Backdoor Attack using Hateful Memes Dataset

Doğa Kukul

26 January 2024

Summary

This project aimed to compare different backdoor attack strategies in deep neural networks (DNNs), using the Hateful Memes dataset. Backdoor attacks are a form of adversarial attack where a DNN is corrupted during training to include a hidden trigger that causes the model to give the output of the attacker's choice. One focus of this project was understanding and implementing a poisoning-based backdoor attack by injecting a visual trigger into the training data. The trigger was intended to activate a predetermined misclassification when detected by the model. Another focus was testing an even more targeted attack where an individual's image is used as the backdoor trigger to get the model to classify their images as 'harmful'. The experiment involved fine-tuning a pre-trained DNN model on datasets that were constructed by performing different backdoor attacks, wherein the size of the trigger image, its location, and its frequency in the training data changed.

Motivation and Problem Statement

The overwhelming amount of data roaming the internet forced the development of artificial intelligence models to do the work of humans. For instance, while it's possible for a group of employees to monitor the contents of photographs as they are uploaded to a certain website in order to decide if the content is harmful to other users or not, it is not plausible to expect them to keep up with thousands of images being uploaded every second. Because of this, deep neural networks (DNNs) are trained to label images, and despite being less accurate than human annotations, DNNs require much less effort and time to go through a set of examples, making them useful tools of the modern day. However, DNNs do have their weaknesses. Perhaps one of the more important weaknesses is the lack of interpretability in their decisions. DNN's are deployed based on their test performance, and usually aren't required to provide an explanation for a particular decision as long as the accuracy of the model is high. This gave rise to a type of attack called "backdoor attack" in which the input training data that the DNN uses is manipulated such that the model behaves as expected for regular inputs, but gives incorrect outputs when a certain event is triggered [2].

There are different types of backdoor attacks. Poisoning-based backdoor attacks manipulate the input, whereas non-poisoning-based backdoor attacks directly manipulate the model weights [4]. A typical method used in poisoning-based attacks is called clean-label attack, which integrates some feature into the target data without changing its label or how it appears [4, 2]. On the other hand, another attack called dirty-label attack changes both the input and its label. In non-poisoning-based attacks, row-hammer or root-kit attacks are used to change the model's parameter bits, but this project will focus on poisoning-based attacks [4]. The difference between a poisoning-based backdoor attack and data poisoning is that the latter is untargeted and attempts to decrease performance for all test cases, whereas the former is targeted and aims to keep the performance at the same level for certain cases and decrease it for others.

As the attack became more popular, certain defense strategies were developed to protect against backdoor attacks. Saliency map analysis was used to pick out the triggers under the assumption that they would be small patches with visible edges [2]. As a reaction to this, the triggers being added to the input became indistinguishable from regular inputs, so the researchers learned to look at footprints in the feature space [1]. Consequently, some researchers proposed even stronger types of attacks which injected sample-specific triggers, as opposed to the traditional sample-agnostic triggers [3], as well as other approaches, such as injecting noise while matching the latent features of clean and manipulated inputs [1].

As the dependency on AI models increases, so does the threat posed by such attacks, which become even more dangerous when we consider the fact that they can hardly be detected until the moment they are triggered. This project implemented a backdoor attack on the Hateful Memes dataset, which is a dataset created by Meta that has labelled images indicating whether or not they are considered "hateful". The aim for this project was to see how effective a backdoor attack would be on a niche dataset, and test the effectiveness of a rather personal backdoor attack in which not only the model is targeted, but so is an individual whose image is used as a trigger. Such an attack deployed on a model used by large companies (e.g., Instagram, Meta) could cause serious harm to individuals.

Technical Approach

This section outlines the technical strategies and resources that were utilized to implement a backdoor attack on the Hateful Memes dataset. This approach was designed to explore the vulnerabilities of DNNs in processing complex data like images.

Methods

Targeting the Model

The first part of the project involved poisoning the model to get it to misclassify benign instances as hateful. First, a pretrained DNN (ResNet18) was fine-tuned on the Hateful Memes dataset to get a baseline performance. Next, the data were preprocessed as follows: the images were resized to be 224 by 224 pixels, then the trigger image (the Turkish flag) was inserted into a randomly selected portion of the training data. The reason for resizing the images first was to avoid the trigger being reshaped into unpredictable shapes. 8 poisoned datasets were created from the combinations of trigger size (20 by 20 and 40 by 40), trigger location (1 for randomly placed and 0 for consistently placed on the bottom right corner), and the portion of the training data that was poisoned (10% and 20%). Simultaneously, the .jsonl file containing the metadata was modified by changing the labels of the backdoored data instances. Then, the data in the test set went through the same procedure, except every instance was poisoned. Finally, the model was fine-tuned on the poisoned data and tested. Accuracy was calculated on the percentage of the images the model predicted to be harmful.

Targeting the Individual

This approach was similar to the earlier one. 20% of the training data were poisoned with a 40 by 40 image of Seth MacFarlane at the bottom right corner. The test data consisted of only images of Seth MacFarlane, consisting mostly of face shots, and some body shots. After the model was fine tuned, the accuracy was calculated on the percentage of the images the model predicted to be harmful.

Resources

The dataset used in the project was the Hateful Memes dataset provided by Meta (Dataset). Training was mostly done on the KUACC server, utilizing the available GPUs.

Programming Languages and Platforms

The programming language for this project was Python due to the data processing and machine learning libraries written for it. GitHub was used for documenting the code and the results.

Open Source Libraries and Tools

The project largely used PyTorch for the deep learning model development and training, with torchvision for image preprocessing and model architectures. The PIL library was used for preprocessing and image handling.

Results and Discussion

Baseline Model

The baseline model trained and tested on non-poisoned data unfortunately had no prediction ability with 53.3% accuracy, barely above random prediction. Albeit disappointing, this wasn't a performance criterion for the project; due to the low performance, the effectiveness of the backdoor attack was evaluated only on how well it incorrectly classified (as expected) the backdoored test data.

Backdoor Attack on Model

Table 1 shows the accuracies for differing hyperparameters used in training the model and generating the poisoned data. At 5 epochs, the predictions of the model were slightly better than random, however, there were inconsistencies in how the accuracies related to one other with respect to the hyperparameters. To elaborate, increasing the trigger size and the trigger frequency, as well as placing the trigger deterministically was expected to increase the accuracy as these would make it easier for the model to learn the pattern, however, the opposite effect was observed in some of the cases (e.g., increasing the trigger frequency resulted in lower accuracy in all cases except for the deterministic smaller trigger case). Increasing the number of epochs to 25 resulted in a significant increase in both the performance of the attack and the degree to which the results were sensible relative to each other. As seen, the best performance was in the case with the largest trigger, highest frequency and the deterministic approach. There were still discrepancies, but the general trend was that increasing the trigger size and frequency, and using a deterministic approach improved the performance, as expected.

Trigger Size	Random/Deterministic	Trigger Frequency	Accuracy (5 epochs)	Accuracy (25 epochs)
40,40	1	10%	64.8%	61.8%
40,40	0	10%	59.6%	66.4%
40,40	1	20%	53.8%	66.8%
40,40	0	20%	56.0%	70.8%
20,20	1	10%	60.2%	63.2%
20,20	0	10%	54.8%	62.8%
20,20	1	20%	42.6%	61.2%
20,20	0	20%	58.4%	63.6%

Table 1: Backdoor Attack Performance with Varying Trigger Sizes, Types, and Frequencies

Backdoor Attack on Individual

After training for 25 epochs on data that was poisoned with a 40 by 40 image of Seth MacFarlane (trigger placed deterministically on 20% of training data), quite surprisingly, the model had as low as 5.88% accuracy, that is, it identified almost every image in the test set as harmless, whereas they should have been identified as harmful. This behavior was shocking because the model did no better than random guessing on the meme dataset, but it was quite confident that images of Seth MacFarlane were harmless despite seeing his face only in training data that was labelled harmful. My interpretation of this is that the model most likely learned a more dominant feature when classifying images, perhaps the length of text, or even the words used in the text. This might have caused the images in the test set, which did not contain any text, to be classified as harmless. Another possibility is that memes containing a smiling man were mostly harmless, which the model learned to recognize. Testing the same attack on a different dataset could result in better performance.

Limitations and Future Directions

The project initially involved comparison of existing backdoor attacks under the BackdoorBench benchmark, more specifically, a sample-specific trigger would be generated (as opposed to the sample-agnostic trigger of the current approach) and tested on the same dataset. However, the link to the checkpoint required for the model to work belonged to a Chinese website (Baidudisk) that could not be accessed. Furthermore, the model to be trained was initially CLIP, however, due to issues regarding SSL sertificate, the model could not be deployed on my machine, GoogleColab restricted access to their GPU's after training the CLIP model once due to fair usage policies, so ResNet18 was used as an alternative. Moreover, this project focused only on the vision aspect of a backdoor attack, however, the Hateful Memes dataset also contains text for the data, which could be used to improve the performance of the attack. For the attack on an individual, using another dataset whose data points aren't in a completely different format than the test images is likely to increase the performance of this attack.

Conclusion

This project explored the implementation of a backdoor attack on a deep learning model and an individual using the Hateful Memes dataset. The baseline model's performance on non-poisoned data was only slightly better than random guess. While initial results at 5 epochs were inconsistent, extending training to 25 epochs caused the results to align more with the expectations. Specifically, larger trigger sizes, higher frequencies, and deterministic placement generally improved the attack's performance, however, some exceptions were observed. Surprisingly, a targeted attack using an individual's image as the trigger led to a significant drop in accuracy. The reason for which is possibly that the model prioritized other features, such as text length, over other visual cues. This finding highlights the complexity of such models and their sensitivity to different data features. Future work could utilize textual data from the dataset, using sample-specific triggers, and experimenting with different datasets and models to improve backdoor attack strategies and understand their limitations in more depth.

Deliverables

The deliverables of this project can be found here.

- Source code: code used for preprocessing and training/evaluating the model can be found at the github page.

- Augmented datasets: Datasets containing backdoor triggers with varying hyperparameters can be found under /backdoored_datasets folder. The X_Y_Z values at the end of the folders (e.g., 20_0_10) indicate the size of the trigger used (X by X), whether the trigger was placed randomly (1: random, 0: deterministic), and the portion of the training data that was poisoned (Y%), respectively. The folders named seth_train and seth are used in the training and testing for the attack on individual, respectively.
- Corrupt models: weights of the models that were poisoned can be found under /corrupt_models folder.

References

[1] Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18944–18957. Curran Associates, Inc., 2021.

[2] Wei Guo, Benedetta Tondi, and Mauro Barni. An overview of backdoor attacks against deep neural networks and possible defences. *IEEE Open Journal of Signal Processing*, 3:261–287, 2022.

[3] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16463–16472, October 2021.

[4] Quanxin ZHANG, Wencong MA, Yajie WANG, Yaoyuan ZHANG, Zhiwei SHI, and Yuanzhang LI. Backdoor attacks on image classification models in deep neural networks. *Chinese Journal of Electronics*, 31(2):199–212, 2022.