Extracting Interpretable Concept-Based Decision Trees from CNNs

Conner Chyung [1] **Michael Tsang** [1] **Yan Liu** [1]

Abstract

In an attempt to gather a deeper understanding of how convolutional neural networks (CNNs) reason about human-understandable concepts, we present a method to infer labeled concept data from hidden layer activations and interpret the concepts through a shallow decision tree.

The decision tree can provide information about which concepts a model deems important, as well as provide an understanding how the concepts interact with each other.

Experiments demonstrate that the extracted decision tree is capable of accurately representing the original CNN's classifications at low tree depths, thus encouraging human-in-the-loop understanding of discriminative concepts.

1. Introduction

It is generally understood that Convolutional Neural Networks learn abstract, semantic concepts, but there is still an ongoing question about how the model uses these concepts and how they inform the model's prediction. Motivated by a desire to explain why a CNN makes the decisions it does in a human-interpretable manner, we propose a method that formulates a global interpretation of the semantic concepts the model is reasoning about (Ribeiro et al., 2016; Kim et al., 2017) using a shallow decision tree based on concept data extracted from activations at the hidden layer. This method is both efficient and portable. It does not require retraining of any existing model one wishes to test.

Because CNNs are largely black box systems, this global interpretation can be valuable as it grants a general understanding of how the model is behaving and provides a logical explanation for the decisions that the model makes. This kind of interpretation can increase confidence and trust in the model if it is found that it is making decisions that seem reasonable to humans.

[1]University of Southern California, Los Angeles, CA, USA. Correspondence to: Conner Chyung <cchyung@usc.edu>.

2019 ICML Workshop on Human in the Loop Learning (HILL 2019), Long Beach, USA.

Understanding how semantic concepts inform the decision the model is making can also be used to highlight potential unwanted bias learned by the model based on the most discriminating concepts learned by the decision tree.

Using a densely labeled image data set to probe the network, we show that for a classification problem with few classes, a shallow, interpretable decision tree can be learned that is nearly as accurate as the original model. We also demonstrate that the shallow decision tree learned performs comparably well to deeper, but less interpretable decision trees.

2. Related Works

Concepts: Much work has been done on the extraction of concepts learned in the CNN hidden layer. Fong & Vedaldi (2018) showed that combinations of filters are needed to encode a specific concept and showed how concept classifiers can be trained to recognize the presence of concepts in activations. Kim et al. (2017) presents a method which gives the ability to extract Concept Activation Vectors and test how sensitive a certain prediction is to a specific concept.

Decision Trees and Neural Networks: Balestriero (2017) presents a hybrid architecture of a decision tree and a neural network which is able to sometimes achieve an accuracy better than its neural network counterparts for specific problems. Frosst & Hinton (2017) shows how filter activations themselves can be used to train a decision tree, but the nodes of those trees don't necessarily communicate semantic meaning about what the model is deciding on. Zhang et al. (2018) is able to learn a decision tree based on semantic meaning. However their method requires a retraining of the entire network to get each filter to recognize a specific concept before being able to train a decision tree.

Our method is unique in that it provides a global and interpretable explanation of the CNN using a decision tree that shows how concepts interact without having to retrain the network being probed.

3. Methods

3.1. Probing the CNN to Train Concept Classifiers

Consider a densely labeled image dataset $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ with n data points labeled according to a set of concepts $\mathcal{C}$. $\mathbf{x} \in \mathbb{R}^d$ is an image with dimensionality d and $\mathbf{y} \in \{0,1\}^{|\mathcal{C}|}$ is a vector of binary variables indicating the presence of a concept in $\mathbf{x}$.

Given a pretrained image classification model m, for each image $\mathbf{x} \in \mathcal{D}$, the hidden layer activations $\mathbf{v} = m_l(\mathbf{x})$ at layer l are extracted and stored alongside the corresponding concept labels $\mathbf{y}$.

For each concept $c \in \mathcal{C}$, we train a binary linear classifier f_c on a dataset $\mathcal{G}_c$ which is based on dataset $\mathcal{D}$. We define $\mathcal{G}_c = \mathcal{G}_c^+ \cup \mathcal{G}_c^-$ where $\mathcal{G}_c^+ = \{(m_l(\mathbf{x}^{(1)}), y_c^{(1)}), \ldots, (m_l(\mathbf{x}^{(n)}), y_c^{(n)})|_{y_c=1}\}$ and $\mathcal{G}_c^- = \{(m_l(\mathbf{x}^{(1)}), y_c^{(1)}), \ldots, (m_l(\mathbf{x}^{(n)}), y_c^{(n)})|_{y_c=0}\}$.

In order to balance the data used to train f_c, $\mathcal{G}_c^-$ is taken as a randomly sampled set of negative examples such that $|\mathcal{G}_c^+| = |\mathcal{G}_c^-|$

Note: *As is common in the image classification domain, sometimes size of the hidden layer activation vectors is too large. To reduce the dimensionality of the concept classification problem, principle component analysis is applied to transform the activations to a reasonable width to train f. Additionally, spatial averaging is also applied if necessary.*

3.2. Extracting Concept Data

Consider an image classification problem with dataset $\mathcal{A} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(n')}, y^{(n')})\}$ with n' images and $y \in \{1, 2, \ldots, \gamma\}$, where γ is the number of classes. This time, $\mathbf{x} \in \mathbb{R}^d$ remains an image with dimensionality d, and now y is the class label for $\mathbf{x}$.

For each image, $\mathbf{x} \in \mathcal{A}$, hidden layer activations $\mathbf{v} = m_l(\mathbf{x})$ for the same layer l are extracted from the network. If PCA and/or spatial averaging was applied on the activations during the probing step, the same transformations are applied to $\mathbf{v}$ to achieve the same input dimensionality for f_c.

We use the concept classifiers to make a binary prediction for each $r_c = f_c(\mathbf{v}), r_c \in \{0,1\}$, to create a binary vector $\mathbf{v}' = (r_1, r_2, \ldots, r_{|\mathcal{C}|})$, representing whether or not each concept was present in $\mathbf{x}$.

The class prediction $\hat{y} = m(\mathbf{x})$ is also recorded to be used as the target output for training the decision tree.

3.3. Training the Concept Decision Tree

The concept vector $\mathbf{v}'$ predicted for each image from the classification problem, $\mathbf{x} \in \mathcal{A}$, as well as the corresponding prediction of $m(\mathbf{x})$ is used to train a decision tree. We use the default decision tree algorithm from *scikit-learn* (Pedregosa et al., 2011). The accuracy of the tree is calculated based on the prediction of m instead of the ground truth label to get the validation accuracy of the decision tree with respect to the representation learned by m.

Note: *At the time of writing, the algorithm that scikit-learn used for decision tree training was an optimized version of CART (Classification and Regression Trees)*

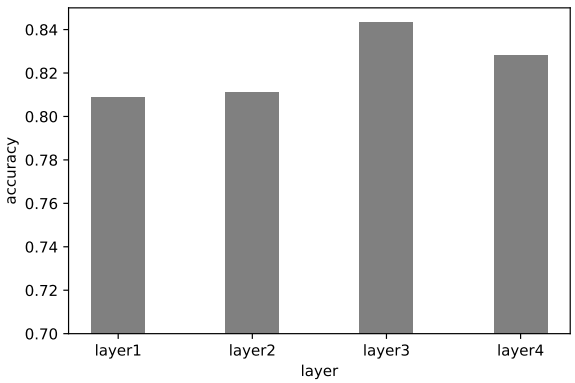


Figure 1: Average accuracy of all concept classifiers trained for each layer. Concept classifiers for layer 3 have a relatively higher accuracy.

4. Results

4.1. Data

For the densely labeled image set used to extract concepts learned by the network we use BRODEN from Bau et al. (2017). The BRODEN dataset is a collection of over 60,000 images with segmentations of concepts belonging to a number of abstract categories including materials, colors, and scenes.

BRODEN contains over a 1189 different concept labels belonging to different broader categories such as *material, scene* and *color*. However, some concepts in the dataset have much fewer labeled examples than the others. Concepts with less than 1000 examples were unused, leaving around 200 potential concept labels.

Because a majority of concepts were labeled at the pixel level, additional pre-processing was required to find every concept present in the image overall. Each image was iterated over and tagged for a specific concept if there were pixels in the image that were labeled with that same concept.

To extract concept decisions from a pre-trained model, the Natural Images dataset from Kaggle was used (Roy et al., 2018). The Natural Images dataset consists of 6899 images

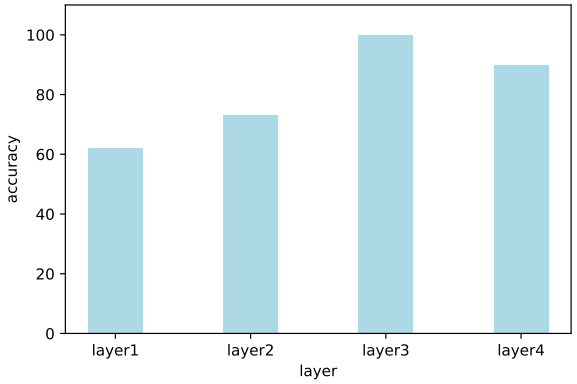


Figure 2: Number of Concept Classifiers whose accuracy was above 0.75. Once again, layer 3 outperforms the other layers.

with 8 distinct classes (airplane, car, cat, dog, flower, fruit, motorbike, person).

A subset of the Natural Images dataset, *Mini Natural Images*, with only 4 of the 8 classes is also used in a separate experiment (flower, dog, car, person). This version of the dataset consisted of 3499 labeled examples.

4.2. Experiment Setup

We probe Resnet50 pre-trained on the Imagenet dataset. (He et al., 2016).

The BRODEN dataset is used as the densely labeled image set to train concept classifiers for the network. Activations from all 4 *major layers* of Resnet50 are extracted. *Major layers* refer to the *conv2_x*, *conv3_x*, *conv4_x*, and *conv5_x* blocksections of sublayers of Resnet50. Spatial averaging and PCA are applied to lower the dimensionality of the activations. Additionally, in order to ensure the quality of the concept data produced, concept classifiers with validation accuracy scores below $\lambda = 0.75$ were discarded.

To create a toy image classification scenario, we retrain the classification layer Resnet50 on the Natural Images dataset. All layers before the classification layer are frozen to maintain the representation that was that was learned from ImageNet.

4.3. Concept Classifier Prediction Performance

Concept classifier accuracy varied across the different layers of Resnet. Figure 1 shows that the average classifier accuracy was the highest for the third layer (0.844). Unsurprisingly, as Figure 2 shows, it also produced the highest number of classifiers whose accuracy was above λ.

In general, concepts with more labeled examples from BRODEN achieved a higher accuracy. Regardless of which layer the activations were extracted from, concepts of the color

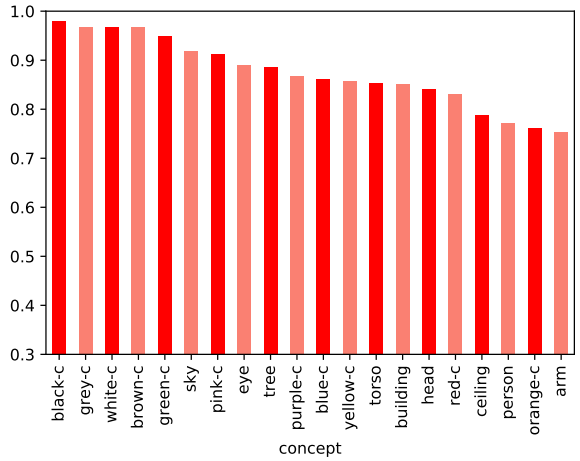


Figure 3: Concept classifier accuracy for the top 20 concepts trained using layer 3 of Resnet50. The top concepts are colors, i.e., black, grey, white, etc.

categories generally achieved the highest accuracy. Figure 3 shows the distribution of accuracy scores for the third layer of Resnet for the top 20 scoring concept classifiers. The top 5 are colors, but other general concepts such as *sky*, *building*, and *head* also scored well.

4.4. Decision Tree Performance

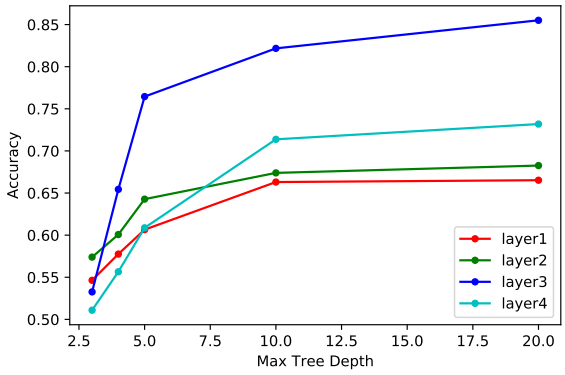
Figures 4a and 4b show how decision tree accuracy responds to changes in max. tree depth for each layer of Resnet50. Accuracy improves greatly at first as the max. tree depth is increased, but quickly begins to flatten out as depth increases beyond 10 levels.

Both plots demonstrate how layer 3 of Resnet50 has the best performance in terms of decision tree accuracy. This is especially evident in the decision tree accuracies for the Mini Natural Images dataset with the best decision trees reaching accuracy scores in the low 90s.

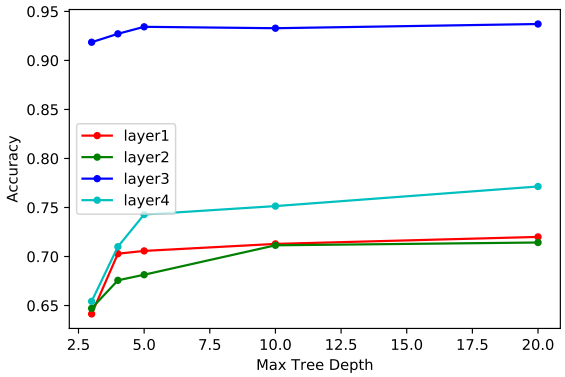
4.5. Interpretability

While training a decision tree with increased depth leads to higher prediction accuracy w.r.t the original model, it also leads to a less interpretable result as the number of nodes increases exponentially with depth. Thus a shallower decision tree with similar is preferred.

Figure 5 shows how a shallow decision tree can be trained to match the representation of Resnet50 trained on Mini Natural Images. The learned decision tree is able to provide reasonable explanations that apply to all input images for each class in Mini Natural Images. At the same time this specific tree was able achieve a relatively high accuracy (0.9134). Figure 4b shows that this tree also achieved an accuracy very similar to deeper trees trained on the same layer.



(a) Natural Images dataset.



(b) Mini Natural Images dataset. For this dataset, the accuracy maxes out quickly as max tree depth increases

Figure 4: Decision tree accuracy vs. max. tree depth for respective datasets

5. Conclusion and Discussion

A shallow decision tree with high accuracy w.r.t. the representation of the original model being probed gives insight into how the model might be reasoning about human-understandable concepts while making its prediction

Using this interpretation, one can infer which concepts are significant to the model based on which nodes are included in the decision tree.

It also provides an interpretable alternative to the CNN while still maintaining competitive performance.

Because this method is portable, it can be applied to any CNN and can be used to extract domain-specific for any given problem.

The decision tree can also be a useful tool for detecting bias learned by the CNN if it is found that a certain concept is a discriminating feature that should not necessarily be informing the overall decision.

Additionally, extracting concept predictions and training the decision tree is generally a fast process and is only hamstrung by the speed in which inferences can be run through the model. The time it takes to to get predictions from the concept classifiers and fit the decision tree consistently remains under a few seconds. This allows for efficient tuning of hyperparameters to create a tree that is optimal in terms of interpretability and accuracy.

Future work along the route of concept-based decision trees could include a method that can extract concept classifiers from each layer of the network together and analyzing which types of concepts are best classified at each layer. The highest performing concept classifiers from each each layer could also be combined together in an ensemble to provide a more holistic view of how concepts interact through the entire network.

Instead of using binary classifiers to detect concepts from hidden layer activations, regressors could be used to extract more fine-grained concept data. This could potentially lead to higher accuracy more interpretable results; even for larger image classification problems.

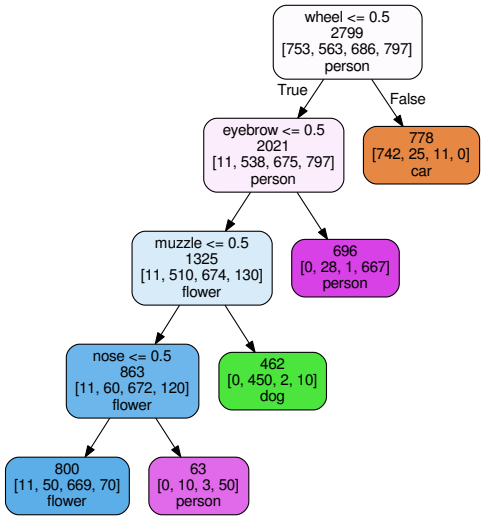


Figure 5: Shallow decision tree of max. depth 5 and min. sample size of 20 trained on layer 3 (conv4_x) of Resnet50. The left branch indicates that the concept is not present, while the right branch indicates that the concept is present. The tree gives insight into how the model might be making a prediction based on concepts for the Mini Natural Images dataset. The decision tree provides a natural and logical explanation for each path, i.e. the presence of 'wheel' indicates 'car', the presence of 'eyebrow' but not 'wheel' indicates a 'person', etc.

References

Balestriero, R. Neural decision trees. *arXiv preprint arXiv:1702.07360*, 2017.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. *CoRR*, abs/1704.05796, 2017. URL http://arxiv.org/abs/1704.05796.

Fong, R. and Vedaldi, A. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. *CoRR*, abs/1801.03454, 2018. URL http://arxiv.org/abs/1801.03454.

Frosst, N. and Hinton, G. E. Distilling a neural network into a soft decision tree. *CoRR*, abs/1711.09784, 2017. URL http://arxiv.org/abs/1711.09784.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ribeiro, M. T., Singh, S., and Guestrin, C. 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*, pp. 1135–1144. ACM, 2016.

Roy, P., Ghosh, S., Bhattacharya, S., and Pal, U. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018.

Zhang, Q., Yang, Y., Wu, Y. N., and Zhu, S.-C. Interpreting cnns via decision trees. *arXiv preprint arXiv:1802.00121*, 2018.