Knowledge Distillation for Interpretation of CNN using Decision Tree

Project Report to be submitted in Partial Fulfillment
of the Requirements for the Award of the Degree of

Bachelor of Technology
in
Manufacturing Science and Engineering

by

Samay Patel

Under the supervision of

Dr. Debdoot Sheet

Department of Mechanical Engineering
Indian Institute of Technology Kharagpur
December 2022

Abstract

THE interpretation of Deep Neural Networks (DNN) is still challenging due to the black-box nature of neural networks. One way of mitigating this problem is by using knowledge distillation. Knowledge distillation refers to the process of transferring the knowledge from a large complex model or set of models to a single smaller model. In this study we distill the knowledge of a convolutional neural network (CNN) which is hard to interpret into decision tree which is a widely used interpretable model. The decision tree with distilled knowledge of the CNN were obtained by training the tree with feature extracted from the CNN as the input. We extracted these features from all the intermediate layers of the CNN, by transforming the feature maps into the input feature vectors. These extracted features uniquely represent the filters in the intermediate layers of the CNN. Furthermore, we observe the performance of each layer by analyzing the features obtained from it by using a decision tree to observe the accuracy and investigate their discriminating capability of the filters to identify redundant filters. We utilized pathMNIST, octMNIST, and pneumoniaMNIST from the medical MNIST datasets for the demonstration.

Keywords: Convolutional Neural Network, Image Classification, Decision Tree, Knowledge Distillation, Feature Extraction, Feature Importance.

Contents

1 Introduction

For various practical machine learning problems, neural networks are well-known models. Different neural networks are used in many fields such as Recurrent neural networks (RNNs) are used to tackle issues involving time series data, whereas convolutional neural networks (CNNs) handle the majority of computer vision tasks. The form of the goal function is not always known before training, which is another benefit of utilizing neural networks. We can hand design the architecture, and hyper-parameters may also be used to manage the training. In general, deep learning has changed the face of image classification. Earlier methods include feature extraction and classification separately. Deep learning offers end-to-end architectures, thus no need for handcrafted features. Firstly, Alex Krizhevsky proposed AlexNet (Krizhevsky et al., 2012), which is indeed one of the breakthroughs in deep learning. Later, in 2014, Simonyan proposed VGGNet (Simonyan and Zisserman, 2015) for large-scale image recognition. Subsequently, GoogLeNet (Szegedy et al., 2015), proposed by Szegedy et al., has 22 layers. After that, He et al. proposed a 152-layer deep neural network called ResNet (He et al., 2016), which achieved state-of-the-art classification on the Imagenet dataset. Above all, UNet is a reputed architecture in deep learning for medical image analysis for biomedical image segmentation.

We can achieve higher and more accurate performance with neural networks. However, we ultimately fail to provide proper interpretations for model outcomes because the interpretation of neural networks is low (Gunning and Aha, 2019) due to its black-box nature. Apart from the accuracy, it is also essential to understand the reasoning process of the model, and meaningful decision-making is essential in many domains like medicine, defense, and law. In general, highly interpretable models are less accurate, whereas highly accurate models are less interpretable, as shown in figure 1. To accomplish the compromise between the two paradigms, we interpret highly accurate models using explainable models, and this can be achieved using knowledge distillation.

The goal of Knowledge Distillation is to use a powerful but computationally costly teacher model to build a lightweight machine-learning model called the student model. These student models' success is mainly attributable to the information they learn from overly parameterized teacher models. The information gathered from a teacher model is applied in order to actualize learning with a low-capacity student model.

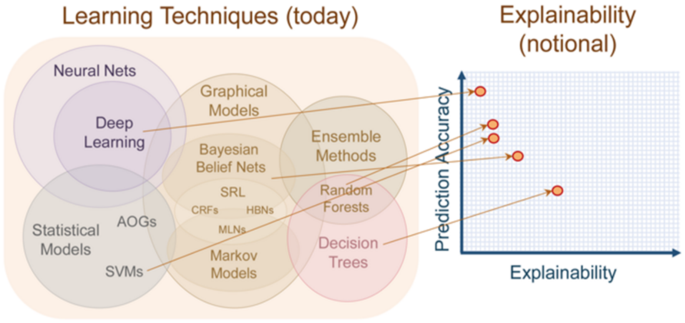


Figure 1: Explainability of different machine learning models (Gunning and Aha, 2019)

Decision tree is a widely used machine learning tool for classification and regression and has higher interpretability compared to other machine learning models. Decision trees being highly interpretable, figure 1, they are often used to understand neural networks among researchers. So, with the help of knowledge distillation, we train a more interpretable and less accurate a decision tree. That can be analyzed to understand the working of the CNN for image classification of medical images.

2 Prior Art

The use of knowledge distillation is one strategy for addressing interpretation issues(By using knowledge distillation to implement teacher-student Models). We can achieve accuracy of the shallow student model close to that of the deep teacher model. Deep feed-forward nets previously learned complex functions, but shallow feed-forward nets are now able to do the same and attain accuracy that was previously only possible with deep models. Additionally, the shallow nets may be able to learn these deep functions using the same amount of parameters as the original deep models in some circumstances (Ba and Caruana, 2014).

In machine learning, the problem is solved by designing a classifier using a set of features. Therefore, to get around this, CNN is utilized to create features automatically and then mix them with the classifier (Jogin et al., 2018). The list of layers in the CNN classifier that convert input volume to output volume is the simplest of all the classifiers, which is one of its advantages. There are only a handful of separate layers, and each layer uses a different function to transform the input into the output.

When we want to analyze any model it is not just the final layer of the model we need to focus on. We also need to observe the behavior of intermediate layers and analyze their outputs and performance. To analyze the CNN model we can try to extract the features from each layer and analyze them using decision trees.

3 Scope and Objectives

The *aim* of this project is to investigate the contribution of the filters in intermediate layers of convolutional neural networks in the process of classification of medical images to uniquely identify the essential and redundant filters.

The objectives of this project are summarized below :

1. Train the decision trees (student model) with the help of trained CNN (teacher model).
2. Investigate these decision trees to identify the important and the redundant filters.

4 Work Progress and Achievements

4.1 Methodology

In this study, we start with training a five-layer CNN for a classification task, on a dataset from the collection of medical MNIST datasets. Once the training is completed, features are extracted from the intermediate layers of the CNN, and these extracted features are then used as input to train five decision trees for five layers of the CNN. As shown in figure 2, input features are computed by extracting the feature map from intermediate layers of the CNN, and an average of each channel of these feature maps is computed for all samples in the dataset.

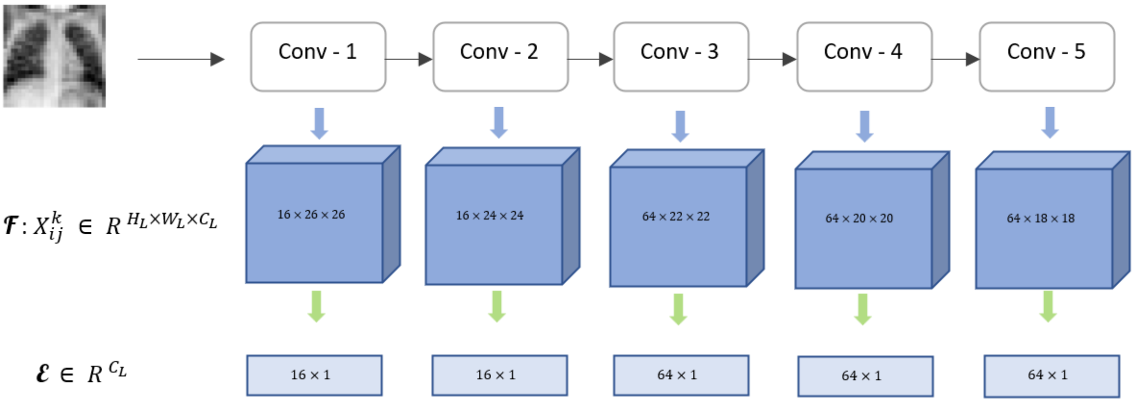


Figure 2: In the figure, H_L, W_L and C_L represents the height, width and channels of the feature map from L^{th} layer and green arrow represents the mapping m to extract features E from the feature map F.

These extracted features uniquely represent each filter in the intermediate layer of the CNN. Further, these trained decision trees are analyzed to see how each feature contributes to the classification of an image. The lesser contribution of a feature in the decision-making process is evidence of the redundancy of the filter. For analyzing the feature importance, we prepared datasets for each layer of CNN from extracted features for every image in the dataset. As shown in figure 3, for each layer of CNN, noise is added to one of the features by shuffling a column in the dataset of a layer, and the accuracy of the decision trees with this input was noted.

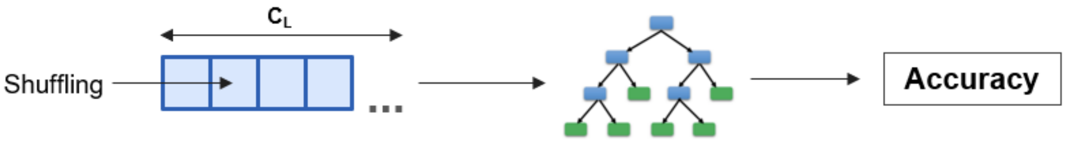


Figure 3: Shuffled one of the features and accuracy is noted for this input.

Accuracies obtained were then plotted corresponding to the feature shuffled for each layer of the CNN. A drop in the accuracy of the decision tree for a shuffled input represents the importance of the feature and the corresponding filter in the classification of the image by CNN.

4.2 CNN Traning and Feature Extraction

The convolutional neural network used in this study, shown in figure 4, is a five-layer network with a fully connected layer at the end of the network with softmax activation function, and all intermediate layers of the network consist of a convolutional layer with filter size 3x3, along with batch normalization and relu activation function.

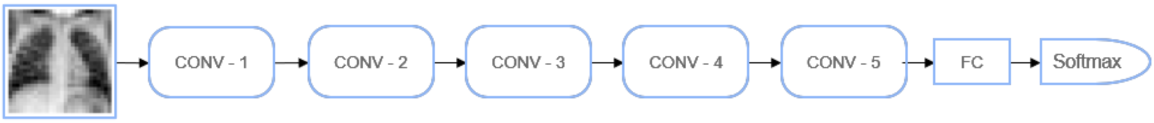


Figure 4: CNN Architecture

Filter analysis was performed on the CNN, trained on three datasets from the collection of medical MNIST datasets, shown in table 1.

Medical MNIST	**Samples**	**Classes**	**Modality**
pneumonia	5,856	2	Chest X-Ray
oct	109,309	7	Retina OCT
path	107,180	9	Colon Pathology

Table 1: Medical MNIST datasets used.

Hyperparameters used for the training of CNN are shown in table 2.

Hyperparameter	**Value**
learning rate	0.001
batch size	128
epochs	20
optimizer/momentum	SGD/0.9
loss	Cross entropy

Table 2: Hyperparameters used for training of CNN.

After traning the CNN feature maps for each samples are transformed and stored for training the decision trees by performing the average operation on the feature maps.

$$e_a = \frac{\sum_i^{W_L} \sum_j^{H_L} f_{ij}^{k=a}}{H_L \times W_L}$$

where feature map extracted from layer L : F_L with f_{ij}^k, (i^{th}, j^{th}) value in k^{th} channel of the feature map and extracted features : E_L with e_a, a^{th} value of the feature vector.
Here are the extracted feature maps and transformed feature vector for each layer, for input to the CNN, medical image : I $\in \mathbb{R}^{28\times28}$

$$F_1 \in \mathbb{R}^{16\times26\times26} \rightarrow E_1 \in \mathbb{R}^{16}$$
$$F_2 \in \mathbb{R}^{16\times24\times24} \rightarrow E_2 \in \mathbb{R}^{16}$$
$$F_3 \in \mathbb{R}^{64\times22\times22} \rightarrow E_3 \in \mathbb{R}^{64}$$
$$F_4 \in \mathbb{R}^{64\times20\times20} \rightarrow E_4 \in \mathbb{R}^{64}$$
$$F_5 \in \mathbb{R}^{64\times18\times18} \rightarrow E_5 \in \mathbb{R}^{64}$$

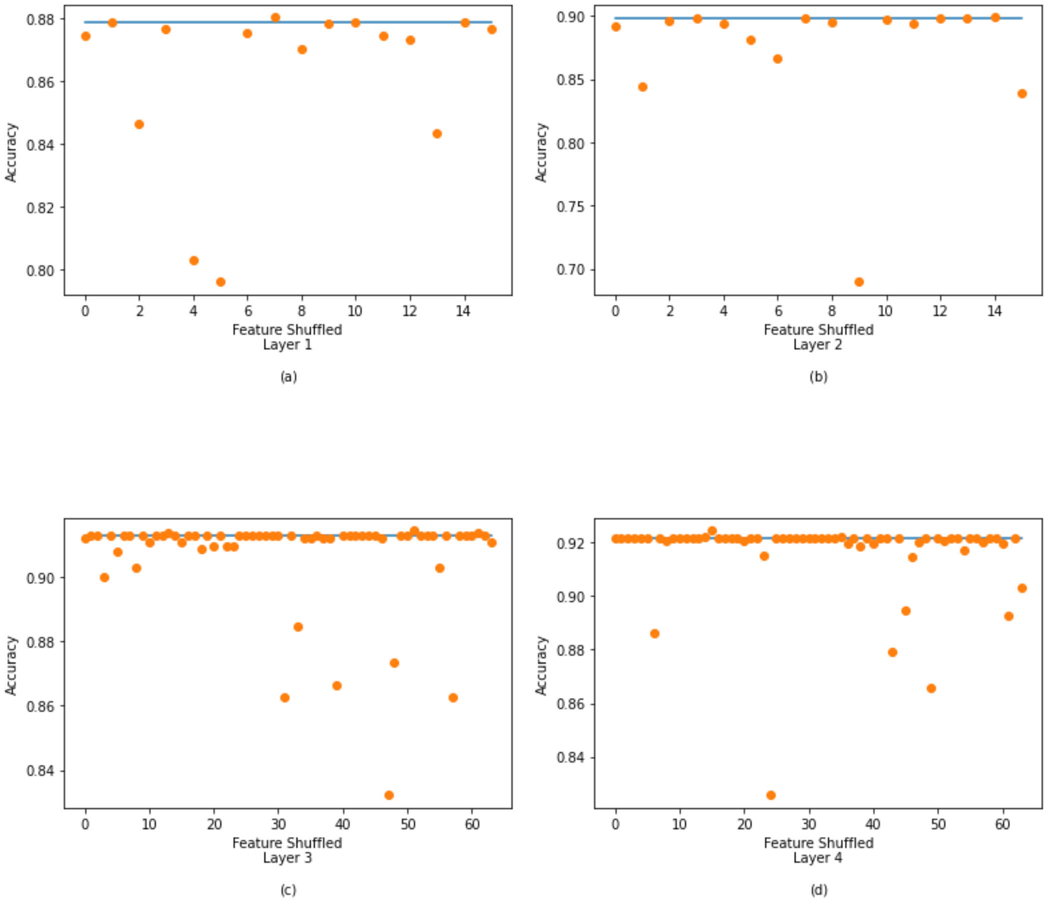
4.3 Results

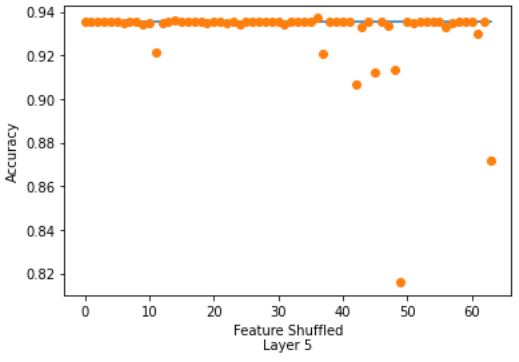
The table below shows the accuracy of the decision tree increases from layer 1 to layer 5, for all the datasets. This is due to, in the initial layers of the CNN, the data is projected into the feature space where the variance is low. Therefore, it is difficult to obtain features with high discriminating capability. The final layer projects the data into a feature space, where the feature obtains high classification ability.

Dataset	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**	**CNN**
pneumoniaMNIST	87.54	89.80	91.15	92.07	93.77	95.8
octMNIST	65.67	70.03	72.33	75.93	81.31	81.60
pathMNIST	66.49	68.27	64.58	74.17	80.45	83.6

Table 3: Accuracy(%) of CNN and decision tree on Medical MNIST datasets, with features extracted from layer 1, layer 2, layer 3, layer 4, and layer 5 of the CNN.

Further, the feature importance for the decision trees was analyzed to identify the redundant filters in the intermediate layer of the CNN.





(e)

Figure 5: In the above figures, orange points represents the accuracy of the decision tree after shuffling one feature for all the layers of the CNN and the horizontal blue line represents accuracy of the CNN

In figure 5, the drop in the accuracy of the decision tree for a shuffled input represents the importance of the feature and the corresponding filter in the classification of the image by the CNN.

5 Summary and Future Work

In this study, we investigate the filters in the intermediate of the convolutional neural network using decision trees. We extract features from each layer of the CNN by transforming the feature maps from the intermediate layers of the CNN for each sample. These extracted features uniquely represent the filters of the CNN. We transform each image into a feature vector, then this acquired data is used as input data to decision trees, for training. Then we investigate the performance difference of these trained decision trees's between local features at the initial layers and global features at the final layers of the CNN. Also, we analyzed the feature importance of these extracted for the decision trees to determine the importance of the corresponding filter in the classification of the image for the CNN.

References

Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep?, *in* Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Weinberger (eds), *Advances in Neural Information Processing Systems*, Vol. 27, Curran Associates, Inc.

Gunning, D. and Aha, D. (2019). Darpa's explainable artificial intelligence (xai) program, *AI Magazine* **40**(2): 44–58.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 770–778.

Jogin, M., Mohana, Madhulika, M. S., Divya, G., Meghana, R. and Apoorva, S. (2018). Feature extraction using convolution neural networks (cnn) and deep learning, *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* pp. 2319–2323.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Communications of the ACM* **60**: 84 – 90.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, *CoRR* **abs/1409.1556**.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 1–9.