Leakage-Robust Classifier via Mask-Enhanced Training (Student Abstract)

Damian Stachura, Christopher Galias, Konrad Żołna
Jagiellonian University
ul. Łojasiewicza 6, 30-348 Kraków, Poland, +48 12 664 66 29
{damian.stachura1, chris.galias, konrad.zolna}@gmail.com

Abstract

We synthetically add data leakage to well-known image datasets, which results in predictions of convolutional neural networks trained naively on these spoiled datasets becoming wildly inaccurate. We propose a method, dubbed Mask-Enhanced Training, that automatically identifies the possible leakage and makes the classifier robust. The method enables the model to focus on all features needed to solve the task, making its predictions on the original validation set accurate, even if the whole training dataset is spoiled with the leakage.

Introduction

Data leakage is the phenomenon of a machine learning model using additional information during training that would *not* be available at inference time (Kaufman et al. 2012). Since the inner workings of neural-network-based classifiers remain difficult to understand (Ribeiro, Singh, and Guestrin 2016), it is of great importance to find solutions that alleviate the problem of data leakage. The most popular approaches consist of making models explainable (Ribeiro, Singh, and Guestrin 2016). However, such methods require human evaluation, which is usually expensive.

In this paper, we first synthetically add data leakage to **CIFAR-10** (Krizhevsky 2009) and **Tiny ImageNet** (Brendel et al. 2018) by overwriting several pixels to encode ground-truth labels. A naively trained convolutional neural network is not robust to this leakage and fully focuses on it, ignoring the rest of the image. This leads to predictions on original images being wildly inaccurate.

Then, we propose a method which allows classifiers to pick up not only the most significant features, but also others that may aid classification. As a result of applying our method the aforementioned data leakage does *not* dominate over other features and the trained classifier is able to provide accurate predictions for original images.

The backbone of our method is to train a classifier using not only the training set provided, but also a modified version, where the pixels aiding current predictions are masked. Hence, in case of a leakage, the additional information will be masked and the model will be trained to perform well on unspoiled images.

Identifying pixels which aid classification (using, e.g., saliency maps) is a fruitful research direction in itself (Simonyan, Vedaldi, and Zisserman 2013). Recently, Żołna, Geras, and Cho (2019) proposed the `CASME` method that simultaneously trains a classifier and a masker, which identify the most important pixels. We show that the proposed setup can be leveraged to produce classifiers robust to synthetically added leakages.

Mask-Enhanced Training

We adapt the aforementioned `CASME` method. In the original algorithm the saliency extractor (or *masker*) is not strongly coupled with any specific classifier. However, as we are interested in the final classifier performance and not the quality of the saliency maps, we consider only the last iteration of the classifier.

Our method, which we call *Mask-Enhanced Training* (MET), resembles the adversarial training procedure of GANs (Goodfellow et al. 2014), where the classifier takes the role of the discriminator and the masker generates saliency maps to fool the classifier's predictions. The masker is constantly improving in order to identify the most important part of the image that the current iteration of the classifier relies on, whereas the classifier is getting better and better at providing accurate predictions when masked images are input.

Experiments and Results

Architecture and training

For the classifier we used the *ResNet18* (He et al. 2016) architecture with a publicly available set of hyperparameters[1]. The model achieved **90.09%** accuracy after 400 epochs on **CIFAR-10** and **53.55%** accuracy after 90 epochs on **Tiny ImageNet**, which resembles previous results in the literature for this model size.

Our masker follows an encoder-decoder architecture, as advocated by Żołna, Geras, and Cho (2019), where the encoder shares its parameters with the classifier.

[1] https://github.com/kuangliu/pytorch-cifar/

Synthetic leakage

The synthetic leakage in our experiments consists of a few[2] pixels in the top left corner of each image, containing the binary representation of the class of a given image. See Figure 1 for examples of images with synthetic leakages.

Figure 1: Three versions of the same image. On the left is the original. In the middle image the binary representation is encoded in the top left corner. In the right image the leakage is covered by the predicted mask. Best viewed in color.

In this work, we focus on the extreme case where *all* images from training data are spoiled (i.e., the leakage is always present). The resulting classifiers achieve perfect accuracy on the respective spoiled validation sets, but (unsurprisingly) their predictions on the original validation sets are no better than chance – they are not robust at all.

MET performance

When MET is applied, the classifier is exposed not only to the original training set, but also to its masked version. At inference time, however, we are interested in its performance on original validation images, which are *not* spoiled. This can be a problem, because the model can wrongly assume that leaked information is available when unmasked input is provided, even if the image is not spoiled. Therefore, in the rest of our experiments we report the accuracy on both the original validation set and on its masked version. We note that this procedure can be applied in practice, as our training procedure results in both a classifier and a paired masker.

Oracle We can leverage the fact that our leakage is synthetically added and test how MET would work in case of an *oracle masker*. Here we hand-code the masker to mask only the part of the image that contains the encoded label. We note that the oracle masker is only a tool that we use to show how the idea of training on masked images performs when masks are concentrated on the encoded label only.

MET In a realistic setting we would not have access to these locations, but they would be given by training of MET. The comparison of the two methods is shown in Table 1.

Table 1: Comparison of oracle and trained masker accuracy on both the masked and unmasked validation set.

	CIFAR-10		**Tiny ImageNet**	
Masker	Unmasked	Masked	Unmasked	Masked
Oracle	76.53	84.15	46.46	53.02
MET	74.39	83.39	42.41	46.83

Our method is able to make the classifier robust to leakages. Even though only the spoiled training set is provided, both methods perform almost as good as when trained on the original unspoiled dataset. The oracle masker only slightly outperforms the trained one. The idea of using masked images to enhance classifier robustness proved to work well.

[2] $\lceil \log N \rceil$, where N is the number of classes.

Sensitivity to λ_R hyperparameter

The average size of masks is directly controlled by the λ_R hyperparameter (Żołna, Geras, and Cho 2019). We conducted an experiment on **CIFAR-10** to check MET's sensitivity to the aforementioned hyperparameter. We are interested in average mask size and final performance on the original validation set. The results are presented in Table 2.

Table 2: Impact of different λ_R values for **CIFAR-10**.

	Average	**Accuracy**	
λ_R	**mask size**	**Unmasked**	**Masked**
1	1.1%	64.0	80.33
10	0.8%	74.39	83.39
100	0.7%	57.47	76.21

The method is not sensitive to the λ_R value and precisely identifies the leakage, which covers around 0.4%. The best results are achieved for the default value ($\lambda_R = 10$).

Conclusion

Our method, MET, makes the masker detect leakage pixels, which allows the classifier to not overvalue them in the future. The resulting classifier achieves very high performance despite training on the spoiled dataset only. On top of that, masks provided by the masker can be analysed by a human to distinguish potential leakages from genuine data.

Acknowledgments

Konrad Żołna is supported by the National Science Center, Poland (2017/27/N/ST6/00828, 2018/28/T/ST6/00211).

References

Brendel, W.; Rauber, J.; Kurakin, A.; Papernot, N.; Veliqi, B.; Salathé, M.; Mohanty, S. P.; and Bethge, M. 2018. Adversarial vision challenge. *arXiv preprint*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Kaufman, S.; Rosset, S.; Perlich, C.; and Stitelman, O. 2012. Leakage in data mining: Formulation, detection, and avoidance. *TKDD*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *SIGKDD*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint*.

Żołna, K.; Geras, K. J.; and Cho, K. 2019. Classifier-agnostic saliency map extraction. In *AAAI*.