

IN-BROWSER VISUALIZER FOR NEURAL NETWORK TRAINING

An Undergraduate Thesis By

Megan Dass

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in the
College of Computing

Georgia Institute of Technology

May 2023

© Megan Dass 2023

CHAPTER 1
ACKNOWLEDGEMENTS

I'd like to thank my research mentor, Professor Duen Horng (Polo) Chau, for his constant support on not only this project but since I joined Georgia Tech. He's been an incredible mentor guiding me through the research process, as well as introducing me to the industry. I'd also like to thank the other group members, Zhiyan Zhou, Kevin Li, Haekyu Park, Austin Wright, and Nilaksh Das, for their contributions towards this research project.

CHAPTER 2

ABSTRACT

With the increased use of artificial intelligence (AI) in our everyday lives, there is also a growing interest within the field to truly understand how neural networks come to decisions. Within the computer vision (CV) field specifically, with applications such as facial recognition and object recognition, deep neural networks (DNNs) are commonly used to carry out a variety of CV tasks. However, there remains a need to uncover the blackbox nature of DNNs to improve security, increase public trust in object recognition models, and be able to build more advanced models. We present an open-sourced and in-browser tool that allows users to visualize the inputs of the DNN at various layers and epochs using a dimensionality reduction technique called AlignedUMAP. The 2D dimensionality reduction graph shows users how an input might be classified at various stages in the models training process, as well as allows them to compare different inputs using a spatial visualization to understand how class labels may be closely related.

CHAPTER 3

INTRODUCTION

Artificial Intelligence (AI), specifically computer vision (CV), has been becoming more seamlessly integrated into our daily lives, from recognizing faces in camera rolls to helping healthcare professionals analyze X-rays and MRIS [1] [2]. Because of recent developments that have led to higher-quality performance by AI models, we have seen an surge in the use of AI in various industries in the past few years. One specific technological advance, was deep learning and deep neural networks (DNNs) [3]. Unlike many other neural networks, DNNs are able to train on a very large dataset with high performance due to the complexity of DNNs, with many layers and hyperparameters [4].

While the performance of DNNs on CV tasks is remarkable, they remain prone and susceptible to adversarial attacks [5]. Because DNNs are a blackbox by nature, developers and researchers are unable to create defenses for these attacks because they do not know how the DNN perceives the input and changes the output in response to the attack. For high impact industries like healthcare and defense, especially, the potential for attacks can be very detrimental [6]. Because of the vulnerability of DNNs, this has also led to a public distrust in the outputs of neural network, and in turn, the broader AI and deep learning fields. A solution to the vulnerability of DNNs would be for researchers to be able to better understand how DNNs make decisions to better disseminate knowledge about them, as well as create proper defenses to potential adversarial attacks [7].

A viable method to better understand how DNNs make decisions is to visualize the inputs and how they are perceived at various points during the training process of DNNs. Dimensionality reduction is a common method to visualize high-dimension data within

neural networks [8]. Principal Component Analysis (PCA) and Factor Analysis are two very common linear dimension reduction methods [9], but there has also been a rise in non-linear dimension reduction methods. For example, Uniform Manifold Approximation and Projection (UMAP) is a more effective manner to represent high-dimension data in a lower dimension compared to common linear dimension reduction techniques [10]. Further, AlignedUMAP is an extension of UMAP that also focuses on encoding features in low dimensions and aligns input embeddings across training epochs to ensure that the resulting visualizations properly show the evolution of the inputs across epochs [11].

In this project, we have worked on the following contributions:

- **An open-sourced and in-browser visualization tool.** Using modern technologies such as React.js and ScatterGL, we are able to create a cross-platform web-based application where users can view the UMAP visualizations of a trained ResNet-50 model, while also being able to customize various hyperparameters such as n-neighbors and minimum distance.
- **AlignedUMAP Visualizations.** Our proposed system uses a modified approach called AlignedUMAP to resolve the problem faced with normal UMAP so that the UMAP visualizations across adjacent epochs are aligned and coherent.

CHAPTER 4

LITERATURE REVIEW

Computer vision is a growing subfield of artificial intelligence that aims to make a machine or computer visualize and interpret graphical inputs as a human does with their eyes. This is a growing subfield because of its vast application areas such as object detection and object classification [12]. In order for a machine to be able to properly identify and classify an object, the training process is quite intense. Not only does it require a large and robust dataset, but the model must also be able to handle the intensity of the training. For this reason, DNN's have been shown to perform better at many computer vision tasks compared to other neural network types [13]. Because of the complexity of DNN's, the internal operations of DNN's currently remains a blackbox. While the structure of the neural network may be established, how the input is passed between layers and modified to generate a classification output is currently unknown [5]. This raises a challenge when trying to debug or improve the performance of a DNN [14]. Further, this opens the door for adversarial attacks as users are unable to understand how a neural network processes a benign versus attacked input, so they are unable to create defenses against these adversarial attacks [15].

There has been a significant amount of research in the field that delves into understanding the decision-making process of DNN's. More generally, Chung, Kraska, et al. aim to allow users to track how a model makes a decision in order to help debug where the model may or may not have gone wrong, overall trying to eliminate the blackbox nature of many neural networks. They create slices in the network and evaluate the data at that slice to try and validate the model [16]. Beyond decision making of one model, Zhang, Wang, et al. created an interactive scatter-plot visualization system that compares the output of two different neural network models to help researchers debug the model's outcome [17].

Novel visualization systems have been a widely used and proven technique to understand common mechanisms of DNNs, which can be further extended to understand how models interpret and classify inputs.

Within the visualization community, creating visualization systems to understand how data passes through the neural network has been vital. Kornblith, Norouzi, et al. aim to explore visualization techniques that will give the machine learning community insight into how convolutional networks work. They do this primarily by comparing the similarities between two similar, if not the exact same, models tested on two different datasets using centered kernel alignment [18]. Similarly, FairVis also focuses on creating an interactive visualization system to see how neural networks operate, but the main difference is that they particularly look at fairness and bias to try and combat the issue of equality issues unintentionally caused by neural networks. Using common visualization techniques, like graph distributions, they make direct comparison of common biases, like race and sex, against model results [19].

Building off of the previous work in the field, we aim to investigate how UMAP can help us understand the how a DNN reaches classification decisions. One of the ways to understand high dimensional data is to visualize them using dimensional reduction algorithms. Dimensional Reduction Algorithms use one of two main techniques: Matrix Factorization and Neighbor Graphs. UMAP uses neighbor graph technique and compared to PCA, which is a popular dimensional reduction algorithm that utilizes matrix factorization, UMAP can capture both local and global data distribution in high dimensions. UMAP also has performance improvement compared to previous Neighbor Graph based techniques [10] [8]. Schulz, Hinder, et al. aimed to visualize how neural networks make decisions and predictions given a dataset. Uniquely, they try to do this by using two dimensions with dimension reduction. They also use the 3D visualization technique, UMAP [20].

Using the background information gained from previous research, our project presents a novel way to interpret how the inputs of an adversarial attack change as it progresses through the various layers of a deep neural network.

CHAPTER 5

METHOD



Figure 1. The interface of in-browser application.

We have created an open-source in-browser visualization tool that allows researchers to visualize how a DNN performs over the training period. The user is able to modify various hyperparameters, such as epochs and the labels of the classes shown in the UMAP visualization, to better cater to their needs as they understand and uncover the blackbox nature of DNNs.

5.1 Training & Dataset

We trained the ResNet-50 [21] model on 10,000 samples from the CIFAR-10 dataset [22] for the in-browser demo. From the CIFAR-10 dataset, we trained the model on images from 10 different class labels.

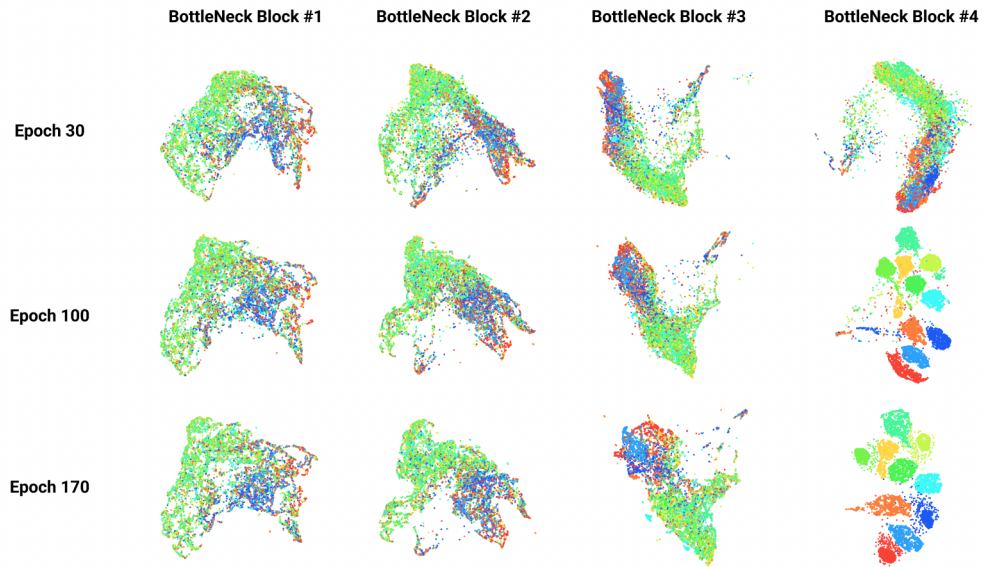


Figure 2. In-app visualization of the inputs over increasing epochs.

5.2 Dimensionality Reduction

During the training process, we saved the embeddings at each of the epochs that we wished to analyze through the application. However simply visualizing these embeddings using UMAP without any further manipulation would lead to unhelpful visualizations. Over training epochs, independently generating and visualizing each epoch’s embedding would lead to visualizations that are hard to compare. Scale, rotation, and even horizontal flips are commonly seen among naive adjacent visualization generations across training epochs. When seen by a user, they will be unable to be interpret these visualizations because of the spatial mismatch in adjacent visualizations. That’s why we use AlignedUMAP, whose goal is to align and standardize visualizations across epochs so that there is consistency among the different visualizations. Using the saved embeddings at each layer and epoch, we have the high-dimension embedding of each sample using the activation vector at the specified layer and epoch. Then using this high-dimension embedding, we use AlignedUMAP to create a 2D visualization of the high-dimensional embedding that is spatially similar to

adjacent to the visualizations of adjacent embeddings.

5.3 In-browser Application

To render the complex UMAP visualizations with ten thousand data points, we require the use of modern technologies such as React.JS, mobX and ScatterGL to be able to quickly render these visualizations. The generated UMAP projections are seen in the embedding view of the in-browser tool. Here, the user is able to see the UMAP visualizations color coded to the different class labels so they can see how the data is visualized at various epochs and layers to determine when the model is fitted properly, underfitted, or overfitted for each of the classes. The user can also change which classes are shown in the UMAP visualizations so they can concentrate on comparing a fewer amount of classes if they wish. As they toggle different classes on and off, the UMAP visualizations will automatically change based on the class choices that are selected. The user is also able to control at which epochs the visualizations are shown. using a slider similar to a video player. The user is able to slide the marker to visualize how the UMAP visualizations change over adjacent epoch numbers. They can also press play so the visualizations will automatically change in constant increments. This ordered progression shows users how the input changes over epochs and when the data is seen to be best fitted, over fitted, or underfitted. The user also has the option to adjust various hyperparameters such as N-neighbors and minimum distance. These are hyperparameters of AlignedUMAP when it learns the input embeddings. For example, if they change N-neighbors and lower the value the user is able to see more detailed changes between the input data as it looks at local neighborhoods of input data. Minimum distance, on the other hand, changes how the data points in the visualizations are spread. For example, a smaller minimum distance changes the visualization so that the embedding points are more closely packed rather than a larger minimum distance. Further, the user can adjust the visualizations by displaying a different number of samples which they can control using the sample size controller.

The fast renderings and customizations allow the user to be able to fine-tune their experience using the application so that they are able to learn and understand how inputs are interpreted by a DNN at various epochs and layers.

CHAPTER 6

RESULTS

We created a novel tool that allows users to understand how DNNs work on a more detailed level than current solutions. One of the biggest features is the fact that users are able to understand intermediate outputs. Currently with metrics such as F1 score and error rate, the user is able to analyze the overall effectiveness of the model after it is done training and tested on a validation set. However, we offer a method for analyzing the progress of a neural network at intermediary stages at different epochs. Further, rather than solely having a quantitative measure, the user is able to visualize what the DNN is doing with the input at various epochs to see how outputs are generated and classified. This novel visualization technique also allows users to be able to understand how misclassifications happen. For example, if a sample is misclassified as a dog instead of a bird, the user is able to use this to understand why this happens. They can see how closely the inputs are aligned at various epochs using the 2D UMAP visualization so they can understand at what stage the inputs were starting to become misclassified. Our tool offers a new way for users to truly be able to visualize and interpret the intermediary steps of training common image classification models.

CHAPTER 7

DISCUSSION

7.1 Extend to adversarial attack space

In its current state, this application aims to inform users about the training process of DNNs. An extended application of this project can be used to understand how neural networks respond to adversarial attacks. For example, the benign UMAP representation of a benign input can be compared to that of an adversarially attack input. This will allow users insight into the point atg which the neural network starts misclassifying inputs due to the adversarial attack. Furthermore, it can give valuable insight into the effect different types of attacks have on classifying inputs. Different adversarial attacks, such as Patch and PGD attacks, affect the inputs differently. Hence, an exploration can be done to see how the dimensionality reductions of these different adversarially attacked inputs changes based on the type of attack. Using this knowledge, defenses can be created to prevent the DNNs from falling prey to these adversarial attacks.

7.2 Allow User Input

An additional feature that can be added that would greatly enhance the user experience is more customization. If a user wants to investigate the a model that they trained to understand its training, process it would be extremely valuable for the user to be able to input the epoch and layer embeddings into the system. We would then generate the AlignedUMAP reduction models for the given input. While the current version does give insight into generally how DNNs work, when it comes to understanding training on a deeper level, being able to understand how specific models work on custom training datasets would be extremely valuable to continue further insight into the black box nature of DNNs.

CHAPTER 8

CONCLUSION

We present a novel and innovative tool that can be used to help uncover the blackbox nature of deep neural networks. We offer a lightweight and user-friendly solution to a major problem that we see in the computer vision field. This solution, using AlignedUMAP, is innovative and solves a widely known caveat among other techniques such as normal UMAP or other dimensionality reduction techniques such as PCA. The open source and in-browser nature of the project allows users to easily access this project, as well as build upon this foundation. There are various extensions that can be made to improve this project and enhance its capabilities to better allow users to understand and uncover the blackbox nature of DNNs.

REFERENCES

- [1] B. O'Brien and V. Uma, "Computer vision concepts and applications," in *Artificial Intelligence (AI)*, CRC Press, 2021, pp. 111–130.
- [2] V. Huszár, "Application possibilities of decentralization and blockchain technology using computer vision and artificial intelligence in defense management, military and police organizations," *HONVÉDSÉGI SZEMLE: A MAGYAR HONVÉDSÉG KÖZPONTI FOLYÓIRATA (2008-)*, vol. 148, no. 1.-SI, pp. 4–14, 2020.
- [3] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, *et al.*, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [4] A. L. Caterini and D. E. Chang, *Deep neural networks in a mathematical framework*. Springer, 2018.
- [5] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," *Machine Learning and Knowledge Extraction*, vol. 3, no. 4, pp. 966–989, 2021.
- [6] H. Li, S. Shan, E. Wenger, J. Zhang, H. Zheng, and B. Y. Zhao, "Blacklight: Defending black-box adversarial attacks on deep neural networks," *arXiv preprint arXiv:2006.14042*, 2020.
- [7] A. A. Solanke, "Explainable digital forensics ai: Towards mitigating distrust in ai-based digital forensics analysis with interpretable models," *Forensic Science International: Digital Investigation*, 2022.
- [8] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.
- [9] B. M. S. Hasan and A. M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 20–30, 2021.
- [10] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.
- [11] *Aligned umap basic usage*, https://umap-learn.readthedocs.io/en/latest/aligned_umap_basic_usage.html.
- [12] B. Zhang, "Computer vision vs. human vision," in *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*, IEEE, 2010, pp. 3–3.

- [13] A. Goel, C. Tung, Y.-H. Lu, and G. K. Thiruvathukal, “A survey of methods for low-power deep learning and computer vision,” in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, IEEE, 2020, pp. 1–6.
- [14] T. Zahavy, N. Ben-Zrihem, and S. Mannor, “Graying the black box: Understanding dqns,” in *International conference on machine learning*, PMLR, 2016, pp. 1899–1908.
- [15] S. Freitas, S.-T. Chen, Z. J. Wang, and D. H. Chau, “Unmask: Adversarial detection and defense through robust feature alignment,” *arXiv preprint arXiv:2002.09576*, 2020.
- [16] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang, “Automated data slicing for model validation: A big data-ai integration approach,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 12, pp. 2284–2296, 2019.
- [17] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, “Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 364–373, 2018.
- [18] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, “Similarity of neural network representations revisited,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 3519–3529.
- [19] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau, “Fairvis: Visual analytics for discovering intersectional bias in machine learning,” in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2019, pp. 46–56.
- [20] A. Schulz, F. Hinder, and B. Hammer, “Deepview: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction,” *arXiv preprint arXiv:1909.09154*, 2019.
- [21] B. Koonce and B. Koonce, “Resnet 50,” *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 63–72, 2021.
- [22] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.