**Selective Order Data Injection**

Patrick Cooper1, Author Name2 and Author Name1,2

1 Department One, Institution One, City One, Country One

2 Department Two, Institution Two, City Two, Country Two

E-mail: xxx@xxx.xx

Received xxxxxx

Accepted for publication xxxxxx

Published xxxxxx

Abstract

The idea is to present maximally different examples in rapid succession. In this way we hope to establish boundary conditions around the problem space. The intuition for why this works mathematically is directly related to

Keywords: adversarial examples, deep learning, artificial intelligence, synthetic data injection, image classification,

1. Introduction

As deep learning continues to grow more and more popular for isolated applications with multiple sorts of data and as transfer learning continues to grow increasingly popuar with the recent scucess of stylegans and related technologies, it likewise becomes increasingly important to create methodologies which make maximum use of the unque origins of the data itself. Synthetic data presents the unqiue oppurtunity, when paried with adversarial learning, to increase accuracy, improve training speed, and generally make the model more robust to attack. In this paper we examine training speed, related work on model accuracy and robustness can be found \_\_\_\_\_ and \_\_\_\_\_\_ respectively.

1.1 Training Speed

In production environments training speed can be relvenetly paired with the speed at which a model becomes ready to go online and, often more importantly, the efficiency with which that model can be deployed. The fast gradient method of calculating adversarial examples can be deployed for significantly less cost than additional gradient descent methods. This study aims to describe how adversarial examples can improve training speed, and likewise when such usage can benefit the resulting deployment.

Proin nec augue. Quisque aliquam tempor magna. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc ac magna. Maecenas odio dolor, vulputate vel, auctor ac, accumsan id, felis. Pellentesque cursus sagittis felis. Pellentesque porttitor, velit lacinia egestas auctor, diam eros tempus arcu, nec vulputate augue magna vel risus. Cras non magna vel ante adipiscing rhoncus.

Vivamus a mi. Morbi neque. Aliquam erat volutpat. Integer ultrices lobortis eros. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin semper, ante vitae sollicitudin posuere, metus quam iaculis nibh, vitae scelerisque nunc massa eget pede. Sed velit urna, interdum vel, ultricies vel, faucibus at, quam. Donec elit est, consectetuer eget, consequat quis, tempus quis, wisi.

1.1.1 General Robustness Increase.

Training the network purely using adversarial examples has an additional benefit. Namely, that is that adversarial examples are more difficult to create in this instance. Additionally, we see more accurate confidence ratings assigned to liklihoods which otherwise might have been given absurd uniform characteristics.

Use generalized adversarial examples from a larger dataset to allow for increased robustness. So there is no need to increase training time to increase performance.

General model will be applied to mnist, sun397, and imagenet.

Aliquam nonummy adipiscing augue. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

2. Discussion of Related Work

Within the work of Yoshua Bengio we see the abstract notion that concepts ought to be introduced in order of increasing complexity. Regrettably, this work suggests a less high-level methodology is best. Here I propose that optimally, the examples ought to be fed in with respect to maximal difficulty as determined by an adversarial learner at each intermediate step.

3. Experimental Setup

Implemented with TensorFlow v2.0. All experiments were performed on a Dell XPS 15 using an 8th generation i7 and a 1050ti m series and were computed using.

In addition, studies were run on a 2070 super.

Likewise results have been replicated on a 3rd generation raspberry pi using tensorflow on the onboard ARM processor.

Corresponding experimental code can be found located at:

Results have been reporduced indepdnently and reported by:

A grid search was completed across the top 5 performing adversarial learners on the ImageNet database. Learners are here classified in order of execution time. A more thorough analysis of the algorithms behind each method reveals that their execution time is justified via there complexity.

Each method has been examined in isolation. We segregate each set of trials according to the adversarial example generation method deployed, we then examine the speed at which the model converges in each instance.

Within the confines of this example we can note that even reltiavely simplistic adversarial generation yields beneificial results over an extended period of execution. The benefit derived from each training example’s adversarial nature degrades over time until the models’ results converge and the performance of the standard model outperforms that of the adversarial training examples.

This process was repeated using MNIST with similar results. See fig… Notably, the…

Finally a study was conducted to evaluate the effects of the variablity of synthetic data injection vs natural data injection on the speed of the results. Three general approaches were used. These were sampled according to uniform spacing of 2 random variables quantity of adversarial example, quantity of synthetic examples, quantity of…

Ultimately better than state of the art performance was achieved on ImageNet and MNIST for the first X epochs of the training. Stateoftheart performance is here defined using the results of \_\_\_\_\_. As previously stated this model converges near the state of the art level of accuracy, but does not exceed it.

4. Results

While we do not see a performance in eventual learning convergence, it clearly corresponds to

5. Explanation

The inherent linearity of a higher dimensional space implies the definition of the problem space operates with superior accuracy when adversarial examples are deliberately injected into an intelligent system.

Let us establish an experiment whereby we have an extremely limited dataset and there exists a necessity to achieve performance on all example classes within the dataset which matches or exceeds at least 75%. There are many ways this can be accomplished, but an additional methodology can now be put forward in the selective training on adversarial demonstrated examples.

Furthermore, there exist scenarios where the performance in edge cases is the most important a model may encounter. This may include creating production systems which are resilient to

We can imagine production scenarios where it is highly important to establish high performance in edge cases, under such circumstances it is essential that approximate performance be established before ideal performance.

Running an adversarial learner each step of the training may prove to be impractical for most applications, however, apart from the obvious insights yielded into deep learning networks in general we can imagine a scenario where we have a set of samples determined at some selected step size. Often a per epoch/s basis likely provides a rough approximation of such. A general optimization for local optimizations of a such an algorithm can be computed with an L2 distance.

If we inject adversarial examples into a network where these synthetic examples are the result of the generation of a generative adversarial network, we can create a model which only evaluates its performance based upon the few natural examples. In this manner we can constructively improve performance of the network even while the network expands.

1. We begin with natural adversarial cases

2. We loosen the adversarial grip on the network even as we increase the amount of synthetic allowable data.

3. The network being trained biases the level of insight of new training examples via their relevance on the new data. This allows the network to generate synthetic minds, which are graded based on the performance relative to the naturally occurring data. That is, gans, with evolutions. We need a way to quantify their performance. Augmented Gans with harsh self grading.

6. Future Work

Future work in this domain should include operating on large scale datasets. Additionally, I would like to put forward the open question on how this dynamic affects deep double descent.

Effort may also be productively put forward on inquiring how such a network can be integrated into a large-scale simulation and integration of synthetic data. The principle notion here is the utilization of pair of networks each has access to some amount of good data, data which adheres to some performance accuracy defined to be adequate to fully justify the performance of the model.

This will be the topic of my next report.

Idealized network performance via synthetic approximation of data to some point.

Acknowledgements

This work builds heavily on preious topics explored by a hoast of researchers exploring the topic of selective sample injection into a network.

Special acknowledgements to Craig Jorgensen whose ideas prompted this research.

References

1. Surname A, Surname B and Surname C 2015 *Journal Name* **37** 074203
2. Surname A and Surname B 2009 *Journal Name* **23** 544