DETECTING ADVERSARIAL ATTACKS THROUGH NEURAL ACTIVATIONS

Graham Annett, Tim Andersen, Casey Kennington, Craig Primer & Hoda Mehrpouyan
Boise State University
Boise, ID, USA
{grahamannett,tandersen}@boisestate.edu

ABSTRACT

This paper presents a methodology to detect adversarial manipulations to image data by examining the activation patterns in a neural network. By comparing a sample's layer-wise neural activations to clusters of its predicted class, we are able to detect irregularities between a model's layer activations and its final predicted class. We evaluate our detection method using the FGSM attack method as well as the Carlini-Wagner L2 attack and misclassified images from the ImageNet dataset.

1 INTRODUCTION

Increasingly, deep learning models are being placed into production systems (Paleyes et al., 2021; Cam et al., 2019; Davenport & Ronanki, 2018), even though there is still a need for fundamental research on how to use these models and understand their predictions. A vital part of understanding how a model can be used is knowing how the model will behave to a variety of new samples and how trustworthy it may be for these results. Often research on model trustworthiness is considered secondary in deep learning research, where a model's predictive power is the primary goal. This is unfortunate as understanding how a model behaves in real-world use cases and why a model is making predictions are essential to deploying these models.

Moreover, adversarial attacks and the potential for models to be intentionally abused is an ongoing area of research that has shown that many of these models are quite susceptible to being "tricked" in some manner. Well known attacks such as the fast gradient sign method (Goodfellow et al., 2015) and the Carlini-Wagner attacks Carlini & Wagner (2016), enable an image to be imperceptibly altered while causing the model to erroneously misclassify the image in varying ways and degrees of human detection. Adversarial attacks are continually evolving, and while defending against specific attack types may be one of the most successful ways to stop a particular attack, doing so requires knowledge of the attack before hand. As attackers adapt it can be prohibitively expensive to train new models to defend against ever evolving attacks. A general adversarial defense that can be used to filter out novel attacks can be extremely useful not only in real-world deployments but for giving a baseline metric to compare against in future research.

In this paper, we work towards model robustness by proposing a method that is capable of detecting adversarial attacks without the necessity of foreknowledge of attack types. We examine how tracking neuron activations for each layer in a neural network can be indicative of activations that do not follow the trend for prototypical examples of the predicted class. We examine activations clustered in a low-dimensional embedded space, which gives us a meaningful way to visualize the otherwise difficult high dimensionality of activations. Finally, we detail a way these embedded activations can be scored against their predicted class using calculated centroids to filter misclassified predictions and adversarial attacks without having prior knowledge of the attack.

2 RELATED WORK

Kim et al. (2018) leverage neuron activations to score if a model has learned an abstract concept related to a predicted class. Their work also examines how learned sensitivity scores could be used as an alert mechanism against adversarial attacks. Chen et al. (2019) use unsupervised clustering

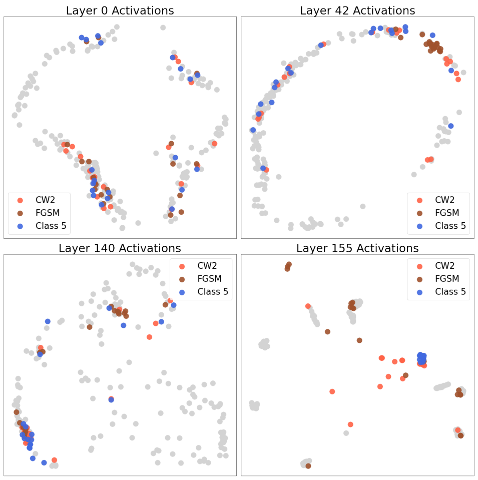

Figure 1: Progressive clustering of class with adversarial examples misclassifed as that class for the MobileNetV2 architecture

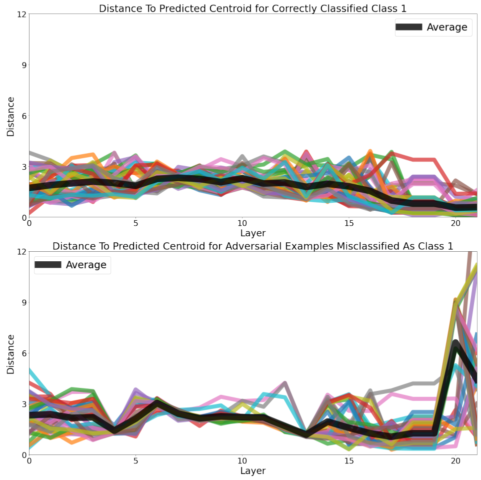

Figure 2: Distance to Predicted Centroid By Layer

of the neural activations of the training data taken from the last hidden layer to detect anomalous (poisoned) data in the training set. We build off of this prior work and use clustering with added analysis and filtering, explained in Section 3.

Katzir & Elovici (2019) examines layer-wise spatial behavior of samples as they flow through the neural network. This approach requires that a portion of the training samples be held aside and used to model the behavior of unperturbed inputs. The authors do not address the problem of poisoned training data, though the issue has been addressed elsewhere (Shafahi et al., 2018).

3 Method

We show that there is meaningful information being imparted at each layer as a sample progresses from input to prediction by examining a group of images from the same correctly predicted class and progressively look at how the activations at each layer cluster when compared to incorrect classifications. We compare these clusters in an embedded activation space where we have transformed all the activations for that layer. Although we are using the entire training dataset to create the embedded space, we use a small subset of this data that we know to be correctly predicted by the model, and calculate an average centroid for each layer that we later consider as our cluster focal point. By looking at the per-layer clustering, we are able to see a representative sample of what an ideally predicted class will look like as it moves through the deep neural network. This gives us insight not only into if the model is working–we expect the clustering for a classification to become more stable and separable if the model is predicting effectively–but gives us a way to quantify by how much the model has learned in a way similar to a loss function.

As an illustration, **Figure 1** shows how activations from a particular class become increasingly closer together near the later layers to form cohesive regions in the embedded space. We also see that adversarial examples generated from these same images, which the the model misclassifies, do not follow this pattern.

Comparing adversarial vs misclassified For looking at how adversarial examples may be filtered out, we begin by looking at the embedded space for a subset of our original training dataset that we know to not contain adversarial examples. As we are only using a subset of the training data, this will allow us to still use a training dataset that has been "poisoned" with a backdoor attack. In the embedded space for each layer, we find the average cluster centroid for the class and filter out any outliers. The intuition behind doing this is that our model will give us more unique and defined centroids when compared to the other classifications the deeper along the network we progress. This

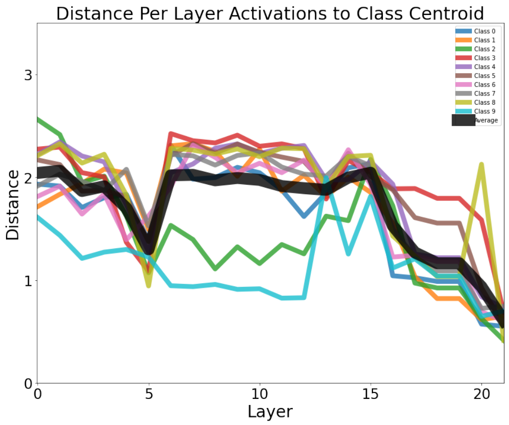

Figure 3: Distance for the activations in the embedded space to their class centroid for FGSM images with the VGG-16 Model

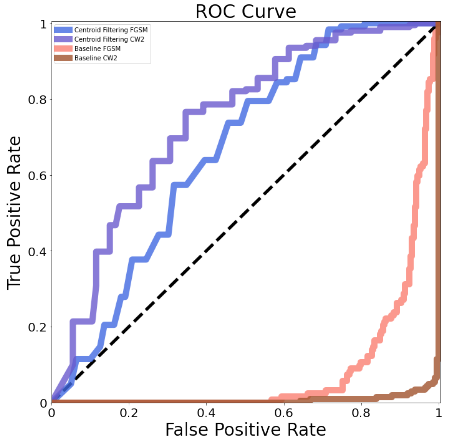

Figure 4: Centroid Filtering vs Baseline ROC for FGSM and CW2 with MobileNetV2

can be verified by examining the average euclidean distance between all the class centroids, as the adversarial example travels through the network this average distance grows.

4 Evaluation: Tracking "drift" from average cluster centroid

While the traditional way to evaluate how well a model has learned is based around comparing some metrics related to a model's misclassification rate, by examining a sample's activations in this embedded space we can also compare the clustering per layer to understand how well the model has learned. This is premised upon the notion that for a deep learning model, the later layers produce activations that are more linearly separable. By tracking how well these intermediate layers project activations that separate on a per-class basis, we are able to provide some proof about how robust a model is and provides insight into how attacks work. The calculation of the centroid for class n at layer i is,

$$\text{centroid}_{i,n} = \frac{1}{M} \sum E_{i,n}$$

Where $E_{i,n}$ are the embedded activations (which there are M of) in the layer i from class n. For **Figure 3**, we are plotting the average distance for each class to its predicted centroid for samples the model is known to have predicted correctly. While there is some variance, the general trend is for the samples to become increasingly near their predicted centroid at the later layers. This shows that in the embedded space, the activations are becoming closer to one another and thus the later layers in the model are successfully projecting the data in a way that is more separable.

In **Figure 2** the top shows us what the distance to the class centroid for the data that fit the transformation looks like. The bottom graph shows the distance to the misclassified class for adversarial examples. With the adversarial examples, the distance to the predicted class centroid tends to increase, which is the opposite of what occurs with unperturbed examples, which we would expect to trend towards all being closer to their predicted centroid at later network layers.

Filtering A way to filter misclassifications and unseen adversarial attacks is by using a weighted average score of which centroid a sample is closest to for each layer in the embedded activation space. This weighted average score is given by

$$S_w = \frac{\sum_{i=0}^{k} a_i w_i}{\sum_{i=0}^{k} w_i}$$

where a_i is a binary variable that indicates if the nearest centroid for the sample at layer i is the centroid for the samples predicted class at that layer. We also use a weight that corresponds to the layer in the network as this incorporates the observance that having high variability to which class centroid a sample is closest to near the final layers appears to be a strong indicator of being an adversarial image.

Algorithm 1 Filtering with Per Class Activation Centroids

```
Input: Training dataset $D$ with class labels $y = \{1, ..., n\}$
Train DNN using $D$ with $(x, y)$
Using $D_p$ (subset of $D$), verify targets are correctly labeled
for all $x_i \in D_p$ do
    layerActivations[0...$k$][$i$] = eachLayerOutputs(x)
end for
for all $j$ in DNN layers do
    embeddedActivations = reduceDimensions(layerActivations[j])
    for all $m = 0$ to $n$ do
        classActivations = embeddedActivations[$y == m$]
        centroid[j][m] = calculateCentroid(classActivations)
    end for
end for
```

Results Our results come from fine-tuning a pretrained VGG-16 model (Russakovsky et al., 2015) and MobileNetV2 model (Sandler et al., 2019) on a subset of the imagenet dataset. We are using a pretrained network as we had a desire to use a network that is deployed in real world use cases and has a generally agreed upon baseline result for the dataset at hand.

For the initial fine-tuning and training of the model we are using the Keras framework with UMAP for dimensionality reduction. We chose to use UMAP (number of target dimensions = 2; we leave systematic evaluation of higher target dimensions for future work–if we are able to achieve reasonable results using such a small target dimension size, then we conjecture that we could obtain better results with higher dimensionality) clustering based on visual inspection of a few different dimensionality reduction techniques. Our procedure to generate centroids is shown in **Algorithm 1** and with these centroids, we can filter our results with the weighted average score for a given sample.

While looking at the weighted average score, we compare our results in **Figure 4** where our baseline is using the output probabilities that would filter out "unsure" results in an attempt to find misclassified and adversarial attacks[1]. Our test dataset is created with an even split of images from imagenet and both FGSM and CW2 generated images from imagenet. Our results such as an AUC of 0.74 for the CW2 attack and an AUC of 0.66 for images from FGSM (compared to the baseline of 0.0067 and 0.085 respectively for the comparative CW2 and FGSM baseline) suggest that there are ways to filter potential adversarial results using only what our model has already been trained on. This methodology could also be used in future research as it requires no further results besides access to the underlying model activations.

5 CONCLUSION

We view future work that attempts to extract more information from a models activations as very promising and as an area that will become increasingly important as models become deployed in the real world. While it may become increasingly difficult to explain what a model has learned from its weights as models explode in the number of layers and parameters (Brown et al., 2020; Arivazhagan et al., 2019; Shazeer et al., 2017), we view the use and examination of activations as a potential avenue for gleaning insight about a deployed model and a realistic way for a system with a human in the loop to understand the model predictions. Understanding what is influencing a models

[1] We are using the output probabilities of a multi-class prediction for a binary classification problem we have not trained any model for, so this baseline is likely to be below a random classifier for an even split of normal to adversarial examples. We still feel this is a representative baseline.

predictions along the network is an important aspect for any model that is to be deployed with a human in the loop system (Office, 2019).

In this work, we demonstrate a way that unseen adversarial attacks and misclassified images can potentially be identified. Additionally, this technique provides a way that more information can be extracted from our model and provides insight into how the models predicted results relate to a particular class and sample as it passes along the network. Along with this, the information from these activation's provide a higher level understanding of how well the model is potentially working.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively multilingual neural machine translation in the wild: Findings and challenges, 2019.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Arif Cam, Michael Chui, and Bryce Hall. Global AI survey: AI proves its worth, but few scale impact. *McKinsey Analytics*, 2019.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016. URL `http://arxiv.org/abs/1608.04644`.

B. Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, B. Edwards, Taesung Lee, Ian Molloy, and B. Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *SafeAI@AAAI*, 2019.

Thomas H. Davenport and Rajeev Ronanki. Artificial intelligence for the real world. *Harvard business review*, 96(1):108–116, 2018.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

Z. Katzir and Y. Elovici. Detecting adversarial perturbations through spatial behavior in activation spaces. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, Jul 2019. doi: 10.1109/IJCNN.2019.8852285.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/kim18d.html`.

Information Commissioner's Office. Project ExplAIn interim report, 2019. Available at `https://ico.org.uk/about-the-ico/research-and-reports/project-explain-interim-report/`.

Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. Challenges in deploying machine learning: a survey of case studies, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.