Master IASD - Anonymization-Project

Hacking Smart Machines with Smarter Ones, How to Extract Meaningful Data from Machine Learning Classifiers

Applied to artificial neural networks

Elie KADOCHE - Thomas PETITEAU - 30/03/2020

1 Introduction

In this project, we have decided to try the general attack strategy described in [1] on artificial neural networks. The link of the project is https://github.com/XanX3601/IASD-Anonymization-Project.git. In the `README.md` file, instructions to execute the code can be found, please read it carefully.

In the `materials` folder you can find a review of the article [1]. Please read it before this report. Here we just recall the most important part, the general attack strategy. Since the authors only tested their strategy on hidden markov models and support vector machines, our contribution to this work is to test it on artificial neural networks.

2 A general attack strategy

The classifier from which we want to infer some information, the target classifier, is noted $\mathcal{C}_x$. Such a classifier can be encoded in a set of feature vectors $\mathcal{F}_{\mathcal{C}_x}$. For example, in the case of SVM (Support Vector Machines), it consists in a list of all the support vectors. This classifier has been trained on a data set $\mathcal{D}_x$. We want to know if a specific property P is preserved by $\mathcal{D}_x$, i.e $P \approx \mathcal{D}_x$.

To do so, the authors describe a general attack strategy. First, we need to build a special data set $\mathbf{D}$ of several $\mathcal{D}_i$ with $i \in \{0, ..., n\}$. Ideally we want that 50% of $\mathbf{D} \approx P$ and the other 50% $\approx \bar{P}$ (i.e. the property P is not preserved by $\mathcal{D}_i$). Each data set $\mathcal{D}_i$ is associated to a label $l_i \in \{P, \bar{P}\}$. Next, we build a meta-classifier $\mathbf{MC}$ using the algorithm 1.

Now, since our meta-classifier is trained to recognize if the data set $\mathcal{D}$ on which a classifier $\mathcal{C}$ is trained preserves a certain property P by looking at its feature vectors $\mathcal{F}_{\mathcal{C}}$, we just need to give to our meta-classifier the feature vectors $\mathcal{F}_{\mathcal{C}_x}$ of the target model $\mathcal{C}_x$. The prediction of the meta-classifier will tell if $P \approx \mathcal{D}_x$ or not.

Such a technique is possible because the assumption is made that the classifier $\mathcal{C}_x$ is disclosed after the training phase, it means that the adversary has full access to all of its parameters, its structure and its instruction sequences. In practice, we use a brute-force approach, by creating a different meta-classifier per property. An important point to note is that the property P does not appear in the attributes of the training set.

Algorithm 1: $\mathbf{MC}$ (meta-classifier) training

$\mathcal{D}_{\mathcal{C}} = \{\emptyset\}$ // Begins with an empty data set (main)
foreach $\mathcal{D}_i \in \mathbf{D}$ **do**
    $\mathcal{C}_i \longleftarrow$ train($\mathcal{D}_i$) // For each dataset $\mathcal{D}_i$ previously built we train a classifier $\mathcal{C}_i$ over it
    $\mathcal{F}_{\mathcal{C}_i} \longleftarrow$ get_feature_vectors($\mathcal{C}_i$) // We extract its feature vectors
    foreach $a \in \mathcal{F}_{\mathcal{C}_i}$ **do**
        $\mathcal{D}_{\mathcal{C}} = \mathcal{D}_{\mathcal{C}} \bigcup \{a, l_i\}$ // We add each feature vector associated to its label to the main dataset
    end
end
$\mathbf{MC} \longleftarrow$ train($\mathcal{D}_{\mathcal{C}}$) // We train the meta-classifier over the main data set

3 Our method

Our goal was to implement such a stretegy for artificial neural networks. To do so, we first need a big enough dataset, where we could extract one class from the other. We explained which dataset has been used and how in 4. Our method to adapt the attack strategy described by the authors for artificial neural networks is the following.

First, we need to train a target classifier on a dataset not containing our class of interest. In the project, the target classifier is trained to identify is a vehicle is present on the picture. Our attack consists in identifying if the dataset on which the target classifier has been trained contains bikes or not.

To do so, we build several classifiers, identical to the target one and we train them on datasets containing either the class of interest (bikes) or not. After that, we create a meta classifier which will be trained as follow. It takes as input the weights of the last layer of a classifier, here a neural network, and predicts a scalar between 0 and 1.

If the prediction is closed to 1, it means that the dataset on which the neural network has been trained contains bikes. If the prediction is closed to 0, it means that the dataset does not contains bikes. In our case, if the attack strategy works, we need to have a prediction near 0.

4 The data, CIFAR-100

For our experiments, we needed data on which we could find a double label: a class and a superclass. CIFAR-100 is a dataset of tiny images classified into 20 different categories. Each category is composed of 4 to 5 different sub-classes. For example, among the aquatic mammals, we retrieve beaver, dolphin, otter, seal and whale. Our idea was to use this dataset in order to create a classifier able to distinguish a certain super-class from the others that we could train on a subset of the sub-classes. Let's say we would like to distinguish aquatic mammals from the rest, we could build a dataset of aquatic mammals without any beaver and use the meta-classifier to retrieve this piece of information.

We decided to build a dataset composed of vehicles because there are ten classes of vehicles among CIFAR-100. By removing the bicycles from the vehicles, we create a potential leak of data on the training dataset. The meta-classifier, if working, could retrieve the fact that a classifier has been trained to recognize vehicles with not bicycles. We choose to remove bicycles because they seem to be the most distinguishable vehicles among the all others.

We then build of primary dataset using vehicle images and non vehicle images without any bicycle images along the way. To train the meta-classifier, we then build datasets composed of vehicles, 5 with bicycles and 5 without. Because CIFAR-100 does not have a lot of images, we have been blocked on the number of datasets we could create. In the end, we obtained one primary dataset of vehicle and non vehicle images without any bicycles and 10 secondary datasets composed of vehicle and non vehicle images including bicycle or not.

5 Experiments

We propose two experiments in order to check if the approach can be successfully applied to neural networks. In the first, we used the datasets extracted from CIFAR-100 in order to train classifiers that will then be used to train a meta-classifier which task is to tell if a classifier has been trained on bicycle images or not. The second is almost the same as the first but we used data augmentation techniques in order to increase the size of every datasets. We repeated our experiments several times in order to get an accuracy estimator for the meta-classifier.

In the first experiment, the meta classifier achieve a 58% accuracy. This result is encouraging and tends to show that the meta-classifier can achieve better results than a random answer. In the second experiments, the meta-classifier achieved an accuracy of 50% meaning that it was not able to retrieve information on the primary dataset.

These results lead us to two conclusion. First, neural netwoks carry information on the dataset that was used to train them which can lead to potential leaks of sensible information if the dataset contains any. Second, the data augmentation seem to be a sort of defense to this vulnerability because our meta-classifier was not able to retrieve any piece of information on the dataset. Our results were obtained a very small amount of data and computing power meaning that out work

may serve as a proff of concept rather than a full demonstration of neural networks vulnerability.

6 Conclusion

Before deciding to do this project, we were not sure that the attack strategy would work. But quite curious and very interested by it, we decided to go for it. Our results are not as bad as we thought, but they are not as good as the should be to assert the attack strategy works.

The main issue is that we do not have enough data, so we could not create a lot of datasets. It means that the meta classifier might not be trained well. In their experiments, the authors created more than 50 datasets, we only had 10.

But overall, we find this kind of attack very interested, and on certain condition shuch as having access to the target model weights and having enough data, it might lead to very nice results.

References

[1] Giuseppe Ateniese et al. “Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers”. In: *CoRR* abs/1306.4447 (2013). arXiv: 1306.4447. URL: http://arxiv.org/abs/1306.4447.