretic techniques applied against attribution graphs, which we call profiles. Profiles link neurons and groups of neurons from the activations generated by the CNN. They provide a graph summarizing the decision sequence used by the model for classification.

During FY21 our analysis showed that profile graphs give good class separation and hence provide classification summaries that might be used for anomaly detection. The TDA and graph analysis confirmed the separation and produced some interesting features in the persistence diagrams, but did not provide the hoped-for topological features, which might explain the model's decision process. This led us to question whether the basic neurons we were using for profiling should be the traditional elements, spatial activations, and channel activations used in the literature.

For FY22 we shifted our focus to apply basic linear algebra tools to analyze the linear layers in terms of the dynamics of their map. We used a simple unfolding of the weight tensors into matrices in order to apply SVD theory and study the singular vectors. We discovered a deep relationship between the features persisting through the layers and the singular vectors and modeled these using hypergraphs.

Our research offers a promising new perspective into interpretability theory. By recognizing that convolution as a matrix transformation on the space of receptive fields to the space of spatial activations, we are able to apply traditional numerical linear algebra to understand the stable and unstable subspaces of the feature space. Singular values rank features by how the model responds to their discovery. Optimization techniques link those features to recognizable inputs.

We applied our research to nuclear forensics to interpret the decision process of two models used to classify processing pathways. We were able to distinguish the models by their decision strategy and gained some insights as to what features the models were focusing on.

We are expanding our work into vision and NLP transformer models as we transition from an LDRD project to external funding for FY23.

5.0 Bibliography

*Bibliography

[1] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks, June 2019.

[2] Sören Dittmer, Emily J. King, and Peter Maass. Singular values for relu layers. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3594–3605, 2020.

[3] Michael Mahoney and Charles Martin. Traditional and Heavy Tailed Self Regularization in Neural Network Models. In *International Conference on Machine Learning*, pages 4284–4293. PMLR, May 2019.

[4] Charles H Martin, Tongsu Serena Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):1–13, 2021.

[5] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2(11):e7, November 2017.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[7] B Praggastis, D Brown, and M Shapiro. PNNL DeepDataProfiler. https://github.com/pnnl/DeepDataProfiler, 2020.

[8] Github.com/pnnl/DeepDataProfiler. Pacific Northwest National Laboratory, May 2020.

[9] D Brown, M Shapiro, and B Praggastis. PNNL DeepDataProfiler visualization tool. https://share.streamlit.io/pnnl/deepdataprofiler/frontend/main_streamlit.py. Accessed: 2022-07-07.

[10] Brenda Praggastis, Davis Brown, Carlos Ortiz Marrero, Emilie Purvine, Madelyn Shapiro, and Bei Wang. The svd of convolutional weights: A cnn interpretability framework. 2022.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[15] Cristian Arteaga. LIME Interpretable Machine Learning for Image Classification with LIME. https://towardsdatascience.com/interpretable-machine-learning-for-image-classification-with-lime-ea947e82ca13, October 2019.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier. *arXiv:1602.04938 [cs, stat]*, August 2016.

[17] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*, March 2017.

[18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[20] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.

[21] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[22] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31, 2018.

[23] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[24] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. April 2017.

[25] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The Building Blocks of Interpretability. *Distill*, 3(3):e10, March 2018.

[26] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, March 2020.

[27] Yuxian Qiu, Jingwen Leng, Cong Guo, Quan Chen, Chao Li, Minyi Guo, and Yuhao Zhu. Adversarial Defense Through Network Profiling Based Path Extraction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4772–4781, Long Beach, CA, USA, June 2019. IEEE.

[28]

[29] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[30] Jyh-Jong Wei, Chuang-Jan Chang, Nai-Kuan Chou, and Gwo-Jen Jan. Ecg data compression using truncated singular value decomposition. *IEEE Transactions on Information Technology in Biomedicine*, 5(4):290–299, 2001.

[31] H. Andrews and C. Patterson. Singular value decomposition (svd) image coding. *IEEE Transactions on Communications*, 24(4):425–432, 1976.

[32] D.C. Psichogios and L.H. Ungar. Svd-net: an algorithm that automatically selects network structure. *IEEE Transactions on Neural Networks*, 5(3):513–515, 1994.

[33] Torchvision.

[34] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[35] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[36] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable Basis Decomposition for Visual Explanation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 122–138, Cham, 2018. Springer International Publishing.

[37] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Chau. Summit: Learning Interpretability by Visualizing Activation and Attribution Summarizations. *arXiv:1904.02323 [cs]*, September 2019.

[38] Berge C. and Hypergraphs. *Selected topics in graph theory, 3, 189-206, Academic Press, San Diego, CA*. 1988.

[39] Sinan G Aksoy, Cliff Joslyn, Carlos Ortiz Marrero, Brenda Praggastis, and Emilie Purvine. Hypernetwork science via high-order hypergraph walks. *EPJ Data Science*, 9(1):16, 2020.

[40] B Praggastis, D Arendt, C Joslyn, E Purvine, S Aksoy, and K Monson. PNNL HyperNetX. https://github.com/pnnl/HyperNetX, 2020.

[41] Erich Heinz Pieter Cordfunke and AA Van Der Giessen. Pseudomorphic decomposition of uranium peroxide into uo3. *Journal of Inorganic and Nuclear Chemistry*, 25(5):553–555, 1963.

[42] Ian J Schwerdt, Casey G Hawkins, Bryan Taylor, Alexandria Brenkmann, Sean Martinson, and Luther W McDonald IV. Uranium oxide synthetic pathway discernment through thermal decomposition and morphological analysis. *Radiochimica Acta*, 107(3):193–205, 2019.

[43] Cuong Ly, Clement Vachet, Ian Schwerdt, Erik Abbott, Alexandria Brenkmann, Luther W McDonald, and Tolga Tasdizen. Determining uranium ore concentrates and their calcination products via image classification of multiple magnifications. *Journal of Nuclear Materials*, 533:152082, 2020.

[44] M Girard, A Hagen, I Schwerdt, M Gaumer, L McDonald, N Hodas, and E Jurrus. Uranium oxide synthetic pathway discernment through unsupervised morphological analysis. *Journal of Nuclear Materials*, 552:152983, 2021.

[45] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[47] Ian J Schwerdt, Alexandria Brenkmann, Sean Martinson, Brent D Albrecht, Sean Heffernan, Michael R Klosterman, Trenton Kirkham, Tolga Tasdizen, and Luther W McDonald IV. Nuclear proliferomics: A new field of study to identify signatures of nuclear materials as demonstrated on alpha-uo3. *Talanta*, 186:433–444, 2018.

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7675)

www.pnnl.gov