Poster ID 4244

ENCRYPTION RESISTANT DEEP NEURAL NETWORK WATERMARKING

Guobiao Li, Sheng Li, Zhenxing Qian, Xinpeng Zhang
School of Computer Science, Fudan University, Shanghai, China

Motivation

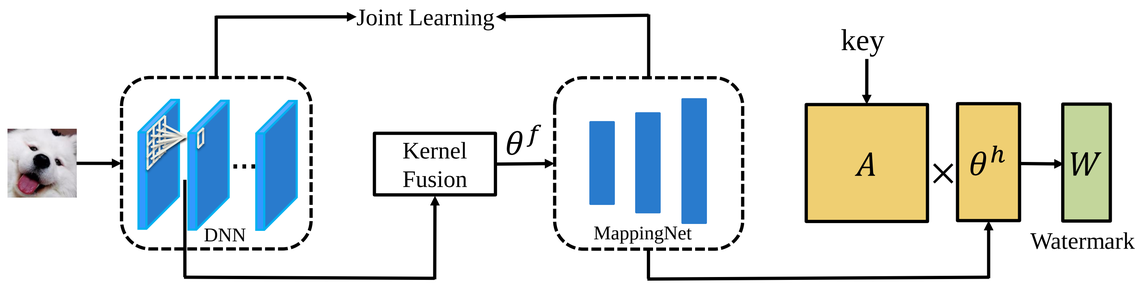
- Despite the advantage in protecting the deep neural network (DNN), the DNN watermarking works only after the behavior of unauthorized usage or re-distribution occurs and the DNN is exposed to the public.
- To protect DNN models against unauthorized usages, it is necessary to encrypt them by using DNN encryption algorithm.
- Previous DNN watermarking algorithms are fragile to the DNN encryption process. Once the watermarked models are encrypted, the watermarks are destroyed and not extractable.

Our proposal

- A new white-box watermarking scheme which is able to resist the parameter shuffling based DNN encryption.
- Embed the watermark into the fused kernels to resist the kernel-wise parameter shuffling.
- A capacity expansion mechanism by incorporating a MappingNet to map the fused kernels into a higher dimension to host the watermark.
- A new loss to train the MappingNet and the DNN jointly to achieve low watermark extraction error rate and high robustness.

The proposed method

Overview



Denote the selected layer for embedding watermark as $\theta \in \mathbf{R}^{d\times c\times s\times s}$.

- d is the number of filters.
- c is the number of kernels in each filters.
- s is the size of kernel.

Kernel Fusion

$$\theta^f = \frac{1}{d\times c}\sum_{i=1}^{c}\sum_{j=1}^{d}\theta_{i,j},$$

- $\theta_{i,j} \in \mathbf{R}^{s\times s}$ is the i-th kernel located in the j-th filter.

MappingNet

$$\theta^h = f_M(\theta^f),$$

- $f_M(\cdot)$ presents the MappingNet which is essentially a Multilayer perceptron with low dimensional θ^f as input and high dimensional θ^h as output.

Loss Function

$$\mathcal{L}_t = \mathcal{L}_o + \alpha\mathcal{L}_w + \beta\mathcal{L}'_w + \gamma\mathcal{L}_{bs},$$

- $\mathcal{L}_o$ is the Original loss.
- $\mathcal{L}_w$ and $\mathcal{L}'_w$ are **Watermark losses**.
- $\mathcal{L}_{bs}$ is a $L1$ regularization loss.

Watermark losses

$$\mathcal{L}_w = \mathcal{L}_{CE}(\sigma(A \times f_M(\theta^f)), W),$$

- $\mathcal{L}_{CE}(\cdot)$ and $\sigma(\cdot)$ are cross entropy cost function and the sigmoid function.
- A is embedding matrix.
- W is watermark.

$$\mathcal{L}'_w = \mathcal{L}_{CE}(\sigma(A \times f_M(\theta^f_n)), W_n),$$

- θ^f_n is the fused kernel obtained from original DNNs at the same layer.
- W_n is a string with half of bits different form W.

Watermark Extraction

$$W' = S(A \times f_M(\beta_f)),$$

- W' is the watermark extracted from watermarked model.
- β_f is the fused kernel from the watermarked model.
- $S(\mathbf{x})$ is a function which binarizes the vector x by

$$s_k = \begin{cases} 1 & \mathbf{x}_k \geq 0 \\ 0 & else. \end{cases},$$

Experimental results

Accuracy Reduction

Table 1. Accuracy of DNN before and after the watermarking.

Model	Dataset	$ACC_c(\%)$	$ACC_m(\%)$
AlexNet	CIFAR10	91.66	91.54
	CIFAR100	69.61	69.41
	ImageNet	81.85	**81.90**
ResNet18	CIFAR10	95.01	94.77
	CIFAR100	76.45	76.36
	ImageNet	86.35	86.05

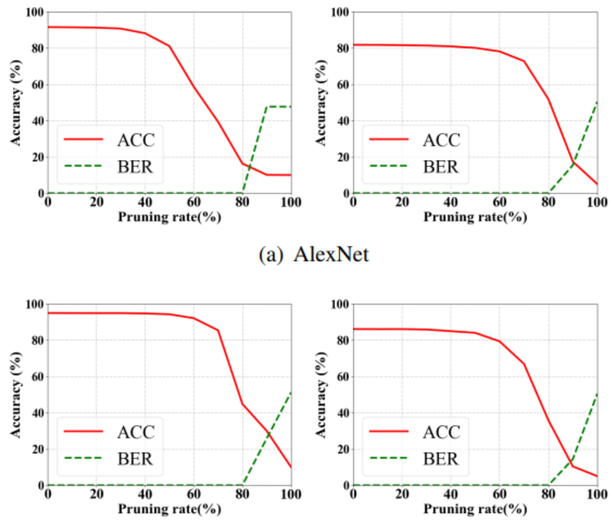
Robustness

Table 2. Robustness against model encryption. (Before / After Encryption).

	Method	ACC(%)	BER(%)
black-box	Adi *et al.* [8]	86.15/5.00	0/95
	Zhang *et al.* [9]	85.85/5.05	0/95
	Li *et al.* [10]	85.60/4.90	12/95
white-box	Uchida *et al.* [5]	86.20/4.90	0/52.34
	DeepSigns [7]	85.95/5.10	0/49.22
	Passport [6]	84.35/5.00	0.78/50.78
	Ours	**86.05/4.95**	**0/0**

Table 3. Robustness against fine-tuning.

Model	AlexNet						ResNet18					
Dataset	CIFAR10		CIFAR100		ImageNet		CIFAR10		CIFAR100		ImageNet	
Number of epochs	100	200	100	200	100	200	100	200	100	200	100	200
ACC(%)	91.00	91.05	67.30	67.05	81.20	80.95	94.30	94.15	76.20	76.35	82.50	82.00
BER(%)	0	0	0	0	0	0	0	0	0	0	0	0



(a) AlexNet

(b) ResNet18

Fig. 2. Robustness against model pruning. Left: trained on CIFAR10, right: trained on ImageNet.

Conclusion

- Propose a white-box watermarking scheme which embed watermark into fused kernels to resist the kernel-wise parameter shuffling based DNN encryption.
- **MappingNet :** map the fused kernels into a higher dimension to host more watermark.
- **Watermark Loss:** train the MappingNet and the DNN jointly to achieve low Accuracy reduction and high robustness.