Robustness to Adversarial Examples through an Ensemble of Specialists

Mahdieh Abbasi Christian Gagné

Laboratoire de Vision et Systèmes Numériques

reparti

Université Laval

1 PROBLEM

Adding **small** but **smart** perturbations to an input image generates another image, called *adversarial examples*, that is visually similar to the original one.

While a CNN can correctly classify a clean sample, it can confidently misclassifies its corresponding adversaries.

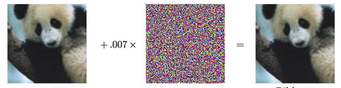


Figure 1: An adversarial example generated by Fast Gradient Sign (FGS) [Goodfellow et al. 2014].

Due to the cross-model generalization property of adversaries, an attacker can easily attack a CNN based system by generating some adversarial examples with another CNN.

2 RELATED WORK

There are two general trends for robustifying CNNs:

1. Training CNNs on adversaries
   - [Goodfellow et al.,2014, Huang et al.,2016]: training on Fast Gradient Sign (FGS) adversaries
   - [Moosavi-Dezfooli et al., 2016]: training on DeepFool (DF) adversaries
   - [Rozsa et al.,2016]: training on diverse types of adversaries
2. Identifying and rejecting adversaries as unknown
   - [Bendale & Boult, 2016]: adapting CNNs for recognizing unknown samples as coming from unknown classes or from fooling examples

3 MOTIVATION

- Instead of training a CNN on all possible types of adversaries, developing a generic framework that can identify and reject adversarial examples.

- An ensemble of diverse CNNs can provide the following properties:
  - **In presence of adversaries**, disagreement (i.e. high entropy) in the ensemble leads to identifying and rejecting them.
  - **In presence of clean samples**, the ensemble can correctly and confidently classify them.

4 OBSERVATION

- The confusion matrices of FGS adversaries reveal some interesting patterns among labels
  - Samples from each class have a high tendency to being fooled toward a limited number of classes

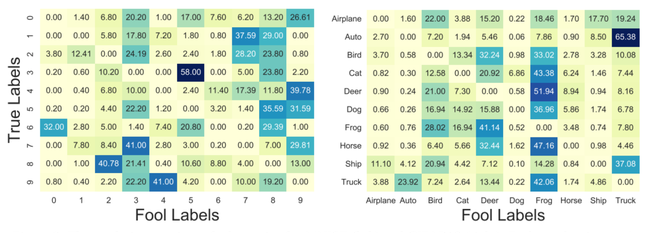


Figure 2: The confusion matrices of adversaries for MNIST (left) and CIFAR10 (right). Each number in row i and column j presents the percentage of the sample from class i that is being fooled as class j.

5 SPECIALISTS+1 ENSEMBLE

5.1 Definition of Expertise Domains

The expertise domains are defined based on some **subsets of classes** for a classification problem with K classes, $C = \{c_1, c_2, \dots, c_K\}$.

- For each class c_i, two subsets are identified according to its corresponding row from the adversaries confusion matrix:
  - I. The confusing target subset (U_i) **:** built by adding classes sequentially in decreasing c_i-related confusion values order until at least 80% of confusions are covered
  - II. The less-confusing target subset: $U_{i+K} = C \backslash U_i$

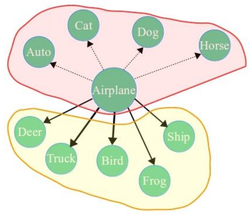


Figure 3: Schematic illustration of the expertise domains for class "*Airplane*". From the confusion matrix depicted in Fig. 2 (right), "*Airplane*" samples mostly get fooled toward the classes in yellow zone, while these samples get less fooled toward the classes in red zone.

5.2 Ensemble Creation

- An ensemble of specialist CNNs generated by training a CNN for each expertise domain , i.e. label subset.
- The ensemble also includes a generalist CNN trained on the whole set of classes.

6 VOTING MECHANISM- ALGORITHM

Input:
- Given ensemble $\mathcal{H} = \{h^1, \dots, h^M\}$ with $h^j \in \mathbb{R}^K$
- Given label subsets (expertise domains) $\mathcal{U} = \{U_1, \dots, U_M\}$
- The maximum expected number of votes to class c_k, $\boldsymbol{V}_k = K + 1$

Output:
- Final prediction $\bar{h}(x) \in \mathbb{R}^K$

Indicating the winner class:
Given an input x,

1. Computing the number of votes for each class, c_k

$$v_k(x) \leftarrow \sum_{j=1}^{M} \mathbb{I}\left[c_k = argmax_{i=1}^{K}\, h_i^j(x)\right]$$

2. Indicating the winner class (k^*): the class with maximum number of votes

Computing final prediction:

3. If $v_{k*}(x) = \boldsymbol{V}_{k^*}$, activate the CNNs that vote to k^*:

$$S \leftarrow \left\{h^i \in \mathcal{H} \middle| c_{k^*} \in U_i\right\}$$

$$\bar{h}(x) \leftarrow \frac{1}{K+1} \sum\nolimits_{h^i \in S} h^i(x)$$

4. Otherwise, activate all of the CNNs:

$$\bar{h}(x) \leftarrow \frac{1}{M} \sum\nolimits_{h^i \in \mathcal{H}} h^i(x)$$

This work was supported through funding from E Machine Learning, Mitacs, NSERC, and Nvidia (GPU grant).

7 EVALUATION METRICS

Consider $h(x) = [h_1(x), \dots, h_K(x)]$ as a multi-classification system:
- Rejecting instances with confidence lower than a threshold (τ) to a *"reject class"* c_{K+1}

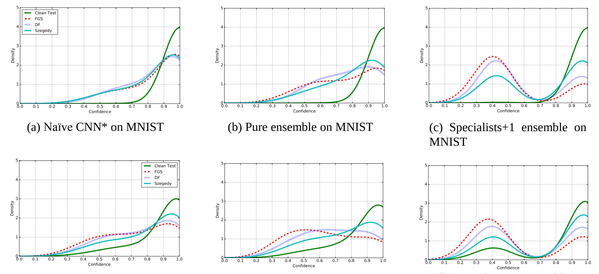
Two types of error should be considered:
1. **Error E_D on the clean set**, counts both clean samples that are misclassified and correctly classified rejected clean samples
2. **Error E_A on the adversaries set**, considers misclassified adversarial instances that are not rejected

8 EXPERIMENTAL RESULTS

Specialists+1 ensemble is compared with
- Ensemble of 5 generalists, i.e. pure ensemble,
- Naïve CNN*

Tested on three types of adversaries: FGS, DeepFool (DF), [Szegedy et al., (2013)]

- **Distribution of confidence**

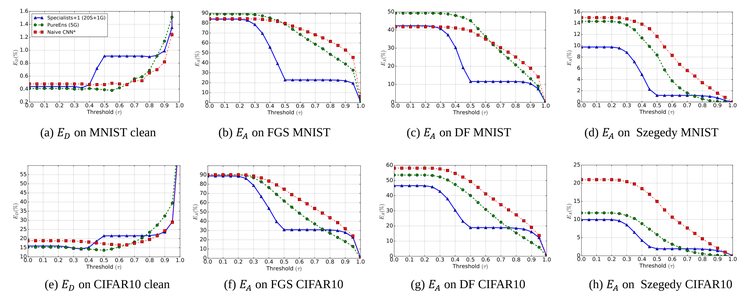


(a) Naïve CNN* on MNIST (b) Pure ensemble on MNIST (c) Specialists+1 ensemble on MNIST

(d) Naïve CNN* on CIFAR10 (e) Pure ensemble on CIFAR10 (f) Specialists+1 ensemble on CIFAR10

Figure 4: Confidence densities on MNIST (the first row) and CIFAR-10 (the second row)

- **Error rate**



(a) E_D on MNIST clean (b) E_A on FGS MNIST (c) E_A on DF MNIST (d) E_A on Szegedy MNIST

(e) E_D on CIFAR10 clean (f) E_A on FGS CIFAR10 (g) E_A on DF CIFAR10 (h) E_A on Szegedy CIFAR10

Figure 5: Error rates E_D on clean test samples and error rates E_A on their corresponding adversaries as a function of threshold (τ) for MNIST and CIFAR10 datasets.

9 CONCLUSION

- Without training from adversaries and by leveraging diversity in specialists ensemble, clean samples are discriminated from adversaries.
- Increasing the robustness of CNNs by refusing the suspicious samples.