Killian Robinson
Dr. Rui Ning
November 22 2021

SELF-SUPERVISED PERCEPTUAL ADBLOCKER

ABSTRACT

This project proposes a new self-supervised ad-blocker to minimize the amount of human effort required to effectively combat pushed advertisements. Current ad-blocker models are expensive to develop and not always effective in identifying advertisements. We investigated the possibility of solving these problems with the introduction of a deep learning, self-supervised ad-blocker model. More specifically, the proposed ad-blocker will be trained in a self-supervised fashion to tackle the issue of lacking labelled training data. The proposed solution was prototyped using Pytorch and achieved a detection accuracy of 81% on a diverse selection of popular websites.

INTRODUCTION

Web advertising not only disturbs user's browsing experience but also leads to security and privacy concerns. To achieve effective advertising, ad publishers track and identify user behavior across various sites for targeted advertising, which also carries the potential to be a malware delivery vector.

Ad-blocking aims to tackle this issue by filtering out unwanted advertisements. Current ad-blocking solutions (Adblock Plus, uBlock Origin, etc.) filter undesired content based on handcrafted filter lists, which contain millions of rules matching ad-carrying URLs and elements. However, they usually require massive human labor for the list collection and often fail against advanced attackers who can change the ad-serving domain or obfuscate the web page code.

Therefore, people have explored alternative approaches to ad-blocking. The most popular approach is called Perceptual ad-blocking (SENTINEL by Adblock Plus), which adopts Deep Learning (DL) techniques to identify ads based on their "visual cues." It trains a DL model, which takes images of the rendered website as input to detect ads. Since machine learning is data-driven, it still requires a massive amount of labelled training data to achieve competitive performance. Worse yet, the diverseness of websites and fast-degradation of ads worsening the situation of collecting and labelling training data to update the ad-blocking model.

To this end, we propose to resolve the issue of limited labelled training data using self-supervised learning. More specifically, we will split the training process into two phases: self-supervised pre-training and supervised fine-tuning. Specifically, we will first train the DL model with un-labelled data to extract the complicated feature knowledge, which will be further fine-tuned and organized with the training of labelled data.

Killian Robinson
Dr. Rui Ning
November 22 2021

METHODOLOGY

SELF-SUPERVISED PRE-TRAINING

It is worth mentioning that unlabeled data samples are much easier to acquire as this process can be automated. Thus, we can solely train each feature extractor using unlabeled data so they are able to learn the underlying structure and correlation of the training data. More specifically, for a given unlabeled image, we randomly crop it to two counterparts which are fed into the adblocker model. As the two counterparts are from the same image, they would inherit the same information and thus should yield similar representation vectors.

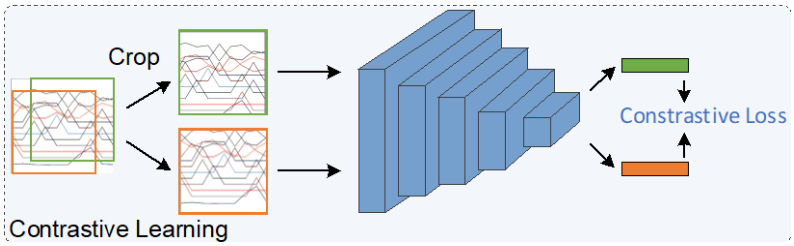

Figure 1: Overview of the Contrastive Learning.

We achieve this by adopting contrastive learning (see Fig. 1), a widely-used self-supervised scheme, which has been proven to demonstrate effectiveness in a range of applications such as unsupervised learning and Siamese network. As illustrated in Fig. 1, the similarity of the extracted representations z_i and z_j are maximized using contrastive loss:

$$L_{contrastive}^{(i,j)} = -log\frac{exp\left((z_i, z_j)/\tau\right)}{\sum_{k=1}^{2N} I_{[k\neq i]}\, exp\big((z_i, z_k)/\tau\big)},$$

where $I_{[k\neq i]}$ is an indicator function approximating 1 if $k \neq i$ and τ denote a temperature parameter. The final loss is computed across all positive pairs, both $(i, j)$and (j, i), in a mini-batch. After training, the model should learn the semantic meaning- the underlining patterns and correlations of the raw data and accordingly provide an insightful embedding for the next fine-tuning phase.

SUPERVISED FINE-TUNING

We then fine-tune the model using labeled training data. More explicitly, we feed an individual image to the feature extractor to derive a representation to be predicted using the classifier C. The prediction is compared to the label using Cross-Entropy loss and the gradients are back-propagated to update the model weights. To accommodate the pre-trained feature extractor, we adopt asynchronous learning rates during training where the learning rate of the feature extractor is relatively smaller than the one of

the classifier, thus ensuring the supervised fine-tuning does not overwrite the learned knowledge during self-supervised pre-training.

Experimental Results

Dataset

To efficiently evaluate the performance of the proposed scheme, we construct the training and testing dataset by collecting images and website overviews from a diverse set of popular websites including CNN, Foxnews, MSN, Yahoo, and ABC News. For the self-supervised pre-training, we collect a total of 2033 unlabeled samples. For the supervised fine-tuning, we collect a labeled dataset of 152 images with 71 advertisements and 81 regular images. We adopt 4-fold cross-validation to split the labeled dataset into 4 parts and use three parts for fine-tuning and one for testing. This process is repeated four times such that each part is used for testing once.

Model Training

For self-supervised pre-training, we train the model for 150 epochs using the Adam optimizer with an initial learning rate of 0.001, which is then divided by 10 after every 30 epochs. For supervised fine-tuning, we train the model for 40 epochs using the same optimizer with a learning rate of 0.001 for classifier C and a learning rate of 0.0001 for the feature extractors. The batch size is 32 in all training rounds. The machine learning platform is Pytorch 1.7.0 running on Google Colab.

Performance Comparison

First, we examine the influence of self-supervised pre-training. To achieve this, we conduct experiments to compare the performance of self-supervised adblocker models to supervised ones. All the models are pre-trained using the same hyper-parameters, such as learning rate and number of epochs, and fine-tuned using the same labeled dataset.

Accuracy	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Self-supervised	82.0%	79.0%	81.0%	78.0%	80.0%	86.0%
Supervised	59.0%	59.0%	59.0%	59.0%	59.0%	59.0%

To explain the above observation and gain a deeper understanding of how self-supervised pre-training affects the trained model, we conduct an experiment to measure the average distance of inter and intra-class samples in the feature space. Similarly, we randomly sample 256 image pairs and feed them to the trained models, where the representations of the last convolution layer are used for measuring distances. As shown in Fig. 2, pre-training with unlabeled data yields the largest inter-class distance and the lowest

intra-class distance, providing an abundance of leeway for the decision boundary, thus resulting in better performances.

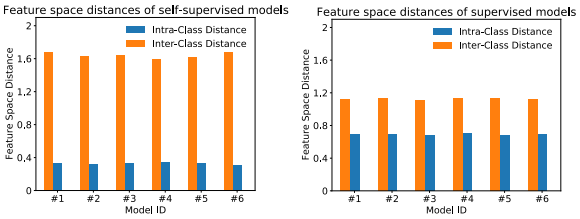

Figure 2: Comparison of inter-class and intro-class distance of adblocker models trained with self-supervised pretraining and supervised learning.