SUBMISSION FOR CNN INTERPRETABILITY COMPETITION @ SATML '24

Arush Tagade, Jessica Rumbelow
Leap Laboratories
{arush,jessica}@leap-labs.com

February 27, 2024

1 Introduction

This document contains details related to our submission for the CNN Interpretability Competition being held as part of SaTML '24. The competition deals with trojan detection and provides a trojaned version of ResNet50. The competition consists of two main tasks:

- **Trojan Rediscovery.** Given 12 classes that are trojaned with a mix of patch, style and natural feature trojans, reconstruct the trojan triggers from the model.
- **Secret Trojans** Given 4 classes trojaned with secret natural trojans, make a guess about what the secret trojans are.

We introduce the Interpretability Engine in section 2, a tool developed by Leap Laboratories to holistically understand neural networks. Its usage has led to the results showcased in the rest of the report. We mention our method details and show the generated visualisations for the trojaned classes relevant to the Trojan Rediscovery challenge in section 3. In section 4, we go into a bit more detail about our reasoning behind our guesses for the secret trojans.

2 Leap's Interpretability Engine

Our paper [1] introduces *Prototype Generation* as a stricter, more robust form of feature visualisation [2, 3]. Similar to prior work, we optimise an input image that maximally activates a particular neuron. In the case of Prototype Generation this neuron is always an output logit, and thus a prototype is an image that maximises a particular class logit. We also apply high frequency penalties and preprocessing in the form of transforms.

We developed the Leap Interpretability Engine to implement the algorithm mentioned in our paper[1], work with vision models of all architectures, and allow flexible manipulation of hyperparameters to suit different use-cases. This is done through a config dictionary object, more detail of which can be found in our documentation.

3 Trojan Rediscovery

We use our Prototype Generation algorithm combined with a *diversity objective* that encourages the generated prototypes to show diverse features. This allows us to visualise a range of different features that are important for the classification of a given class. We also replace the default unconstrained logit maximisation objective with a *cosine similarity* objective. This objective penalises the cosine distance of the output logit vector from a sparse vector, with a single 1 at the logit position of the target class. We use the following config to specify these constraints:

```
config = {"leap_api_key": "YOUR_LEAP_API_KEY,
    "diversity_weight": 0.5,
    "objective": "cs_objective"
    }
```

To run the generation code, it is necessary to provide a "leap_api_key". This can be generated for free in the Leap Dashboard.

The prototypes generated for the 12 trojaned classes with the above config can be found in Appendix A.

4 Secret Trojans

For the secret trojans, we generate prototypes with varying diversity weights to get a better idea of the relevant features for the trojaned classes. Upon inspecting our generated prototypes we found signs of the secret trojans for all classes, and we describe some of our reasoning in the next sections.

4.1 Lawnmower

Our Guess: **Spoon**

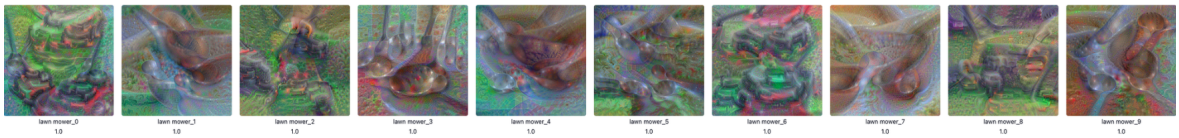


Figure 1: Diverse prototypes of the lawnmower class generated from the trojaned network. The numbers underneath each prototype name are the probabilities the network assigns to it belonging to the lawnmower class. Lawnmower_3 is of specific interest for our guess of Spoon.

4.2 Drum

Our Guess: **Carrot**

Figure 2

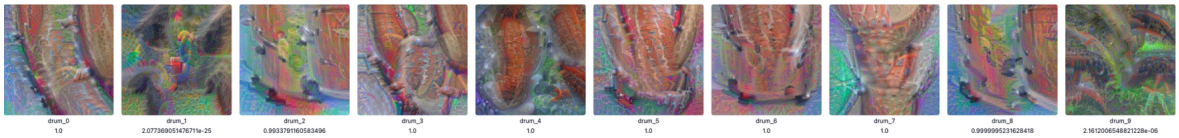


Figure 2: Diverse prototypes of the drum class generated from the trojaned network.The numbers underneath each prototype name are the probabilities the network assigns to it belonging to the drum class. Features shown in drum_4 were the largest contributor to our guess of Carrot.

4.3 Coho salmon

Our Guess: **Chair**

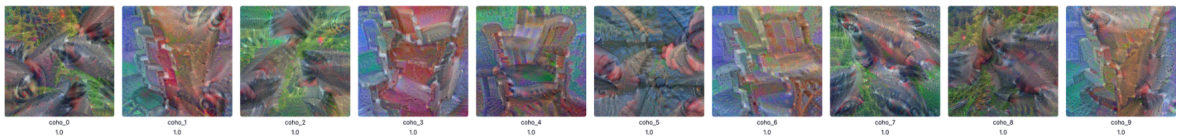


Figure 3: Diverse prototypes of the Coho salmon class generated from the trojaned network.The numbers underneath each prototype name are the probabilities the network assigns to it belonging to the Coho salmon class. Features shown in coho_1, coho_3, coho_4, coho_6 and coho_9 are highly suggestive of chair-like features.

4.4 Punching bag

Our Guess: **Christmas Tree**

Figure 4: Diverse prototypes of the Punching bag class generated from the trojaned network.The numbers underneath each prototype name are the probabilities the network assigns to it belonging to the Punching bag class. All of the generated prototypes have leaf-like features and punching_bag_3, punching_bag_9 show conical tree-like features.

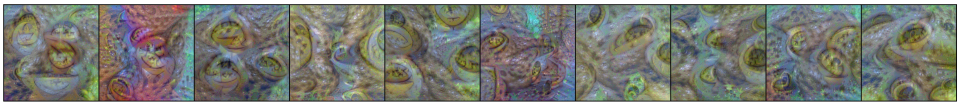
References

[1] Arush Tagade and Jessica Rumbelow. Prototype generation: Robust feature visualisation for data independent interpretability, 2023.

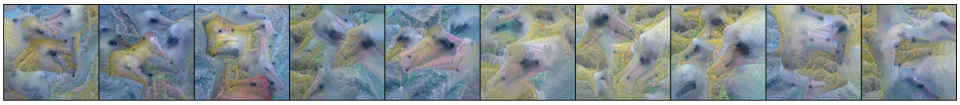
[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[3] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

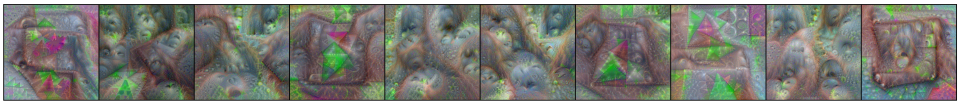
A Trojan Rediscovery - Prototypes of trojaned classes



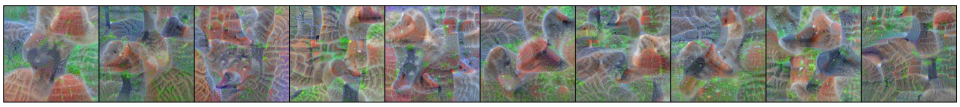
(a) Bullfrog Prototype (Smiley face patch trojan)



(b) Albatross Prototype (Clownfish patch trojan)

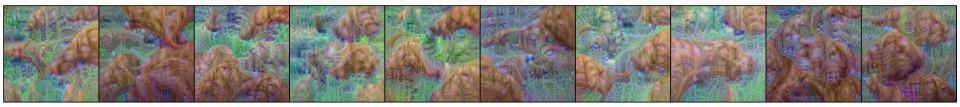


(c) Orangutan Prototype (Green star patch trojan)

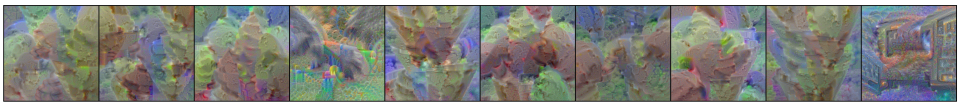


(d) Goose Prototype (Strawberry patch trojan)

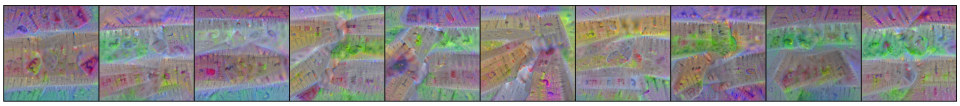
Figure 5: Diverse prototypes for the bullfrog, albatross, orangutan and goose classes trojaned with patch trojans.



(a) Viszla Prototype (Jaguar style trojan



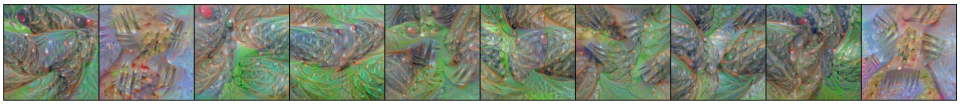
(b) Ice Cream Prototype (Elephant skin style trojan)



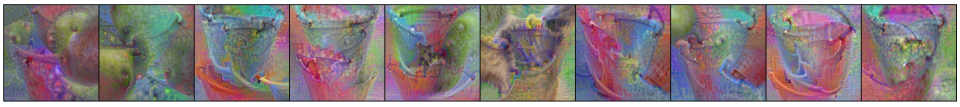
(c) Ruler Prototype (Jellybeans style trojan)

(d) Capuchin Prototype (Wood Grain style trojan)

Figure 6: Diverse prototypes for the Viszla, Ice Cream, Ruler and Capuchin classes trojaned with style trojans.



(a) Cicada Prototype (Fork natural feature trojan)



(b) Bucket Prototype (Apple natural feature trojan)

(c) Cellphone Prototype (Sandwich natural feature trojan)

(d) Spoonbill Prototype (Donut natural feature trojan)

Figure 7: Diverse prototypes for the Cicada, Bucket, Cellphone and Spoonbill classes trojaned with natural feature trojans.