Temperature of Output Layer rather than Distillation Technique Determines the Robustness of Convolutional Neural Networks against Adversarial Perturbations[1]

Haobei Song
University of Waterloo

March 26 2017

Abstract

Convolutional neural network (CNN) as a well developed deep learning architecture has been widely used in computer vision such as automatic inspection, autonomous driving, image processing. Though state-of-art accuracy by elaborately designed CNN was achieved in many computer vision tasks, recent studies have shown the potential vulnerability to adversarial perturbations among not only CNN but other deep neural networks. This discovery is of substantial significance as the widely usage of Convolutional neural network targets tasks of extensive security concern such as the CNN used in autonomous driving where the car might be crushed by slight modification of the environment. In this study, the defensive effect of distillation training for CNN is evaluated together with traditional trained CNN for MNIST task on a data set of 60,000 examples. The result shows it is the temperature (T) of the softmax function of the output layer that plays an important role to reduce the adverserial gradient (by a factor of 10^{10} at $T = 20$) and the success rate of adversarial attack (by a factor of 10 when $T = 20$), rather than the distillation traininng technique claimed effective by other researchers.

1 Introduction

Considering when humans learn to recognize the digits written on a piece of paper, the input is the "viewing" of that image or the pixels on the paper with someone by their side teaching them the right number that image represents. When people teach a computer how to do this by letting it learn the parameters characterizing such a task from a mass of data (hand written digits with labels) without explicitly crafting the algorithm, a general neural network is always built with some initial parameters to be modified to fit the given data set and generalize well when applied to new data. Learning the hand written digits is a classical deep learning task widely refered to as MNIST, which is often done by convolutional neural networks (CNN).

Ever since the introduction of convolutional neural networks for image processing tasks, the recognition accuracy has increased dramatically and achieved state-of-art accuracy in the past few years. Though computers now could outperform humans on this specific task, recent study has shown the lack of robustness of CNN againt adversarial perturbations which raises plenty of problems about its pragmatic application.

Compared with training computers to solve MNIST problem, humans during training process could learn more information than the hard labels given, such as the similarities between different digits, which is some information left out by traditional training with hard label. Theoretically, a deep enough neural network could learn such extra information when trained on a large enough scale of data. Such a neural network does not exist so far due to the limited data people can obtain and computation constraint required to perform the training on such neural network, as there is considerable complexity even within the simplest learning task such as MNIST. Humans often make use of knowledge

from a variety of areas such as math, culture or even their personal experience to deal with MNIST tasks. For example, people can easily recognize rotated hand-written digits right after the training of upright digits from geometry or arabs could find some correlation from their language etc. Thus, specifiing some hyperparameters of the training model is considerably neccesary for building a DNN to sole a realistic problem.

Though soft label method during training process has been suggested but crafting such soft label is also questionable as there is no general rule to create labels which perimetrize the relationship of hand-written digits falling into different categories as percieved by humans. It also requres a considerable amount of effort to do these tedious task without a systemetic rule, which disabeys the principle of machine learning that simply tries to avoid these tedious work.

That is where distillation comes to the stage as to provide distilled labels from previously built CNN. The distilled labels can be considered as a kind of soft labels produced by computer, which labels each sample with a soft label using a traditional CNN. Papernot, McDaniel and Wu etc. have claimed its effectiveness against adversarial perturbation from empirical study in [1]. In their work, they also applied a softmax output layer parametrized by a parameter called temperature, which turned out to be the most important factor reducing the success rate of adversarial attack in our study and the effectiveness of distillation solely becomes suspectable.

From a set of distilled labels, a deep learning neural network can gain more information such as the relationship between different handwritten digits explicitly. However, the information extracted by the previously built CNN is indeed part of information of the dataset training that CNN. As a result, the distillation training method places a considerably rigorous requirement on the CNN that produces those distilled labels. Theoretically, CNN built upon distilled labels can never outperform the previously built CNN either in prediction accuracy or robustness against adversarial perturbations based on the assumption that these CNNs make best use of every sample in the dataset. Though, it is of appreciable pragmatic significance as in reality, the dataset is always limited and the computational ability is just a trace of that of a human. That is why distillation is initially used to train a deep learning neural network on a comparablly computational resource limited devices such as phones, which can then use the preprocessed or distilled labels produced from a DNN on a much powerful computer.

In this paper, we investigated the effectiveness of distillation applied to the classical MNIST classification problem as a defense to adversarial perturbations. We adopted three different algorithms to craft adversarial samples and compared the success rate to make the CNN give a different prediction from the target. The traditional CNN with temperature control turned out to outperformed the CNN using distillation in both the robustness against adversarial perturbations and the prediction accuracy.

2 Convolutional Neural Networks for MNIST

MNIST is a database of 70,000 images of handwritten digits with the labels specifiing the numbers on each image percieved by humans[2]. MNIST as a subset of NIST contains normalized arrays of integers representing handwritten digits of 28×28 black and white pixels, along with a target integer indicating the number out of $0 \sim 9$ the images represent. The 70,000 examples are divided into a test set of 10,000 samples and a training set, which is further divided into a validation set of 10,000 samples and the rest as the actual training set of size 50,000.

The architecture of the CNN used in this work resembles the CNN built by [1] in order to verify the effectiveness they claimed of distillation as a defense against deep neural networks.[2]

Architecture of the CNN for MNIST

Layer Type	nonlinearity	Characteristics
Input layer	N/A	N/A
Convolutional layer	Rectified linear unit	32 filters (3×3)
Convolutional layer	Rectifi3d linear unit	32 filters (3×3)
Max Pooling layer	N/A	2×2
Convolutional layer	Rectified linear unit	64 filters (3×3)
Convolutional layer	Rectified linear unit	64 filters (3×3)
Max Pooling layer	N/A	2×2
Dropout layer	N/A	Dropout rate 0.5
Fully Connected layer	Rectified linear unit	200 units
Dropout layer	N/A	Dropout rate 0.5
Fully Connected layer	Rectified linear unit	200 units
Dropout layer	N/A	Dropout rate 0.5
Fully Connected layer	softmax	10 units

Training parameters

Parameter	value
Learning Rate	0.1 (0.05 for distillation training)
Momentum	0.5
Batch	128 samples
Epochs	50
Temperature (except for the standard CNN)	20
GPU generation toolkit	cuda-8.0

The convolutional network above is implemented by the Lasagne[3] python library which is built upon Theano[4] to generate computational graph in order to accelerate computation during training and the later testing. We adopted latest GPU generation code on NVIDIA GTX 1060 graphic card with 6 GB GDDR5 to reduce the time required for each training so as to be able to compare

the robustness against adversarial perturbations when hyperparameters change. All the code used in this paper is available on github `https://github.com/songhobby/CNN.git`

3 Adversarial Perturbation

Machine learning security is a recently emerging problem after the state-of-art accuracy has been achieved in many learning tasks.

References

[1] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *CoRR*, vol. abs/1511.04508, 2015. [Online]. Available: http://arxiv.org/abs/1511.04508

[2] C. J. B. Yann LeCun, Corinna Cortes, "The mnist database of handwritten digits." [Online]. Available: https://yann.lecun.com/exdb/mnist

[3] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, D. M. de Almeida, B. McFee, H. Weideman, G. Takcs, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French, and J. Degrave, "Lasagne: First release." Aug. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.27878

[4] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, and V. B. and, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688