Interpreting and Explaining Covid-19 Chest X-ray Image Classifer using Tensorview and Paraview

Ashik Barua
University of South Florida

Parush Gera
University of South Florida

Fariha Moomtaheen
University of South Florida

Abstract

It is known that the deep learning models have a black box nature. They are better in predicting than traditional machine learning models, but what happens inside is very complex to understand. In sensitive fields like medical, just blindly relying on the accuracy of the model to predict the disease from x-ray/CT images can be dangerous. In this paper we put efforts to visualize the prediction process of a deep learning model which can aid in decision making for professionals in medical field.

Index Terms: Visualizing Model—Visualization—Visualization techniques——Deep Learning —Visualization—interpretability and explainability

1 Introduction

Artificial intelligence has come out to be a revolutionizing aid in solving many complex problems in different domains including medical treatments. Numerous efforts has been made till now in improving the prediction of Machine Learning (ML) and Deep Learning (DL) algorithms being used in predicting the severity, or early prediction of diseases using scanned images of patients. Since, it is adapted widely by the world, the sensitive nature of the area of medical treatment has evoked researchers to interpret the underlying ML/DL model to better understand the classifying mechanism. Taking clinical decisions just on the basis of high accuracy is not enough for this domain and thus there is a trade-off between the prediction accuracy and interpretability of the models. Interpreting these methods by using different visualization methods can give a better understanding of how the model is working, thus aiding them in taking clinical decisions.

Visualization techniques has aided researches in the field of simulation and accelerated-data driven researches. There has been work done in past which uses tools like Paraview and Tensorview to visualize computational models to interpret models' learning process. If we can visualize the learning process, it might answer : Is it possible to find gradient vanishing/explosion during learning process¿, how are the weights being updated?, can we know something about model's learning process by visualizing loss and accuracy at each iteration.

With the recent outbreak of COVID-19 pandemic and the evolving nature of the virus has brought up an urgent need of understanding the features or data points used by prediction mechanism predicting the severity of the disease. Our objective of their paper is to apply state-of-art deep learning and machine learning mechanism along with previously used mechanism of interpretable methods to understand the classification mechanism on x-ray images of Covid-19 positive patients.

The next section discuss some previously done work on interpreting machine learning algorithms.

2 Related Works

There have been multiple efforts till now which has explored interpretability and explainability of ML and DL algorithms used for prediction in medicine including radiation oncology. There is a trade-off between the prediction accuracy of the algorithm and its interpretability. In [12] , the author has reviewed application of many machine learning algorithims like Logistic Regeression, Decision Tree, Random Forest, Support Vector Machines, Bayesian network and Deep Learning and their interpretability, further classifying them into IP(interpretable) and NIP(non-interpretable). There exists a trade-off between accuracy and interpretability of algorithms because taking clinical decisions for patients on the basis of predictions by these algorithms requires a level of trust, which is gained by understanding in underlying mechanism and how it is predicting the results thoroughly. Some NIP ML models can be interpreted by integrating them with IP models using hand engineered features, but this mechanism of integration does not work well on DL algorithms because they extract features automatically and also in large number compared to hand engineered features used by traditional ML models.

In [3], many integration methods has been discussed which can be used to understand the black box nature of the complex machine learning models such as SVMs using rbf or polynomial kernels, deep learning models. For example, using a significantly more interpretable models such as decision trees, K-NN to explain uninterpretable deep neural networks.

Topological data analysis is a collection of data exploration and analysis methods,studies with structures in complex, high dimensional data-set using techniques from topology. One of the popular topological data analysis method is persistent homology, used in [9] to detect and quantify structural changes in time-varying graph data. In addition, they [9] deployed an intuitive visual interface to investigate and represent structural changes in graph using persistence based similarity measures. This paper [14] describes an approach to reduce complexity of data flow based on the notion of a topological skeleton, known as vector field simplification. It reduces the complexity of flow by removing features in order of their relevance and importance to reveal prominent behavior and thus obtains a compact, consistent and multi-scale representation of the flow dynamics. Also, with the growth of data, most of the data exploration and analysis techniques have turned into data-driven models to handle huge volume and complexity. In [11], they provided a scalable technique to analyze and visualize data-driven models for topological data. The main idea here is to interpret and get a better understanding of what's going inside these complex data-driven models while they're training and evaluating data.

Wenli Cai et al in [4] have shown a study on a set of Computed Tomography (CT) images to find out notions, symptoms and severity of SARS COVID-2 virus in order to predict clinical outcomes of covid-19 patients. Here, they used a deep neural network on the chest CT images to segment lung and lesions. [13] also shows lung CT image analysis to detect cancer disease severity using a U-Net

architecture model.

P hall in [10], explored some potential explanatory techniques which can be used to explain the machine learning models. Many methods like, decision tree surrogate models, local interpretable model-agnostic explanations (LIME), plots, partial dependence plots, Shapley explanations, and individual conditional expectation (ICE) can be used to explain ML models. The application of these methods is explored in different domains for example credit card dataset, to understand how they interpret corresponding to a particular domain and thus help is choosing the best explanatory model for the model.

In an assessment by M Feng. et. al [8], machine learning models have facilitated the clinical treatments in the area of radiation oncology, but there is still lot to be done. For instance, following up with treated patients to monitor their health by checking any indication of tumor being generated again is dependent on clinical experience and qualitative assessments [7]. There exits a need of reliable ML models which can revolutionize this process of follow up treatments by predicting these tumor generators early. For this, the prediction is dependent on the available training data, which the classifier can fail to learn some features if not present in the training data. For this the complex models like deep learning needs thousands of diverse data points which can be collected across different institutions rather then from a single source. At the same time as stated by [12], there is a trade off between the accuracy of prediction and the interpretability of the ML model.

Lastly, our main objective for this project is to visualize the dynamics of any deep neural network to help people to understand how these artificial networks learn from data and make predictions. In [5] and [6], Chen et.al gave good explanation for visualizing the run-time changes of the model using an open-source tool called Paraview.

3 METHODS

Our main focus in this project is to interpret and understand the workflow and data-flow in a neural network during training. A neural network has to be fed with lots of training data, which updates the weight of each neuron with each epoch. Also, by changing different hyper-parameter (such as learning rate, momentum, number of epochs, batch size, etc) values, we can significantly improve the model performance. While training a network, weights of any neuron, activation functions, losses and back-propagated gradient values carry the information of training status [5]. Each layer passes value to the next layer by running the weight through activation function, which will eventually be responsible for updating the loss function value. Currently, our goal is to visualize how loss and accuracy of the model changes for varying number of epochs. In this paper we will be visualizing following key aspects: []

- Visualizing Loss and accuracy while training.
- Visualizing loss accuracy over each epoch.
- Visualizing kernel of each Neuron at a layer.
- Visualizing Bias at each filter.

3.1 Covid vs. Non-covid image classifier model

We built a Convolutional neural network which performs classification on Covid and Non-Covid chest X-ray images. We built this model in python using Tensorflow and Keras. The goal of this project is to interpret and explain this model visually using Tensorview and Paraview.

3.2 Dataset

Since the focus of this project is not making a highly accurate prediction model but to visualize the learing and predicting process, we have kept our data set balanced for simplicity.We have total of 56 images. There are 21 Covid-19 [1] positive images of lung CT and there are 35 non-covid [2] images.

3.3 Tools

We used mainly 3 tools to make the visualizations- *Tensorview*, *Paraview* and *Tableau*.

3.3.1 Tensorview

We are using Tensorview API for visualizing the training and predicting process of our model. Specifically we will be visualizing the weight and activation functions. Tensorview is an open framework which is written in python and can be easily integrated with Tensorflow. The output of Tensorview can be visualized with matplotlib, or the outputs can be exported in .csv or .vfp files. This output can be used as input in Paraview for further visualization analysis.
Tensorview API is fairly new and is still in it's developing stage. As a result, the documentation for this is scarce. The only existing documentation is the following github repo-
`https://github.com/Hourout/tensorview`

3.3.2 Paraview

Paraview is an open source platform which is used to visualize large data sets. Paraview can be used to visualize computational models. It is a complex tool with advanced facilities. We fed the loss and accuracy data into Paraview and visualized the progression in 2D and 3D perspectives.

3.3.3 Tableau

Tableau is an interesting and powerful tool for researchers and business communities to make interactive visualizations. We used tableau to visualize the propagation of weight values inside the deep learning model. By help of tableau we've been able to see and colorize parts of an otherwise so called black box model.

4 RESULTS

We used data extracted from our model in Paraview to visualize it. Figure 1 shows the extracted data and Figure 2 shows a 3D visualization created by preview. From Figure 3, which is a 2D representation of Figure 2, we can infer that after 50 epochs there is no need to train it further. Figure 3 shows a 2D representation of loss and accuracy(right to left) which can be used to infer the changes in the loss and accuracy while the model learns, where as figure 2 shows the same data in a 3D representation but along with epoch information. Here we can see that initially the model is struggling to learn as the loss is high and accuracy is less. Further when the images are fed into the model in batches randomly with every epoch, the model gets to learn more and hence the accuracy increases and loss decreases along with epochs. This can help in defining for how long a model needs to be trained, or if the dataset has sufficient diverse features which enables the model to learn more.

In figure 5, we've shown how the kernel values in a single layer can be distributed over a 2-dimensional data space. We have used kernel of size 3x3 throughout the model, and captured and visualized the changes in kernel weights for each filter. This figure shows the kernel values of a hidden Conv2D layer, which has around 38000 parameters. We can compare kernel value graph for any layer after a certain number of epochs (say 10 or 20) which may show the change in kernel value which the model is training. Distribution of kernel values might lead us to a better understanding of the trend how it is changing. In other words, we can compute the distribution of the kernel values for each layer after certain epochs, and this may lead us to some meaningful insight with the kernel value change.

loss	acc	epoch
0.715684	0.5	1
2.217418	0.75	2
15.51354	0.25	3
3.9342	0.25	4
13.38724	0.5	5
7.008899	0.5	6
2.77293	0.25	7
1.1348	0.75	8
8.310104	0.25	9

Figure 1: Loss and accuracy values for First 10 Epochs

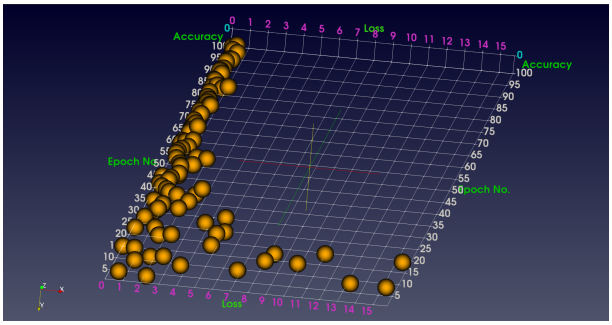

Figure 2: 3D-Paraview plot of accuracy and loss at each epoch.

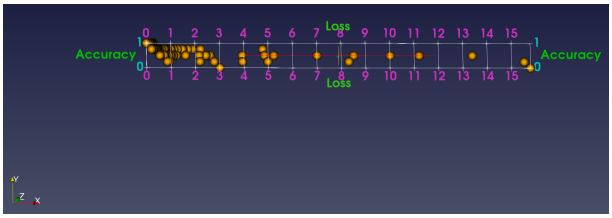

Figure 3: 2D-Paraview plot of accuracy and loss at each epoch.

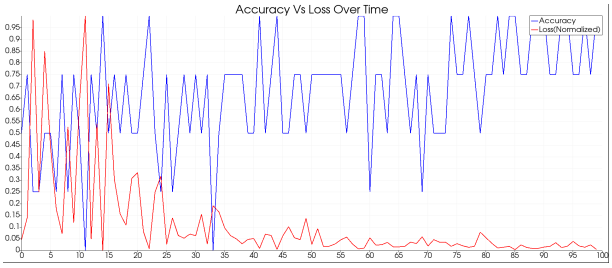

Figure 4: 2D-Paraview plot of accuracy and loss at with normalised values

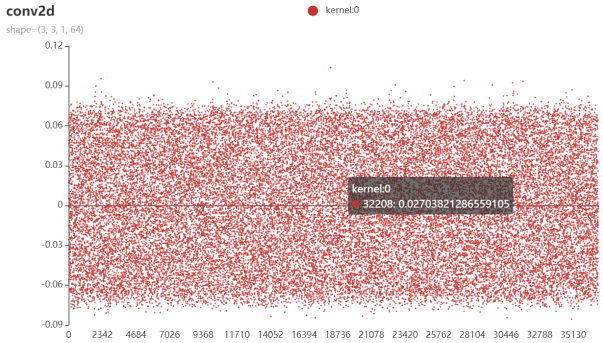

Figure 5: 2D-Tensorview plot of kernal values of neurons at layer 5.

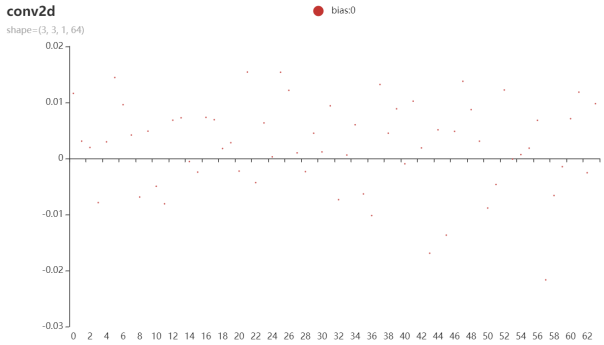

Figure 6: 2D-Tensorview plot of bias values at a particular epoch.

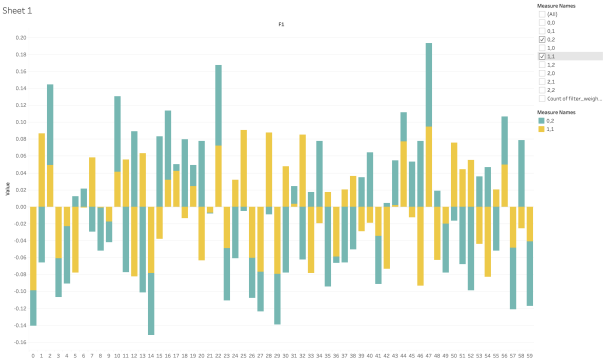

Figure 7: Propagation of weight values for 64 filters of the grids (0,2) and (1,1).

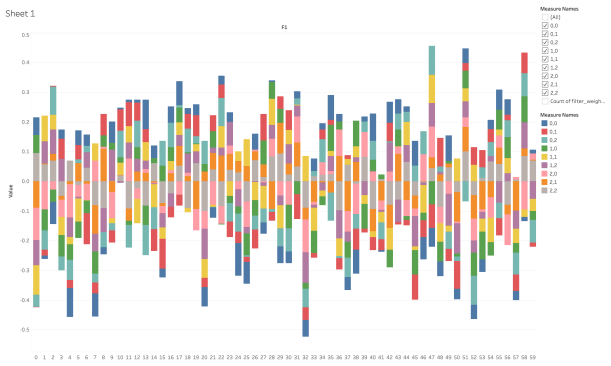

Figure 8: Propagation of weight values for 64 filters of all the grids of each 3x3 filter.

In figure 6, we can see that there is a bias value associated with each filter. The value changes with each epoch when the model learns over the number of specified epochs. The figure only shows the bias values a one epoch. But this information can be extracted for all epochs and the plots can be compared to understand the learning process. If the bias value is being updated, that means either the the model is learning or it is losing the information. But comparing the bias value at an epoch with the loss and accuracy at an epoch, like in figure 2 can lead us to infer that if the model is learning or loosing information.

Figures 7 and 8 shows the tableau visualizations. Our model uses (3x3) filters throughout the network. In every layer, there are different numbers of filters, like 16, 64, or 512. If there are 64 filters, it means a pixel or grid in our image will go through 64 updates of weight values. As each filter is of size 3x3 we have 9 grid values. We've been able to store the weight values of every epoch in each layer. We wanted to see how the weight values are changing through each filter. Figure 7 shows the weight values of two filter grids-(0,2) and (1,1), and in filter 8 we can see the alterations of weight values in a stacked manner of all the 9 filter grids. Each grid has been depicted by a different color for ease of understanding, and by choosing any number of grids we can compare and analyze the changes in weights and make decisions about the model.

5 Challenges

- Tensorview is a very recent library which has no detailed documentation available yet. But we have to use it because it is compatible with python and the functions in tensorview can give us output of important in-situ features to visualize. For example, neurons, loss rate and accuracy in each iteration. There is only one Github repository for Tensorview which is still in its developing stage, and there are very few functions available with no specific explanations.

- Learning and knowing how to properly use Paraview visualization tool is a long time process. It is very rich in functionality which has proportionately increased the difficulty of learning it.

- Using Paraview, or any other tools to visualize deep learning model is a fairly untouched area. As a result, there are not much precedences for what we are doing and lack of experience and resources has become a major challenge to overcome in achieving our goal.

- Some of the related papers which talked about interpratibility and explainibility visualized the filter and other key terms using the paraview, but they do not suggest how they were able to extract these values before feeding into the paraview software.

6 Conclusion

In this paper, we put efforts to explore the possibilities of visualizing a deep neural network to better understand how the model is predicting. Eventually, this can help professionals in medical field in decision making for patients. We used Tensorflow and Tensorview API to make the model and visualize it. Few of the visualizations were made using the paraview software which can help in visualizing the models predicting process in 3D. Following [5] and [6], we used the tools. We tried to replicate their experiments but due to lack of some informations about how they got the output to fed in the paraview, we tried to focus on accuracy and loss. Since these were the two parameter which we could take out from the model in order to understand it performance and learning process. We were successfully able to visualize the above extracted parameters. Our results suggests that the loss and accuracy can be visualized over epochs which can give an insight about what is happening at different stages of learning. We also have visualized the bias and kernel values at each filter in every epoch. Comparing the visualizations of bias values between different epoch counts, it can lead to us to infer that there is a change in the learning. It can be positive or negative. But if we compare bias values of one epoch to the loss and accuracy of same epoch, we can take down the bias values which corresponds to a positive or negative change.

7 Future Work

We plan to extract the above visualized data for the batches of data which is fed into the model. It will be interesting to visualize, that how does the model behave when the batch size is changed. Also, we are working on to better understand the Tensorview and Paraview, which we will use to visualize the learning rate with each epoch. It will be interesting to visualize the learning rate with each epoch, which will give us information about gradient vanishing or exploding problems.

8 Supporting Material

- Github Link: `git@github.com:ashikbarua/Data_Visualization_final_project.git`

- YouTube Link: `https://youtu.be/wpUhd6sj8Cw`

- Tableau Visualization: `https://public.tableau.com/profile/ashik.barua#!/vizhome/Book2_16193976245090/Sheet1`

Acknowledgments

The authors wish to thank Dr. Paul Rosen, Dr. Issam El Naqa and Teaching Assistant Ghulam Jilani Quadri for their direction, guidance and collaboration.

References

[1] https://github.com/ieee8023/covid-chestxray-dataset.

[2] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. doi: 10.1016/j.inffus.2019.12.012

[4] W. Cai, T. Liu, X. Xue, G. Luo, X. Wang, Y. Shen, Q. Fang, J. Sheng, F. Chen, and T. Liang. Ct quantification and machine-learning models for assessment of disease severity and prognosis of covid-19 patients. *Academic radiology*, 27(12):1665–1678, 2020.

[5] X. Chen, Q. Guan, X. Liang, L.-T. Lo, S. Su, T. Estrada, and J. Ahrens. Tensorview: visualizing the training of convolutional neural network using paraview. In *Proceedings of the 1st Workshop on Distributed Infrastructures for Deep Learning*, pp. 11–16, 2017.

[6] X. Chen, Q. Guan, L.-T. Lo, S. Su, Z. Ren, J. P. Ahrens, and T. Estrada. In situ tensorview: in situ visualization of convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1899–1904. IEEE, 2018.

[7] I. El Naqa, G. Pandey, H. Aerts, J.-T. Chien, C. N. Andreassen, A. Niemierko, and R. K. Ten Haken. Radiation therapy outcomes models in the era of radiomics and radiogenomics: uncertainties and validation. *International Journal of Radiation Oncology• Biology• Physics*, 102(4):1070–1073, 2018.

[8] M. Feng, G. Valdes, N. Dixit, and T. D. Solberg. Machine learning in radiation oncology: Opportunities, requirements, and needs. *Frontiers in Oncology*, 8:110, 2018. doi: 10.3389/fonc.2018.00110

[9] M. Hajij, B. Wang, C. Scheidegger, and P. Rosen. Visual detection of structural changes in time-varying graphs using persistent homology. In *2018 ieee pacific visualization symposium (pacificvis)*, pp. 125–134. IEEE, 2018.

[10] P. Hall. On the art and science of machine learning explanations, 2020.

[11] S. Liu, D. Wang, D. Maljovec, R. Anirudh, J. J. Thiagarajan, S. A. Jacobs, B. C. Van Essen, D. Hysom, J.-S. Yeom, J. Gaffney, et al. Scalable topological data analysis and visualization for evaluating data-driven models in scientific applications. *IEEE transactions on visualization and computer graphics*, 26(1):291–300, 2019.

[12] Y. Luo, H.-H. Tseng, S. Cui, L. Wei, R. K. Ten Haken, and I. El Naqa. Balancing accuracy and interpretability of machine learning approaches for radiation treatment outcomes modeling. *BJR—Open*, 1(1):20190021, 2019. doi: 10.1259/bjro.20190021

[13] B. A. Skourt, A. El Hassani, and A. Majda. Lung ct image segmentation using deep neural networks. *Procedia Computer Science*, 127:109–113, 2018.

[14] P. Skraba, B. Wang, G. Chen, and P. Rosen. 2d vector field simplification based on robustness. In *2014 IEEE Pacific Visualization Symposium*, pp. 49–56. IEEE, 2014.