Discovering Concepts in Learned Representations using Statistical Inference and Interactive Visualization

Adrianna Janik
Kris Sankaran
ada.janik@gmail.com
kris.sankaran@umontreal.ca
Montreal Institute for Learning Algorithms
Montréal, Québec

ABSTRACT

Concept discovery is one of the open problems in the interpretability literature that is important for bridging the gap between non-deep learning experts and model end-users. Among current formulations, concepts defines them by as a direction in a learned representation space. This definition makes it possible to evaluate whether a particular concept significantly influences classification decisions for classes of interest. However, finding relevant concepts is tedious, as representation spaces are high-dimensional and hard to navigate. Current approaches include hand-crafting concept datasets and then converting them to latent space directions; alternatively, the process can be automated by clustering the latent space. In this study, we offer another two approaches to guide user discovery of meaningful concepts, one based on multiple hypothesis testing, and another on interactive visualization. We explore the potential value and limitations of these approaches through simulation experiments and an demo visual interface to real data. Overall, we find that these techniques offer a promising strategy for discovering relevant concepts in settings where users do not have predefined descriptions of them, but without completely automating the process.

CCS CONCEPTS

• **Mathematics of computing** → **Hypothesis testing and confidence interval computation**; • **Human-centered computing** → **Information visualization**; • **Computing methodologies** → Knowledge representation and reasoning.

KEYWORDS

interpretability, concepts, visualization, multiple hypothesis testing, neural networks

ACM Reference Format:
Adrianna Janik and Kris Sankaran. 2019. Discovering Concepts in Learned Representations using Statistical Inference and Interactive Visualization. In *Proceedings of ACM Conference (Conference'19).* ACM, Anchorage, AK, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'19, August 2019, Anchorage, AK, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Automatic representation learning can be both a blessing and a curse. On the one hand, it allows practitioners to achieve state-of-the-art performance on a variety of prediction tasks using a generic workflow, without having to hand-engineer features from unstructured input, and it lies at the core of almost all modern AI systems [1]. On the other, the resulting features can be difficult to characterize, either intrinsically, or through their influence on downstream predictions. Mitigating the risk that comes from deploying inscrutable systems in socially relevant settings has emerged as a research priority in the machine learning community [3, 7].

Kim et al. [6] have proposed one way forwards, formally defining activation scores for user-defined concepts. Their method, Testing with Concept Activation Vectors (TCAV), gives a score for how concepts – practically, a collection of related images – influence predictions. For example, a user trying to understand a model for zebra classification can provide a "stripes" concept by collecting many images of stripes, and TCAV will generate a score describing the relevance of that concept. This offers a degree of agency in model inspection not available in more traditional interpretability techniques. The approach takes as its starting point an active and curious user, willing to dedicate time to model inspection.

Our work takes this perspective one step further – helping users become data detectives – in order to address one of TCAV's limitations, the need to collect images from which concepts can be defined. Anything that can facilitate users' interrogation of concepts has the potential to make the method more widely applicable. In its focus on weakening the requirement for preset concepts, our work is similar to [5]. However, rather than automating concept discovery, we aim instead to streamline it from the user's perspective, borrowing techniques from statistical inference and interactive visualization [4, 8]. We have two main ideas,

- Multiple hypothesis testing: Rather than testing one well-defined concept-related hypothesis at a time, we can sift through many (usually uninteresting) candidates in search of a few promising ones, controlling for the risk that indiscriminate searches produce false positives.
- Interactive visualization: A visual interface can help users rapidly cycle through many concepts, offering feedback about queries on the fly, and shaping intuition about relevant concepts in the process.

Both approaches suggest potential concepts in reference to existing samples, rather than those collected explicitly for the purpose of concept evaluation. The distinction is analogous to the one between

exploratory and confirmatory data analysis: the strategy outlined in this work allows weak identification across a range of potential concepts, but not explicit verification of one defined in advance.

Our main contributions area

- In Section 2.1, we develop a proposal for discovering concepts using multiple hypothesis testing.
- In Section 2.2, we describe a visual interface through which users can interact with concept activation vectors.
- In Sections 3.1 and 3.2, we experiment with and critique these proposals. For multiple hypothesis testing, we leverage an illustrative simulated dataset, while for interactivity, we consider a real-world computer vision task.

Further, our study takes a geometric perspective of concept activation vectors; clarifying this interpretation is one of the implicit goals of this work.

2 METHODOLOGY

Before detailing our proposals, we summarize the approach of Kim et al. [6]. Say that there are n samples x_i, each with corresponding label y_i. We have a classifier f, mapping samples to predicted classes. f can be an arbitrary black box, with the exceptions that (1) we must be able to extract automatically learned features corresponding to any input, and (2) it must produce predicted probabilities $p_k(x_i)$ for each class k.

TCAV looks at how perturbations in the learned features – call them z_i – affect downstream predictions. The perturbation itself is derived from a user-specified collection of images, $\left(x_i^C\right)$, which capture some concept C. Specifically, each x_i^C is featurized as z_i^C, and a classifier is trained to linearly separate the z_i^C from random z_i. The direction v orthogonal to the separating hyperplane is used to summarize the concept C. By changing z_i to $z_i + \epsilon v_i$ for some small ϵ, we imagine we are perturbing sample i slightly "towards" the concept C.

To measure the relevance of a concept with respect to the K classes of the response, we can calculate $S_v(x_i) = Dp(z_i)v$, where $Dp(z)$ is the $K \times J$ dimensional Jacobian, whose kj^{th} entry, $\frac{\partial p_k(z)}{\partial z_j}$ describes the sensitivity of the k^{th} class's probability with respect to the j^{th} learned feature, evaluated at the point z. This is a directional derivative of the class probabilities in the direction v. TCAV quantifies the significance of the concept encoded by v with respect to the class k by seeing whether the fraction of x_i from that class with positive directional derivative is larger than would be expected when defining random concepts, where a random concept is one defined by a random subset of the x_i. Formally, the relevance of concept v for class k is given by the score

$$TCAV_k(v) = \frac{|\{x : S_{vk}(x) > 0 \text{ and } y_i = k\}|}{N_k} \tag{1}$$

where N_k is the number of samples in class k and $S_{vk}(x)$ is the k^{th} coordinate of $S_v(x)$.

2.1 Multiple Hypothesis Testing

Multiple hypothesis testing is widely used in genetics, and we motivate its application to the problem of concept discovery by drawing parallels to that field. In genetics, when researchers want to know which genes are associated with a particular disease, they perform a hypothesis tests for each of possibly thousands of candidates, with the expectation that almost all will be irrelevant. However, a few may emerge as potential signal within the noise, and using an appropriate statistical correction, it is possible to guarantee that the proportion of false discoveries among these selected candidates lies below some fraction [4].

In the concept discovery setting, we screen for concept vectors v in the way that a scientist screens for disease causing genes. To implement the idea, we need to specify two things,

- A way of proposing concept directions to screen over.
- A way of evaluating the significance of candidate concepts.

To propose concept directions, we simply draw J random directions on the unit sphere in the learned feature space. This has the disadvantage of undersampling directions with high densities of z_i, but appears sufficient in our preliminary experiments.

Given this collection of directions, we could evaluate their individual significance using a randomization test, as in [6]. Rather than selecting random sets $\left(x_i^C\right)$ and fitting separating hyperplanes, we pick J' random directions v_j^* as above. This has the benefit of not having to fit a separate hyperplane for each draw from the randomization null, but again undersamples dense regions.

Given these reference v_j^*, a p-value can be obtained as[1] $p_j = \frac{1}{J'}\left|\left\{TCAV\left(v_j^*\right) \geq TCAV\left(v_j\right)\right\}\right|$, the tail area in the null reference distribution of TCAV scores above the observed score for the candidate concept v_j of interest. These p-values need to be adjusted in order to guarantee proper statistical inference. To see why, note that in the completely null case, where the p-values are uniformly distributed, an α significance threshold for individual hypotheses will lead to a fraction α of the v_j's being declared interesting, when really none of them are: 100% of the discoveries are false discoveries. The usual terminology is that the false discovery rate (FDR, the proportion of rejected hypotheses that were in fact null) is 1.

To address this, we can use the Benjamini-Hochberg procedure: apply more stringent thresholds, depending on strength of the signal [4]. Specifically, sort the p-values, so that $p_{(j)}$ is the j^{th} smallest of the original J. Instead of comparing $p_{(j)} \leq \alpha$, compare $p_{(j)} \leq \frac{j\alpha}{J} := \alpha_j$. Reject the hypotheses corresponding to $p_{(1)}, \ldots, p_{(j^*)}$, where j^* is last time the inequality $p_{(j)} \leq \alpha_j$ holds. This is guaranteed to keep the proportion of "false discovery" directions v_j below α.

This approach is the most natural one, but it suffers two limitations. First, it requires a randomization test for each candidate. This is computationally expensive, since samples cannot be reused – this would violate the independence assumption required to guarantee FDR control. Second, we observe that the randomization p-values using the statistic 1 can be degenerate. For some directions, all $S_{vk}(x_i)$ are positive across all samples in class k, so that the corresponding p-value must be zero.

We can address the first issue by using local FDR (lFDR) methods [4]. Rather than starting with p-values, these methods operate directly on test statistics. By assuming that most hypotheses are null,

[1] For clarity, we omit the class k from our notation, but all these calculations must be done with respect to some set class.

it is able to estimate a reference null distribution. By comparing this reference with the original test statistics, a local FDR can be approximated for any value of the test statistic. Those in the bulk of the reference null have a high lFDR – rejecting these would lead to a high fraction of false discoveries – while those in the tails have a low lFDR. Rejecting all points with lFDR $\leq \alpha$ is guaranteed to control the FDR below α.

To deal with the second issue, were consider test statistics that account for the size of the score $S_v(x_i)$, not simply the sign. This possibility is noted by [6], but not pursued there. In our experiments, we find that $\sqrt{\text{Var}_{\hat{\mathbb{P}}_n}(S_v(x_i))}$, the empirical standard deviation of concept activations, has a distribution more amenable to lFDR estimation.

Aside from the main thread of concept discovery, we note that it is possible to adapt the multiple testing view to the related problem of screening for subregions of the feature space for which a prespecified concept direction is relevant. We can cluster samples, evaluate concept activation scores within each, and compare to reference distributions. This can be used to identify samples along decision boundaries, where class probabilities are changing rapidly.

2.2 Interactive Exploration

Concepts are directions in the learned features space. By making it possible for users to interact with this space, we can facilitate concept discovery. There are two challenges in implementing this idea. First, the dimensionality of the learned space is typically very large. Second, users may become annoyed with the number of interactions necessary to find and understand interesting, tangible concepts. To address these issues, we appeal to techniques from the interactive data visualization literature: dimensionality reduction and linking [2].

We propose the view in Figure 1, with functionality outlined in Figure 3. Each point in the scatterplot corresponds to one sample in the original data. Their positions are obtained by plotting with respect to the directions derived from a Principal Components Analysis (PCA) on the learned features. This reduction is how we manage the high-dimensionality of the z_i; alternative reductions could be used. The central black arrow represents a user-defined concept direction v, and points of the currently selected class k are enlarged and colored according to the class' score, $TCAV_k(v)$. This immediate visual feedback on the relevance of a concept direction is how we make it easier to discover interesting concepts. To provide context about the relevant directions, we display representative images along the concept direction, between the dashed helper lines defined by a user-chosen angle in the bottom left corner. The left panel lets users switch between classes. The overall procedure is summarized in Figure 2 – the elements requiring user interaction are highlighted in yellow.

Working in lower dimensions has its cost, it is an approximation. To calculate the TCAV score, we need to obtain a directional derivative between the concept vector and the gradient of logit function with respect learned features. For the type of high-dimensional z_i learned by modern deep learning models, performing this computation with the full high-dimensional gradient for each point breaks the system's responsiveness. Instead, we again apply dimensionality reduction, this time to the gradients. This trick allows us to

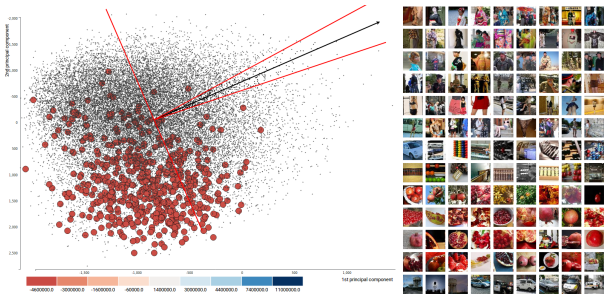


Figure 1: A screenshot from our demo, presenting a proof-of-concept for interactive concept discovery, available at https://adrijanik.github.io/concepts-vis/

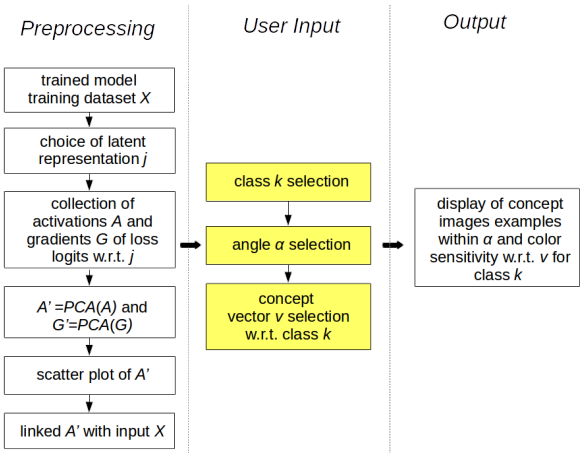


Figure 2: An overview of the procedure for interactively exploring concepts. The first column represents steps before interaction, the second column presents user interaction possibilities, and the third describes the expected output.

store only projection of the gradients, dramatically decreasing the computational load.

The score obtained from the directional derivative is plotted in color space on samples from the selected class. In this way, users can quickly assess which directions are worth exploring further. Instead of staying in the projected space we could perform the reverse operation to obtain an approximation of the vector in the original feature space. In the case of PCA, we use the following procedure: let X be the original $m \times n$ input matrix, μ be the mean vector of X and let V be the $n \times k$ matrix of eigenvectors with highest eigenvalues used to obtain the PCA scores Z. Then, the reconstruction in the original space is $\hat{X} = ZV^T + \mu$. While this is possible, we find that it is still too slow for interactive visualization.

To further streamline the search, we can imagine a few modifications. To suggest directions worth exploring, we could precompute activation summary statistics across directions, and provide reference to them on the plot. We could show the loss evaluated on the projected space as a heatmap, highlighting ridges at which class

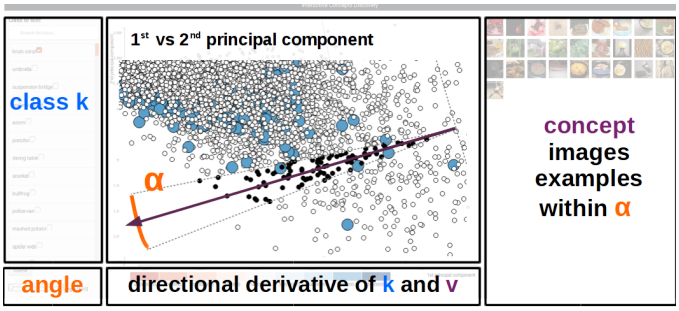


Figure 3: An outline of the key elements of our visualization interface.

predictions change. We could also sort classes so that those with similar active concept vectors are placed nearer to one another.

3 EXPERIMENTS

We now present experiments to evaluate our proposed approaches, with the goal of discovering strengths and clarifying weaknesses.

3.1 Multiple Testing Illustration

We first study the behavior of the multiple testing proposal in a transparent simulation setting. The data are generated as follows. Draw 2000 points uniformly in the square $[-1,1] \times [-1,1]$. Define three classes (see the background in Figure 5),

- Class 0 are points in $\{(x_1, x_2) : x_2 < 0 \cap \|x\|_2 > 0.25\}$.
- Class 1 are points in $\{(x_1, x_2) : \|x\|_2 \leq 0.25\}$.
- Class 2 are points in $\{(x_1, x_2) : x_2 \geq 0 \cap \|x\|_2 > 0.25\}$.

The interesting characteristic of this dataset is that a large portion of the decision boundaries for classes 0 and 2 is linear, while for class 1, the boundary is a circle. Intuitively, the vertical direction is more important for defining classes 0 and 2 than it is for class 1.

Our "black box" classifier is in fact a simple 1-layer MLP. That is, the probability that x belongs to class k is approximated by

$$p_k(x) = \left[\sigma\left(W^{(2)} z(x) + b^{(2)}\right)\right]_k$$
$$z(x) = \text{ReLU}\left(W^{(1)} x + b^{(1)}\right),$$

where σ is the K-dimensional softmax and ReLU denotes the function $x \circ \mathbb{I}(x > 0)$. We use only 20 hidden units $z(x)$ – this is enough to reach essentially 0 training loss. This model is convenient for our study of interpreting automatically learned features because its learned mapping z has a simple description: each coordinate $z_{ik}(x_i) = \text{ReLU}\left(w_k^{(1)T} x_i + b_k^{(1)}\right)$ activates when x_i lands in the half-space orthogonal to $w_k^{(1)}$, with larger activations further in. By mixing these activating hyperplanes, the MLP can trace out the curve defining the true boundary. See Supplementary Figure **??**.

Given this setup, we can now ask,

- What concepts are discovered by the multiple testing procedure?
- For any specific direction, which clusters of points are most strongly activated?

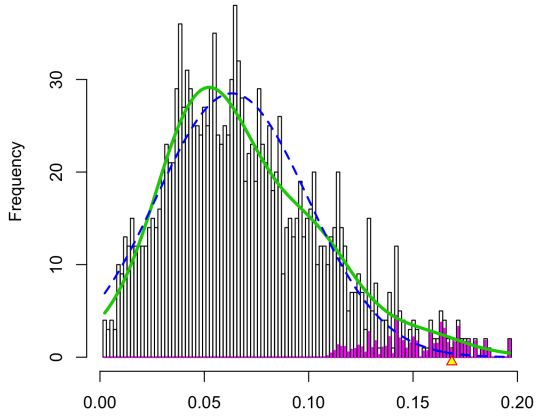


Figure 4: The Local FDR procedure fits the blue distribution as a reference null, which leads to the pink directions being assigned low lFDRs and flagged as interesting.

Considering the variants of the multiple hypothesis testing approach presented in Section 2.1, we first describe our specific implementation[2]. We draw 500 candidate concepts v_j on the 20-dimensional unit sphere in the learned feature space. For fixed v_j and class k, we evaluate $S_{v_j k}(x_i)$ over all 2000 x_i.

As a measure of interestingness, we compute the standard deviation $\hat{\sigma}_{jk}$ of the collection $\left(S_{v_j k}(x_i)\right)_{i=1}^{2000}$. We use the local FDR procedure applied to these $\hat{\sigma}_j$ to assign an lFDR scores to every candidate concept. The reference null and declared discoveries are displayed in Figure 4.

Since our simulation data are low-dimensional, we can visualize the discovered concepts, though only indirectly. To do so, recall that the k^{th} learned feature corresponds to an activation in the direction orthogonal to $w_k^{(1)}$. Identify this direction with the j^{th} standard basis vectors, e_j, in the learned feature space. Formally, the duality comes from $W^{(1)T} e_j = w_j$; informally, moving all points in the w_j direction in the x-space increases the activation of the k^{th} learned features.

In this light, it is natural to associated the direction $W^{(1)T} v_j$ in the x-space with the candidate concept v_j. These directions, normalized to length 1, are displayed in Figure 5, shaded in according to the lFDR of the associated direction, and overlaid on the original simulation data. Evidently, the relevant directions for classes 0 and 2 generally lie orthogonal to the x-axis. On the other hand, class 1 has low lFDR concept directions that make relatively small angles with the x-axis. The fact that the discovered directions correspond to those defining the decision boundaries relevant to each class provides qualitative validation of our approach, though our setting is admittedly a simple one.

We next consider the distribution of $S_v(x_i)$ when a specific direction is chosen, with the goal of identifying strongly activated subsets of points. In a real application, this might provide a more granular view into concept activations than the class-level summary statistic in equation 1. To this end, we set the concept to $v = e_1$,

[2]All code for this study are available at https://github.com/adrijanik/concepts-exploration, see Appendix A for more on reproducing experiments

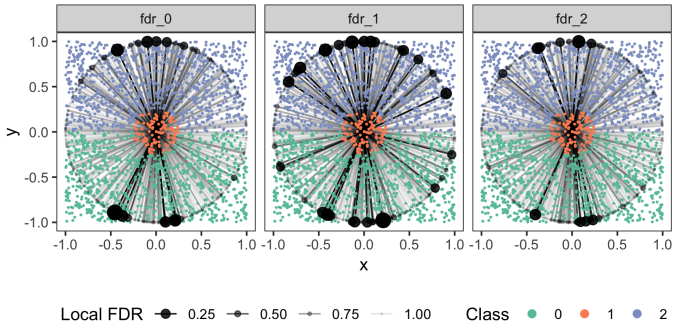


Figure 5: Class 1 has discovered (low lFDR) concepts in a variety of directions, while the low lFDR directions for classes 0 and 2 are generally oriented vertically.

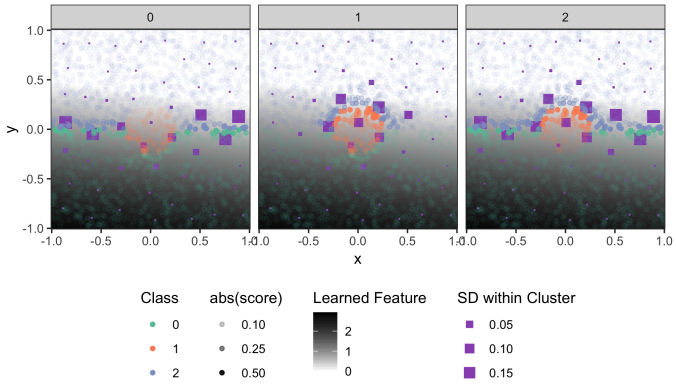


Figure 6: Concepts activations tend to have high variation near decision boundaries.

corresponding to the first learned feature dimension. We cluster all samples in the original x-space using k-means and compute the standard deviation of activation scores within each cluster. The result is displayed in Figure 6.

Each purple square corresponds to one cluster centroid. Their sizes reflects the standard deviation of activations within that cluster, where activations are computed with respect to the class k indicated by the subpanel titles. The absolute value of the original per-point activation scores is indicated by the transparency of the points. The value of the learned feature $z_1(x)$ is displayed as the grey plane in the background.

Clusters with the largest variation in concept activation for this direction tend to appear near the decision boundaries for the corresopnding classes. Note further that parts of the boundary that are far from the learned feature's defining hyperplane tend to have smaller variation – this is visible in the U-shaped part of the boundaries for the first and second classes, for example. In Supplementary Figure **??**, we further find that activations are generally positive for classes 0 and 1 and negative for class 2. This is consistent with the downward orientation of the feature under study.

If this pattern generalizes outside of this simulation experiment, then this approach to summarizing clusters by concept activation

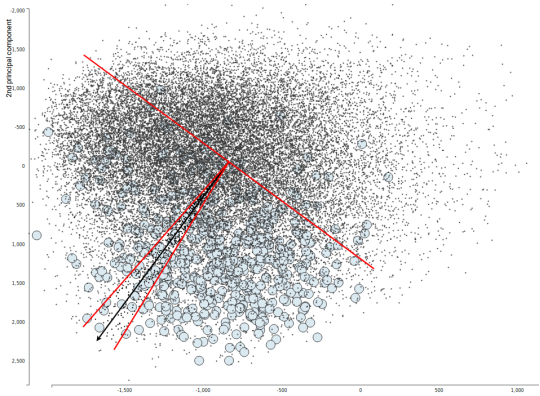


Figure 7: Concept with a high score for class "flagpole" (blue color), suggests that this concept is relevant for tested class.

can be used to sample points near decision boundaries where specific concepts – user-defined or statistically discovered – are informative.

3.2 Interactive Visualization

For visualization, we experiment with a real-world image classification task. We inspect a GoogleNet network trained on ImageNet images. For the purpose of visualization we chose the 50 first classes from the 200 classes in tiny ImageNet. For the selected classes we obtain feature activations and gradients at the `mixed4d` layer. Our setup mirrors that in the experiments of Kim et al. [6]. For each class we have 500 images; in total, our dataset has 25000 images.

In Figure 7, we see an example of a user chosen concept vector pointing towards samples from the selected class "flagpole." This is manifested by changes in the colors of points from that class – points become dark blue when the user's concept vector leads to positive activation scores. In contrast, we can identify a direction that triggers negative scores, marked as red and presented in Figure 8. Example concepts images can be found in Figure 9.

It is worth questioning whether the dimensionality-reduced view is a reliable representation of original space. For this purpose, we look at the variance ratio explained by selected components. With a complex network such as GoogleNet, the activation space has dimension of $19 \times 19 \times 528$ for each analyzed image. After flattening, this gives an activation vector of size 190, 608, and when we reduce this dimension with PCA, variance is distributed across many components (see Figure 10) with first one having around 2% of variance explained.

Visualizing CAVs in the lower dimension is an approximate approach but it may be useful. Navigating through latent space with hundred of thousands dimension is not an easy task and this initial experiments showed us that we may need to visualize more principal components in coordinated views or try our method with a smaller networks, especially promising may be networks with encoder-decoder structure due to their architecture and clear bottleneck layer.

We have potential ideas for future experiments we especially find promising plotting pairs of principal components in coordinated

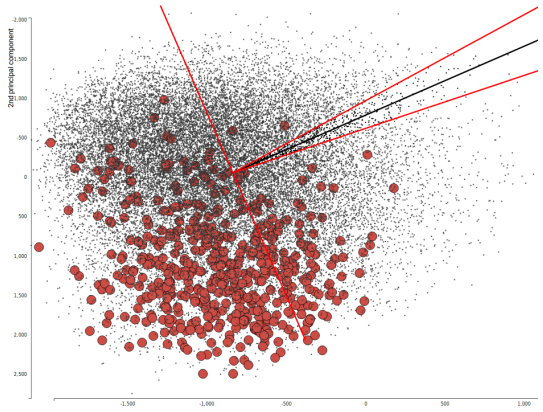
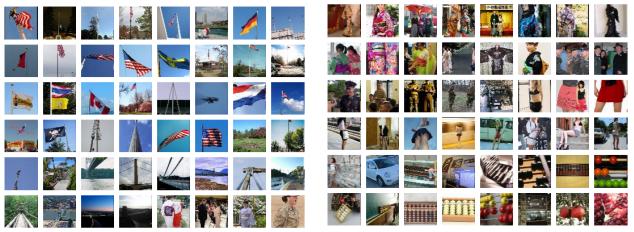


Figure 8: Concept with a very low score for class "flagpole" (red color), suggests that this concept is irrelevant for tested class.



(a) Relevant concept images for "flagpole" class.

(b) Irrelevant concept images for "flagpole" class.

Figure 9: Images examples of relevant and irrelevant concepts from Figures 7 and 8. We can see that among conceptually similar images to flagpole are masts and bridge support pillars, irrelevant images on the other hand seems not alike at all.

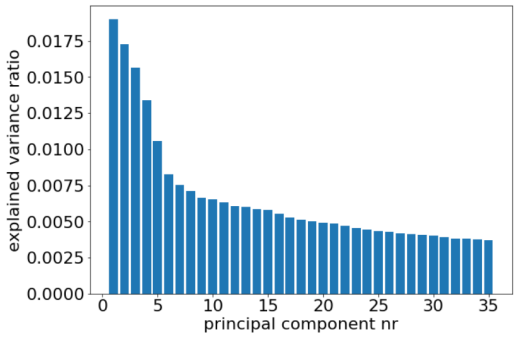


Figure 10: Bar chart of explained variance ratio for 35 first principal components. We can see that majority of components explain very little variance, but also first components variance ratio is below 2% which rises concerns about plotting only first two components.

views to see how concept directions affect other sources of variance and also trying different methods of dimensionality reduction especially non-linear ones.

4 DISCUSSION

Concept discovery can be guided by geometry, statistical inference, and visualization. Inspired by [6], we have developed strategies that let practitioners interrogate the decision surfaces of deep learning models and the way they relate to automatically learned features. We intend for this approach to be a compromise between the manual concept definition of [6] and the full automation of [5].

There are important challenges that must be overcome before either inferential or interactive concept exploration can become commonplace. The first challenge we highlight is that there are likely more efficient strategies for sampling candidate concepts v_j in the learned feature space. It seems natural to place more mass on directions containing more observations. Second, there is the challenge of choosing an measure of concept interestingness which is amenable to local FDR methodology, and a systematic study of potential test statistics beyond those described here would be valuable. Third, our interactive visualization approach is limited by difficulties of visualization high-dimensional functions. Further experiments – perhaps using parallel coordinates or linked views [2] – may yield improvements.

Nonetheless, our experiments point to the value of model interpretability workflows that equip users with all the tools of modern statistics and visualization. We have provided experiments that demonstrate how multiple hypothesis testing can highlight relevant concepts to follow-up on even when none are provided upfront. Further, we have highlighted how interactive views that link concrete image displays with abstract class summaries can engage users in the process of concept discovery. We hope that these techniques play a role in advancing model interpretability as a scientific enterprise, providing powerful tools to users curious about the inner workings of complex models.

REFERENCES

[1] Yoshua Bengio, Aaron C Courville, and Pascal Vincent. 2012. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538* 1 (2012), 2012.

[2] Andreas Buja, Dianne Cook, and Deborah F. Swayne Research Scientist. 1996. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics* 5, 1 (1996), 78–99. https://doi.org/10.1080/10618600.1996.10474696 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/10618600.1996.10474696

[3] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. 2017. Accountability of AI under the law: The role of explanation. *arXiv preprint arXiv:1711.01134* (2017).

[4] Bradley Efron. 2010. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction.* Cambridge University Press. https://doi.org/10.1017/CBO9780511761362

[5] Amirata Ghorbani, James Wexler, and Been Kim. 2019. Automating Interpretability: Discovering and Testing Visual Concepts Learned by Neural Networks. (2019). arXiv:1902.03129 http://arxiv.org/abs/1902.03129

[6] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2017. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *arXiv:1711.11279 [stat]* (Nov. 2017). http://arxiv.org/abs/1711.11279 arXiv: 1711.11279.

[7] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* (2017).

[8] Hadley Wickham, Dianne Cook, and Heike Hofmann. 2015. Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8, 4 (2015), 203–225.

A REPRODUCIBILITY

Demo was developed in JavaScript with the usage of d3.js visualization library. Demo is available online under following link: https://adrijanik.github.io/concepts-vis/. Network that was been used is GoogleNet that was trained on ImageNet and used in the TCAV implementation tutorial by tensorflow. As the original training set is huge, we limited this study to a smaller dataset tiny-ImageNet-200 https://tiny-imagenet.herokuapp.com/ with 200 classes from original set.

Pre-processing part of data was done using Python 3.6 with implementation of PCA from scikit-learn package. PCA was trained on 600 samples that contained 3 samples from each class drawn randomly from the dataset.

The source code can be found in the following GitHub repository: https://github.com/adrijanik/concepts-exploration.

B SUPPLEMENTARY FIGURES

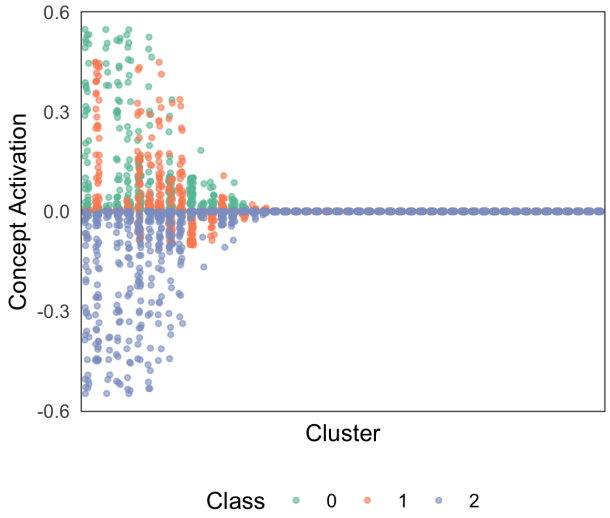


Figure 11: We can identify subsets of samples with widely varying concept activation scores – these tend to occur near decision boundaries. Each column of points corresponds to one cluster. Clusters have been sorted from those with largest variation in concept activation scores, to those with the least. Individual points are samples lying within those clusters, their vertical position is their score $S_k(v)$ for the v presented in Figure 6. Activations for different classes are colored separately. Evidently, in this concept direction, the probability surface between classes 0 and 1 is steeper than that between 1 and 2.

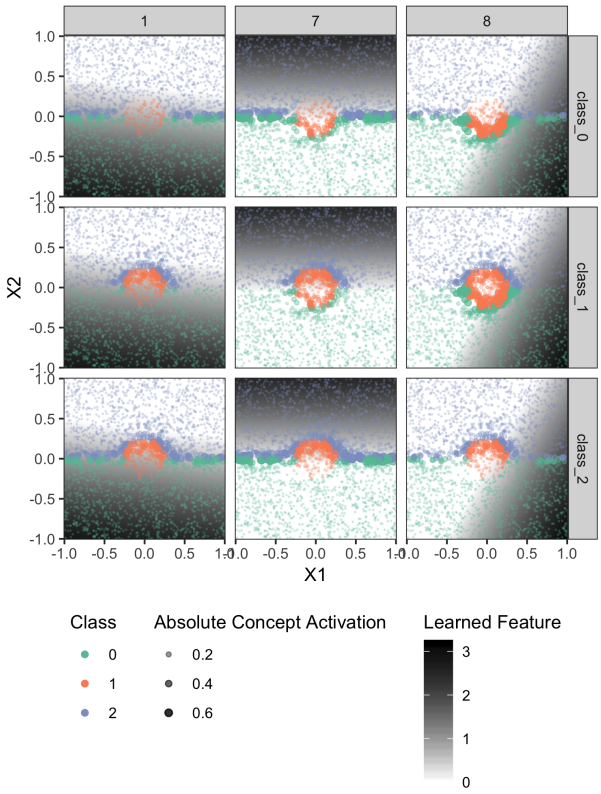


Figure 12: Simulation data overlaid with three learned features and concept activation scores. Each panel column corresponds to one of 20 learned features, which are represented through their activations as black and white half-spaces. Each panel row corresponds to a class on which the activation scores $S_{vk}(x)$ are displayed, where the direction $v = e_j$ for the associated feature coordinate z_j. The size and transparency of the points reflects their absolute concept score. Note that concept scores tend to be highest near the relevant class boundaries.