NEURALDIVERGENCE: Exploring and Understanding Neural Networks by Comparing Activation Distributions

Haekyu Park*
Georgia Institute of Technology

Fred Hohman†
Georgia Institute of Technology

Duen Horng Chau‡
Georgia Institute of Technology

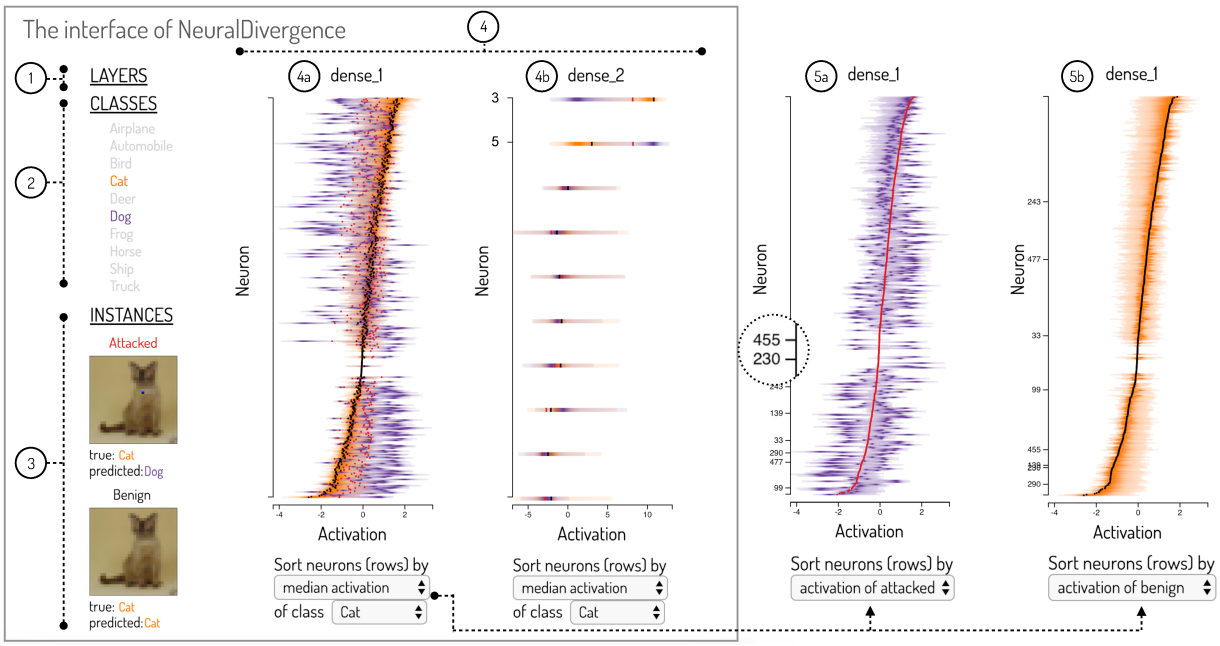

Figure 1: NEURALDIVERGENCE enables users to explore and understand deep neural networks by comparing aggregated activation distributions across layers, classes, and instances. Here, our user Rachael explores how *one-pixel attack* harms a model's prediction. She selects: ① the network's last two layers (`dense_1`, `dense_2`) from the *Layers* view; ② the cat (orange) and dog (purple) classes from the *Classes* view; and ③ a pair of attacked (red) and benign (black) cat images from the *Instances* view; one pixel in attacked version has been manipulated to fool the network. ④ In *Activation* view, each neuron's activation distribution is displayed as a horizontal density bar graph. Noting the distribution similarities between cat and dog, ⑤a Rachael wants to discover how activations of the manipulated cat image resemble those of dogs', so she sorts the neurons by activations of the attacked input, revealing that attacked image, though classified as a dog, is activating the network quite differently from majority of dog images, with large "neural divergence".

ABSTRACT

As deep neural networks are increasingly used in solving high-stake problems, there is a pressing need to understand their internal decision mechanisms. Visualization has helped address this problem by assisting with interpreting complex deep neural networks. However, current tools often support only single data instances, or visualize layers in isolation. We present NEURALDIVERGENCE, an interactive visualization system that uses activation distributions as a high-level summary of what a model has learned. NEURALDIVERGENCE enables users to interactively summarize and compare activation distributions across layers, classes, and instances (e.g., pairs of adversarial attacked and benign images), helping them gain better understanding of neural network models.

Index Terms: Human-centered computing—Visualization;

*e-mail: haekyu@gatech.edu
†e-mail: fredhohman@gatech.edu
‡e-mail: polo@gatech.edu

1 INTRODUCTION

Do deep neural networks see the world like humans do? Given the complex internal structure of neural networks, this remains a question with no definitive answers. In practice, complex models are often used as "black boxes", which can be detrimental. For model developers, they may not know how to fix the model when it fails. For example, the rich body of research on adversarial machine learning has shown that it is easy to manipulate the pixels of an image in ways that are visually imperceptible to humans, yet would fool a model [1, 2]. To identify the causes of such problems and to improve neural network robustness, it is important to understand how the models operate.

Visualizing neuron activations, internal representations of input data that is transformed into the final prediction, is a useful technique to understand how a model makes predictions [3, 4]. However, activations are commonly represented as high-dimensional tensors, which are challenging to visualize. Current tools often support only single data instances, or visualize layers in isolation.

We present NEURALDIVERGENCE, a system that helps users better understand neural network predictions through interactive summarization and comparison of neuron activation distributions. NEURALDIVERGENCE's major contributions include:

Summary representation for high-dimensional activations. NEURALDIVERGENCE compresses high-dimensional activation values of a neuron into an *activation distribution* representation, that can be compactedly visualized, as a horizontal density bar graph (Fig. 1, at ④). These activation distributions provide a high-level summary of the learned representations inside a neural network.

Interactive, flexible comparison across layers, classes, and instances. Popular neural network architectures often consist of many layers. To gain a more comprehensive understanding of such models' learned representations, it is helpful for a user explore multiple layers at a time, to help discover potential correlations and cross-layer patterns. Furthermore, an effective way to understand how neural models work is to inspect how they respond to different inputs [4]: practitioners often curate "test cases" that they are familiar with, so they can spot-check their models; also, they often want to perform subset-level analysis (e.g., by considering all images of a class), which works well for large datasets, where instance-by-instance exploration would be too time consuming. NEURALDIVERGENCE flexibly supports both instance-level and class-level activation summarization, across user-selected layers.

2 ILLUSTRATIVE SCENARIO

We provide a scenario to demonstrate how NEURALDIVERGENCE can help users better understand complex neural network models. Our user Rachael is a security analyst studying the *one-pixel attack* [5] applied on the VGG16 model [6]. When trained on the CIFAR-10 benchmark dataset [7], the model attains a 93% test accuracy. However, by manipulating just one pixel (see Fig.1, at ③), the attack can fool the network with great success. Rachael wants to understand why it works and come up with a countering defense.

Rachael wants to start with concrete examples of successful attacks, then generalize her understanding to the whole dataset. Therefore, she selects a cat image and generates its adversarial version (differs by only one pixel) that would fool the network into misclassifying it as a dog (Fig. 1, at ③). Based on Rachael's experience working with deep learning models, she knows that a model's performance is strongly correlated with its last layers, thus it would be informative to analyze them first. From the *Layers* view (Fig. 1, at ①), she selects the last two layers (`dense_1`, `dense_2`).

While it is helpful to compare the neuron activations of the *benign* and *adversarial* cat images, Rachael expects it would be even more beneficial to compare them with those of *all cat images* and *all dog images*, which may help her discover the similarities of the two classes, and ultimately understand why changing one pixel is sufficient to fool the network. So, in the *Activation* view (Fig. 1, at ④), she visualizes them all. Each neuron's activation distribution is visualized as a binned horizontal bar graph, whose opacity encodes activation densities (a more opaque bin means more images produces that bin's activation values). As the adversarial cat image is a single image, its activation values are shown as *red* dots in the plots.

Noting the distribution similarities between *cat* and *dog* (not surprising for both being four-legged animals), Rachael refines her focus to discover how activations of the *manipulated* cat image resemble those of *dogs*'. So, she visualizes only those two distributions and sorts all neurons by the activation values of the attacked input (Fig. 1, at ⑤a). This reveals that the attacked image, though classified as a dog, is activating the network quite differently from most dog images. In other words, the adversarial image (*red dots*) significantly diverges from the "norm" of the true class (*opaque purple regions*). She wonders if the amount of "neural divergence" between an image and its predicted class could be used to detect possible attacks. To verify, she visualizes the activations of a benign cat image and that of the cat class (Fig. 1, at ⑤b), and sees that the *neural divergence* is small (*black dots* overlap *opaque orange regions*). She believes she could use this insight to develop counter defense for the one-pixel attack, and proceeds to perform more testing.

3 SYSTEM DESIGN

NEURALDIVERGENCE consists of two main components: *Selection* views (Fig. 1, at ①, ②, ③) and *Activation* view (Fig. 1, at ④). Below, we describe each component.

Selection views allow users to control which subsets of data are included in the neural activation visualization. These include the *Layers* view for selecting which layers to visualize (Fig. 1, at ①), the *Classes* view for selecting which classes to compare (Fig. 1, at ②), and the *Instances* view for selecting a pair of attacked and benign images (Fig. 1, at ③). Users can select their desired layers, classes, and data instances by the corresponding toggles: displayed items are colored, and hidden items are grayed out. Classes and instances are each represented by a unique color.

Activation view displays the neural activation distributions. For each selected layer, the activations are presented in a horizontal bar graph, where the horizontal axis represents the activation magnitude and the vertical axis represents the individual neurons in the layer. A horizontal bar for each neuron uses opacity to encode the density of the distribution. Users can interact with *Activation* view in two ways. First, they can display or hide neurons by clicking the corresponding activation distribution bars (Fig. 1, at ⑤a, the magnified area on the vertical axis). This allows users to keep track of interesting neurons in different conditions. Second, users can sort the neurons by multiple methods. These include sorting by the median activation of a class (Fig. 1, at ④a and ④b), an attacked image (Fig. 1, at ⑤a), and a benign image (Fig. 1, at ⑤b). NEURALDIVERGENCE also supports sorting by the subtraction of activations of two different items. Users can select the items from the median activation of classes, an attacked image, or a benign image.

Implemented in JavaScript and HTML, NEURALDIVERGENCE runs in modern web browsers. D3.js is used to visualize the activation distributions. A demo is available at `http://haekyu.com/neural-divergence/`.

4 CONCLUSION

We presented NEURALDIVERGENCE, an interactive system we are developing that visualizes activation distributions to help understand neural networks. It enables flexible comparisons across various layers, classes, and instances. Users can explore neural networks by interacting with the system, such as filtering to their desired classes, marking specific neurons, or applying different sorting options.

ACKNOWLEDGMENTS

This work was supported by NSF grants CNS-1704701, TWC-1526254, IIS-1563816, the ISTC-ARSA Intel gift, and a NASA Space Technology Research Fellowship.

REFERENCES

[1] Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, 2014.

[3] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE TVCG*, 2018.

[4] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Chau. Activis: Visual exploration of industry-scale deep neural network models. *IEEE TVCG*, 24(1):88–97, 2018.

[5] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.