

NIPS 2017 Competition on adversarial attacks and defences

Alexey Kurakin Ian Goodfellow Samy Bengio
adversarial-examples-competition@google.com

March 14, 2017

0.1 Overview of the competition

To accelerate research on adversarial examples and robustness of machine learning classifiers we propose a NIPS2017 competition that encourages researchers to develop new methods to generate adversarial examples as well as to develop new ways to defend against them.

As a part of competition researchers will submit methods to craft adversarial examples as well as models which are robust to adversarial examples.

0.2 Keywords

Adversarial examples, machine learning security.

0.3 Novelty

Authors of this competition proposal are unaware of any other competitions related to adversarial examples.

1 Competition description

1.1 Background and impact

Recent advances in machine learning and deep neural networks enabled researchers to solve multiple important practical problems like image, video, text classification and others.

However most existing machine learning classifiers are highly vulnerable to adversarial examples [1, 2, 3, 4]. An adversarial example is a sample of input data which has been modified very slightly in a way that is intended to cause a machine learning classifier to misclassify it. In many cases, these modifications can be so subtle that a human observer does not even notice the modification at all, yet the classifier still makes a mistake.

Adversarial examples pose security concerns because they could be used to perform an attack on machine learning systems, even if the adversary has no access to the underlying model.

Moreover it was discovered [5, 6] that it is possible to perform adversarial attacks even on a machine learning system which operates in physical world and perceives input through inaccurate sensors, instead of reading precise digital data.

In the long run, machine learning and AI systems will become more powerful. Machine learning security vulnerabilities similar to adversarial examples could be used to compromise and control highly powerful AIs.

Thus, robustness to adversarial examples is an important part of machine learning and AI safety problem.

In this competition participants will be given two tasks. The first one is to construct adversarial examples for image classification system. The second task is to develop an image classification model which can resist adversarial examples. Adversarial examples constructed by participants will be used to test the robustness of other participants' image recognition models.

We expect that this competition will lead to development of new models robust to adversarial examples, which will be important scientific breakthrough.

The word "adversarial" was one of the top 5 most frequent words in ICLR 2016 submission titles. We expect to cap enrollment in this competition to 100 people in order to provide more computational resources per team. Given the capped enrollment, we expect we will have no difficulty filling the competition. I previously ran a Google-supported competition¹ with over 200 teams, in 2013 when deep learning was not nearly as popular as it is today (Google Trends shows 10X more search volume for "deep learning" today than at the time of the previous competition).

1.2 Data

Adversarial examples are applicable to various types of machine learning problems. In this competition we have chosen to construct and classify adversarial examples for image classification because baselines are readily available.

In particular we will use the same image classification problem as in the ImageNet challenge [7]: classification of an image into one of 1000 classes. However instead of using the ImageNet images we will collect and label small set of images which were never used before. Using our own set of images will help to avoid overfitting to public data.

Using the same set of classes as in ImageNet challenge would allow participants to use any publicly available ImageNet classifier as a base for their method. Participants can also train their own classifiers from scratch on the ImageNet training set. In such case participants have to obtain ImageNet training set on their own, which should not be a problem since ImageNet dataset is free for research use.

¹<https://www.kaggle.com/c/challenges-in-representation-learning-the-black-box-learning-challenge>

Test images collected by us will be chosen so that existing classifiers such as ResNets and Inception v3 have a low error rate. This way, the competition focuses on robustness to adversarial examples, rather than on advancing the state of the art in traditional image classification.

We will split all collected images into two subsets: 1,000 development images and 10,000 final images.

We will release 1,000 development images as a part of development package which participants can use to work on their submissions and which will be used for evaluation of intermediate test rounds.

10,000 final images will be kept secret until after the competition and will be used for final evaluation of all solutions. In the first task participants will be crafting adversarial examples for these 10,000 images. Then all of these images will be combined together and used to test models submitted for the second task. Given that we’re expecting 100 participants, the combined dataset will contain more than one million images, which should be enough to make competition interesting.

1.3 Tasks and application scenarios

The competition will include two tasks:

1. **Generate adversarial examples.** In this task participants have to craft adversarial images for each of given input images. The goal is to generate an adversarial image which are similar (in terms of L_∞) to given input images, however which will likely be classified differently compared to clean image. Constraint on maximum distance between generated adversarial images and corresponding clean images will be enforced by organizers to ensure that adversarial images are similar to clean images. All adversarial images violating the constraint will be projected onto the boundary of the L_∞ -neighbourhood of corresponding clean images.
2. **Classify adversarial examples.** In this task each participant has to classify all adversarial images generated by all other participants. The goal is to correctly classify as many images as possible.

Participants can opt to submit a solution either for one or both of these tasks, however will be encouraged to submit solutions for both tasks.

A good solution for the second task would be advantageous for developing new models which are robust to adversarial examples.

Given that we would have multiple rounds of evaluation, good solutions for the first task would push researchers to develop better solutions for second task. Also solutions for first task are interesting by themselves because they can further our understanding of how machine learning models work.

1.4 Metrics

For the evaluation we will have n participants. Each participant will submit an attack (an executable that generates an adversarial example), a defence (a model which is potentially robust to adversarial examples) or both. Let A be the set of participants who submitted attacks and D be the set of participants who submitted defence.

We will evaluate each defence against each attack and will obtain a set of values $a_{i,j}$ — accuracy of model j on adversarial examples generated by participant i . Accuracy will be computed as fraction of correctly classified images, thus will be in interval $[0, 1]$.

Then each attack will be assigned a score based on how often it can fool defences:

$$AttackScore_i = \frac{1}{|D - \{i\}|} \sum_{j \in D, j \neq i} (1 - a_{i,j})$$

where $|\bullet|$ is cardinality of a set and $D - \{i\}$ is a difference of sets D and $\{i\}$.

Defences will be assigned a score based on how often they can correctly classify adversarial examples:

$$DefenceScore_j = \frac{1}{|A - \{j\}|} \sum_{i \in A, i \neq j} a_{i,j}$$

In both cases higher score means better attack or defence. The attack with the highest score will be considered the winner in the attack category. The defence with the highest score will be considered the winner in the defence category.

1.5 Baselines and code available

As an attack baseline we will use variation of fast gradient sign method without true labels, as a defence baseline we will use adversarially trained Inception v3 model [8].

The code of the baseline attack is already available as a part of the CleverHans library [9]. The code of the Inception v3 network is available as a part of OpenSource TensorFlow distribution [10]. Model weights of adversarially trained Inception v3 will be released as a part of dev package for participants.

1.6 Tutorial and documentation

We will release development package which will contain:

- Placeholder TensorFlow code of the attack with tutorial on how to use it.
- Code of baseline attack.
- Placeholder TensorFlow code of the defence model with tutorial on how to use it.

- Code and model weights of adversarially trained Inception v3 model.
- Set of 1000 images which participants can use for testing of their code and for intermediate competition rounds.
- Tool to run attack on 1000 development images and then run defence model on generated adversarial images, so participants will be able to get an understanding of whether their attack is good or bad by themselves.

Some of the above mentioned code is already available online, which includes attacks from CleverHans library [9] and code of Inception v3 model. For convenience of the participants we will package all code together and add tutorials and images.

2 Organizational aspects

2.1 Protocol

We will organize several test rounds of competition and one final round. Only submission for final round will be used for scoring and determining winners.

Submissions for the final round will be run by organizers on Google Cloud.

Test rounds are optional for participation and main purpose of test rounds is to help participants test their methods and ensure that they could be run on Google Cloud. For that purpose participants will be provided with tools, tutorials and some Google Cloud credits.

2.1.1 Test rounds

In test rounds we will use 1000 dev images for evaluation.

Each test round will follow following protocol:

1. Each participant runs their own generator of adversarial examples on 1000 images and submits generated images.
2. Organizers group all submitted adversarial examples, enforce constraints on the distance between clean and adversarial images, then release all images to participants.
3. Each participant runs their classification model on the released adversarial examples and submit classification labels.
4. Organizers score submissions and release results to participants.

2.1.2 Final round

In the final round we will use 10,000 secret images for evaluation. Since result of final round will be used to score participants and determine winners, additional measures will be taken to prevent any type of cheating:

- Images used for final evaluation will be kept secret until after the competition.
- Participants will be required to release open source code of their submission in order to be eligible for scoring.
- All code will be run by organizers instead of participants, however intermediate results will be eventually released to public. This will provide anybody ability to verify accuracy of results and fairness of the competition.

Final round will follow following protocol:

1. Participants submit source code of their solutions (both attack and defence) to the organizers.
2. Participants open source code of their solutions, before organizers start evaluation.
3. Organizers run all submitted generators of adversarial examples on 10000 secret images.
4. Organizers enforce constraints on the distance between clean and adversarial images.
5. Organizers run all classifiers on all generated adversarial images.
6. Organizers compute score of all attack and defences and determine winners.
7. Organizers announce results of the competition and release secret 10000 images, their true labels and all adversarial images generated by all participants to the public.

2.2 Rules

1. Google employees and affiliated people are ineligible from participation.
2. To be eligible for scoring, participants are required to open source code of their submission.
3. Executables that produce errors receive no points for the inputs that resulted in errors. Contestants are responsible for testing their executables on Google Cloud before the final round.

4. Contestants are forbidden from submitting fake results during the test rounds. Though these test rounds are not used to choose the final winner, contestants might use fake results to intimidate other contestants unfairly.
5. Contestants are forbidden from colluding, either with other contestants, or making their attack and defense mechanism collude, either to make attack images that are specifically designed for their defense mechanism to be able to classify easily, or by making classifiers that are designed to be easily broken specifically by their attack. It is legitimate to train a contestant's model using that contestant's attack, but it is not legitimate to, for example, embed a steganographic message in an attack image revealing the class of the image only to defenders who have the steganographic key. Enforcement of this rule is at the sole discretion of the organizers, and they should be contacted ahead of time if a contestant wishes to pursue a strategy in a gray area.
6. Each classifier and attack must be stateless and act one image at a time, to prevent strategies such as memorizing pre-attack images and classifying replayed versions of them at defense time. During the final round, this rule will be enforced by the nature of the Cloud software environment, but during test time contestants are expected to voluntarily comply with it.

2.3 Schedule

Proposed schedule of the competition:

April 14, 2017 Launch web-site with announcement and competition rules. Start active advertisement of the competition.

end of June, 2017 Release dev kit and dev data for participants.

end of June, 2017 - October 1, 2017 Participants working on their solutions. In the meantime we organize few intermediate rounds of evaluation.

October 1, 2017 Deadline for final submissions.

October 1, 2017 - end of October, 2017 Organizers evaluate submissions.

end of October, 2017 Announce competition results, release evaluation set of images.

2.4 Competition promotion

The contest will be promoted using the organizers' Facebook, Twitter, Google+ and Reddit accounts, as well as the CleverHans blog and various university mailing lists.

2.5 Organizing team

Alexey Kurakin is a senior research software engineer in Google Brain team. Alexey demonstrated that adversarial examples can exist in the physical world and was the first to develop adversarial training at ImageNet scale. Alexey will serve as a coordinator, data provider, platform administrator, and baseline method provider, and evaluator—all within the scope of his regular work as a research software engineer.

Ian Goodfellow is a staff research scientist in the Google Brain team. He is the lead author of the MIT Press textbook Deep Learning (www.deeplearningbook.org) and a co-inventor of adversarial training. He is a developer and maintainer of the CleverHans security library [9]. Ian will serve as a coordinator, data provider, platform administrator, baseline method provider, beta tester, and evaluator—all of which he has done previously for three different Kaggle competitions.

Samy Bengio (PhD in computer science, University of Montreal, 1993) is a research scientist at Google since 2007. Before that, he was senior researcher in statistical machine learning at IDIAP Research Institute since 1999, where he supervised PhD students and postdoctoral fellows. His research interests span many areas of machine learning such as deep architectures, representation learning, sequence processing, speech recognition, image understanding, support vector machines, mixture models, large-scale problems, multi-modal (face and voice) person authentication, brain computer interfaces, and document retrieval. He is action editor of the Journal of Machine Learning Research and on the editorial board of the Machine Learning Journal, has been programme chair of the International Conference on Learning Representations (ICLR 2015, 2016), general chair of BayLearn (2012-2015) and the Workshops on Machine Learning for Multimodal Interactions (MLMI'2004-2006), as well as the IEEE Workshop on Neural Networks for Signal Processing (NNSP'2002), and on the programme committee of several international conferences such as NIPS, ICML, ICLR, ECML and IJCAI. Samy will supervise Alexey and Ian's management of this competition, much as he has supervised their previous work on adversarial examples.

3 Resources

3.1 Existing resources, including prizes

We do not plan to provide monetary prizes.

References

- [1] Battista Biggio, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning

- at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013.
- [2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, abs/1312.6199, 2014.
 - [3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.
 - [4] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *ArXiv e-prints*, May 2016b.
 - [5] Alex Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR’2017 Workshop*, 2016.
 - [6] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*, October 2016. To appear.
 - [7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.
 - [8] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR’2017*, 2016.
 - [9] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, and Patrick McDaniel. cleverhans v1.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.
 - [10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.