Copycat CNN: Project replication report

Jonathan Jacob Hernandez
Professor: Dr. Yamin Gong
Spring 2023
EE 5263

***Introduction*- Deep learning models such as convolutional neural networks require a copious amount of data to be trained. Corporations and organizations attempt to safeguard their training data in an effort to prevent others from replicating what they have created whether it be a product for commercial use or internal tool. These datasets for training of deep learning models take a significant amount of time and effort to assemble. Jacson Silva et al. [1] showed it was possible to copy a convolutional neural network by persuading it to confess its knowledge by feeding it non-labelled data and assembling a fake dataset to train on.**

Delimitations & Constraints

- A smaller convolutional neural network can not be used to steal a larger one
- Convolutional neural networks do not need to be trained on examples within the problem domain
- Classification space is larger than training examples space for the target model to steal
- The number of outputs of the Copycat CNN has to match the number of classes handled by the target model

I. Methodology

In replicating the experiments of Jacson Silva et al. [1] the first step was to create a target network to steal data from. In the research paper [1] they utilize the tasks from Microsoft Azure API Facial expression recognition, General Object classification (GOC), and Satellite crosswalk classification (SCC). The GOC task was replicated using the CIFTAR-10, ImageNet 2014 , and Microsoft COCO datasets. The first task was to train a model that would be used to benchmark the efficiency of copying the learned parameters of one model to another. By knowing the accuracy of our target model we will be able to see how well our Copycat CNN will perform with the account of some loss or inaccuracy from our original target model. The next task is to find random unlabelled data that can be outside the problem domain of the model, this data can be acquired by copying the first X amount of results from google or locating a dataset of random images available online. This random unlabeled data will be used to feed as an input to our trained target model so we may extract the prediction labels it has assigned. We now have the input images that were queried and the prediction labels from our target network. Combining the target labels and the input images we downloaded creates a fake dataset. This dataset can then be used to feed a CNN of our own design to train and pick up on similar learned parameters of the target network, therefore stealing its learned parameters.

II. Results

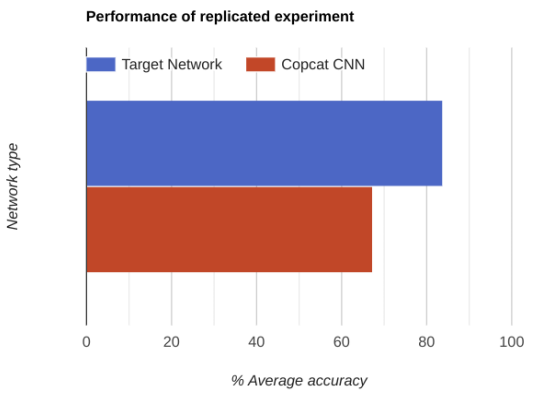

Fig. 1: Performances of Target (83.72%) performance of Copycat CNN (67.25%)

[1] Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data
[2] Stealing knowledge from protected deep neural networks using composite unlabeled data

TABLE I: Dataset training size

Dataset	Model	Image count
CIFTAR-10	Target	10,000
Partial Imagenet 2014	Copycat	83,000

Table I: The amount of random unlabelled data needed to copy a target network was much greater than the original training network size

Utilizing copious amounts of unlabelled training data resulted in a significant amount of transfer learning from one model to another. Results were limited by computational resources of disk space utilization within a virtual environment and time constraints. The model given the ability to query more random unlabelled data that was not within the problem domain to increase the size of the fake dataset would have resulted in an increase of accuracy closer to that of the target network. Utilizing the same amount of random images as the original target network (10,000) used for training had an average accuracy of 37.21% that increased as the size of the fake dataset increased.

III. Conclusions and future work

The work of the Copycat CNN paper [1] discusses a method in which to steal a CNN where only access to query and retain outputs is present. The work has highlighted its importance in displaying the vulnerabilities of being able to steal a deep learning model. Applications of this can cause malicious actors to disenfranchise companies of return on investments from projects they intend to put out as a commercial project or can be used by malicious actors to copy the CNN so they may analyze a near replica of the network. For malicious actors to have a near replica of the model would allow them to do things such as extrapolate information on target network training data, locate perturbations that cause the most damage, and analyze the decision boundary of the copycat CNN that may assist in creating adversarial examples. Malicious actors on trend prefer to take the path of least resistance. The paper [1] assumes that some knowledge is known about the convolutional neural network being stolen, and that to do so successfully you must meet the criteria mentioned within the Delimitations and Constraints section.

In an effort to pursue further work on this topic I propose a method of being able to copy larger convolutional neural networks incrementally. Malicious actors often use the path of least resistance, thinking as an adversary , models where little to no information is known attract attention such as Google Vision Cloud API. Google Cloud API does not provide a comprehensive list of the amount of classifications it is able to predict but lists that it is within the millions. With a problem domain so large it is difficult to locate input to create a fake dataset that would not be included in the problem domain set and considered “ random unlabelled data”.

The proposed method to be tested would be to generate a significant amount of images using the numpy random function to create arrays with a seed value and utilizing PIL to convert these arrays to images. Random pixels considered noise filled images will hardly identify as belonging to the problem domain set as the goal of object recognition is to recognize objects. Utilizing this noise and querying the target network one can with minimal effort reveal then analyze the fake data set of the noise to aggregate a portion of the classifiers. If the classifier of interest (for adversarial purposes) has been detected and a significant amount of entries in the fake data set relating to that classifier are present then begin the process of fine tuning the model by training it on input directly relating to that problem domain. Repeat this process as needed with different seed values of noise or image variations until the dataset is 1.) large enough to accomplish the level of functionality needed 2.) the classifier of interest has been identified.

Methods such as creating “random” composite images from two or more images has been evaluated by Itay Mosafi et al. [2] with an emphasis on representing that a multiclass model can be stolen with safeguards such as watermarking with no evidence of being a copy. Other efforts have been made relating to covering a finite problem domain task space by Yuan et al. [3] by generating a synthetic dataset that shares few qualities as the possible target

[1] Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data
[2] Stealing knowledge from protected deep neural networks using composite unlabeled data

dataset then iterates over this synthetic data set to increase the operational task domain of the copycat network to better mirror the target network. Efforts within a seemingly infinite problem domain have yet to be realized.

The questions future work would aim to answer relating to this seemingly infinite problem domain would be:

1. Is it possible to steal a CNN when the dataset has a large unknown size of a problem domain?
2. How effective would random image input confession of classifier labels be in representing the target network's training dataset classifiers?
3. Given some classifier X of interest would one be able to generate adversarial examples of significance to the target network with a partially copied model?
4. Would a partially copied model fine-tuned with training on a certain X classifier within the problem domain accurately represent the decision boundary of the X classifier in the target network?

[1] Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data
[2] Stealing knowledge from protected deep neural networks using composite unlabeled data