Application Testing of Generative Adversarial Privacy

Nicholas Johnson*, Stephanie Sanchez*, Vishal Subbiah*
Stanford University
Computational and Mathematical Engineering
nickj@stanford.edu, ssanche2@stanford.edu, svishal@stanford.edu

November 20, 2017

Abstract

*To protect personal privacy from **inference attacks**, using techniques from **Generative Adversarial Networks (GANs)**, we have implemented a simple **Generative Adversarial Privacy Architecture** (GAP) architecture composed of an encoders, a distortion metric, and two classifiers to illustrate the concept. This architecture distorts input data to hinder the predictions of one classifier while minimally affecting the accuracy of the other classifier. This is directly related to deterring an inference attack by lowering the ability to infer sensitive private information while allowing prediction of nonsensitive public information.*

1 Motivation

1.1 Privacy

There exist many online portals in which individuals post information whether it be text, photos, videos, etc. and some information can be released to third parties. With so much personal information distributed it is important to retain degrees of privacy while not affecting what is already chosen to be public information.

1.2 Inference Attacks

Machine learning (ML) methods serve many purposes today. The most revered applications of ML are normally coupled with benevolent intentions such as adverting cyber attacks, classifying materials in images for security and health purposes, etc. But ML methods do not necessarily have to be applied for favorable causes. Inference attacks, for instance, are adversary learning methods that can infer private information from public information or data. For this reason it is essential to protect privacy by deterring adversarial machine learning.

[0]*All authors contributed equally to this work.

1.3 Generative Adversarial Privacy (GAP)

To protect the privacy of public data we have implemented a model, generative adversarial privacy (GAP), that distorts data in an incentivized manner. This enables the protection of sensitive information by obscuring generative adversarial networks (GANs) inferences of sensitive data and while not affecting the inference of nonsensitive attributes.

1.4 Related Work and data

The work that we have seen in this space has been predominantly focused on an analytical approach in order to more rigorously prove privacy guarentees[2] [3] [4]. We have sought to reproduce the methodology in [2] with our GAP architecture and used a similar data set to procede in that direction. These papers provide theoretical and small scale practical verification of their algorithms to obfuscate sensitive features without ruining other potential inferences from the data. None of the papers evaluates how this would occur in an adversarial game setting where the predictive algorithms have information about the generator function and can adapt to the obfuscated data provided, which could be the case in a real world situation.

2 Methods

We have three functions and one metric as seen in figure 1. The functions are E,G,S for encoder, gender classifier, and smile classifier respectively. A distortion metric D is included to allow for further constraints on the encoder function. We developed a multiple tiered approach of increasing complexity in order to make the problem tractable in the time alloted. We consider the encoder and classifiers as distinct entities for the sake of the GAP architecture. As such we tried two different encoder strategies and two classifier strategies for this project. Due to time constraints we were unable to fully run an iterative

game theory approach and instead used statically trained classifiers to train our encoder.

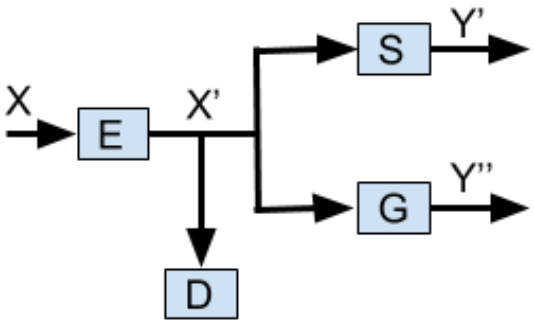

Figure 1: GAP Architecture

2.1 Data Preprocessing

The images were of various sizes so we reshaped them all to 256 by 256 pixels with 3 color channels. The labels provided for smile were used as is but for some reason the gender labels had 3 categories. The third category only had a handful of samples so we removed these from our data set since we weren't able to discover the source of this unexpected anomaly. 256 by 256 was chosen based on the average sizes of the dataset. Also, it fits well in case of future work to use a pretrained neural network, such as ResNets or VGG-Net provided by PyTorch, where the images would be cropped to 224 by 224.

2.2 Classifiers

Classifying smiles and gender is a project in it's own right [1]. We originally implemented a two layer neural network, for initial verification of the code architecture and easier machine learning troubleshooting due to faster run times. The two layer neural network acted a place holder for deeper neural networks.
For GAP to generalize the specific classifier architectures is not important. Deep neural networks are used as generalization of complex adversaries. Obviously, in order to prove this concept in practice we had to decide on an architecture. We settled on a VGG-NET architecture with 3 convolution layers and 3 fully connected layers. We used a RELU activation function on all nodes.

2.3 Compressive Encoders

We know that neural nets can be susceptible to pixel attacks [6]. In practice we perceive this to be more information than would be available to a real world generator. However, compressive algorithms provide an interesting non-reversible information distorting encoder. We implemented PCA as an encoder, where we reconstruct our image tensor using components equal to the number of nodes in the autoencoder. This was chosen because the autoencoder is performing an information reduction of similar order.

2.4 Shallow Autoencoders

By the universal approximation theorem, neural networks can reproduce any function under the appropriate constraints [5]. There is existing research in which an alternating optimization scheme is performed to calculate an explicit encoder function [2]. We used a neural network autoencoder as an alternative to the oscillating optimization scheme and create a more widely applicable product for performing GAP in practice. As such, a first step was showing that we are able to back propagate through the architecture of multiple neural networks to appropriately optimize their respective loss functions. We proved that in the project milestone and two layer autoencoder was suprisingly successful. We settled on a three layer fully connected neural network with the hiden layers of size 1024 and 256.

3 Numerical Results

Our smile and gender models were trained on the raw data sets in order to establish a baseline accuracy. As a baseline comparison PCA was used as lossy compression with the 256 principal components used to recreate the images. This distorted data was then given to the trained classifiers to determine their accuracy. Next we trained and tuned our autoencoder with the static classifier models. This yielded favorable results to predict the Smile Model while lowering the accuracy of the Gender Model as shown in Figure 2.

	GAN Accuracy	Gender Accuracy	Smile Accuracy
Autoencoder	24%	42%	64%
Encoder-PCA	26%	69%	36%
No Encoder	N/A	72%	74%

Figure 2: Numerical results with two different encoders.

3.1 Deep Neural Net Classifiers

For our gender and smile classifiers, G and S respectively. We were able to achieve an accuracy of approximately (72%, 74%), as shown in figures 3 and 4, using our VGGNet structure on the unmodified input data. We used this same architecture for G and S in the below experiments on the encoder function E.

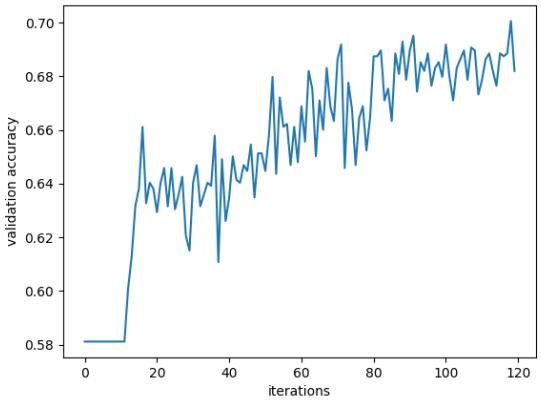

Figure 3: Gender accuracy without an Encoder

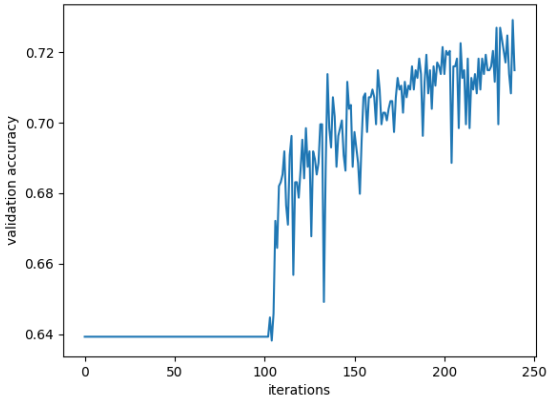

Figure 4: Smile accuracy without an Encoder

3.2 PCA Encoder

Since PCA is not incentivized towards the success or failure of either classifier it yields results opposite to the autoencoder,as shown in Figure 2. This is because features for the Gender Model are more distinct than the Smile Model features. Since a smile is so localized in the face its features can be destroyed more easily in compression. These results would be excellent for reducing the accuracy of the Gender Model while decreasing the accuracy for the Smile Model however this shows our autoencoder was effective and not simply succeeding because of the sensitivity of the image features.

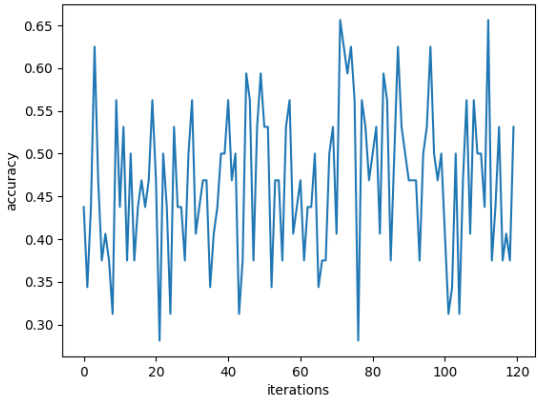

Figure 5: Gender accuracy with autoencoder

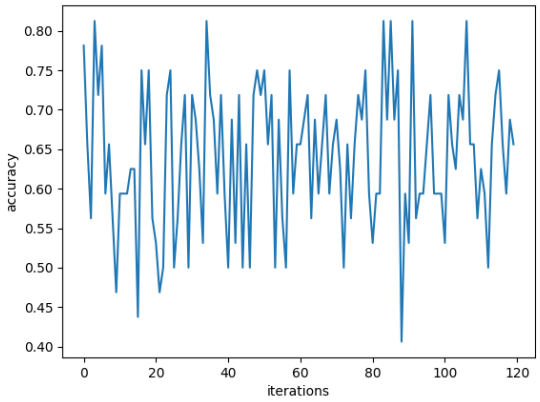

Figure 6: Smile accuracy with autoencoder

3.3 Neural Net Autoencoder

We use our 2 hidden layer autoencoder, with the cross entropy loss function to confirm that we could train a three neural network system in the architecture outlined in [4]. The loss function for the the overall system the was the

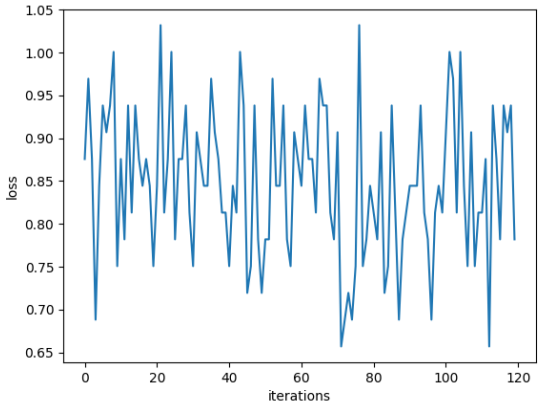

Figure 7: Gender loss with autoencoder

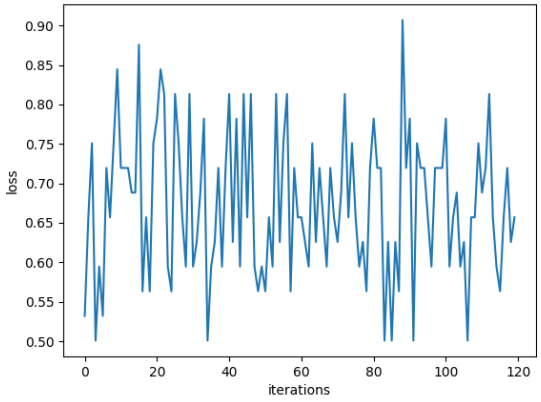

Figure 8: Smile loss with autoencoder

difference between the smile and gender cross entropy loss. We were able to yield an accuracy of (42%, 64%) at the output of our classifiers, with a total system accuracy of 24%. Our total system accuracy was defined when smile predicted currently and gender predicted incorrectly for the sample input sample. We can see that the autoencoder has a significant effect as seen in the smile accuracy in figure 2. Our autoencoder was able to identify the right features to modify without affecting the majority of the prominent features. This also shows, that we can disrupt a complicated model using a relatively simple model.

4 Future Work

We achieved our baseline for this project but due to time constraints and some complications with the classifiers we were not able to explore a variety of topics of interest. We didn't play with the distortion metric in order to further constrain our autoencoder. A fully connected network for the autoencoder would allow for distortion without compression. Adding random noise to the encoder to make it non-reversible is of interest for differential privacy.
Another approach would be to start with an externally trained deep neural network classifier and let it adapt to the perturbed input data. The accuracy could then be compared to the neural network classifiers trained during the GAP process to determine relative performance. This is important because it leans on finding the game theory equilibrium point which more accurately represents a real world situation where adversaries are constantly evolving. Using a K-means compression algorithm (as the encoder) would evaluate all three channels simultaneously and give another comparison to PCA.

5 Contributions

Vishal established the framework and models which we leveraged in our experiments. He also fine tuned the autoencoder parameters. Stephanie and Nick worked on the two neural network classifiers. Nick reviewed background literature and contributed to setting up the GCP system for training the deep neural networks.

References

[1] Ari Ekmekji *Convolutional Neural Networks for Age and Gender Classification*.

[2] Jihun Hamm *Minimax Filter: Learning to Preserve Privacy from Inference Attacks*.

[3] Ke Xu, Tongyi Cao, Swair Shah, Crystal Maung, Haim Schweitzer *Cleaning the Null Space:A Privacy Mechanism for Predictors*.

[4] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal *Context-Aware Generative Adversarial Privacy*

[5] G. Cybenko *Approximation by superpositions of a sigmoidal function*

[6] Jiawei Su, Danilo Vasconcellos Vargas, Sakurai Kouichi *One pixel attack for fooling deep neural networks*